

Πληροφορική και Επιστήμη Ηλεκτρονικών Υπολογιστών, Β΄ Λυκείου

Σημειώσεις και ασκήσεις

Συγγραφή:

Η Ομάδα Υποστήριξης του Αναλυτικού Προγράμματος και των Συμβούλων Καθηγητών Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών:

Α΄ και Β΄ Έκδοση

Σωκράτης Μυλωνάς (ΒΔ)
Πάυλος Παυλικκάς
Παναγιώτης Ηρακλέους
Μιλτιάδης Χαριλάου
Δημήτρης Μαυροβουνιώτης
Πόλα Μάκκουλα

Γ΄ Έκδοση

Πόλα Μάκκουλα
Κωνσταντίνος Σωφρονίου (ΒΔ)

Ηλεκτρονική σελίδωση:

Πόλα Μάκκουλα
Καθηγήτρια Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών

Εποπτεία:

Μάριος Μιλτιάδου
Μιχάλης Τορτούρης
Σωκράτης Μυλωνάς
Επιθεωρητές Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών

Γλωσσική επιμέλεια:

Παναγιώτης Πετρίδης
Λειτουργός Υπηρεσίας Ανάπτυξης Προγραμμάτων

Σχεδιασμός εξωφύλλου:

Σωκράτης Μυλωνάς
Επιθεωρητής Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών

Επιμέλεια έκδοσης:

Μαρίνα Άστρα Ιωάννου
Λειτουργός Υπηρεσίας Ανάπτυξης Προγραμμάτων

Συντονισμός έκδοσης:

Χρίστος Παρπούνας
Συντονιστής Υπηρεσίας Ανάπτυξης Προγραμμάτων

Α΄ Έκδοση 2016

Β΄ Έκδοση 2017

Γ΄ Έκδοση 2020

Εκτύπωση: Printco Cassoulides Ltd

© ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΠΟΛΙΤΙΣΜΟΥ, ΑΘΛΗΤΙΣΜΟΥ ΚΑΙ ΝΕΟΛΑΙΑΣ
ΠΑΙΔΑΓΩΓΙΚΟ ΙΝΣΤΙΤΟΥΤΟ ΚΥΠΡΟΥ
ΥΠΗΡΕΣΙΑ ΑΝΑΠΤΥΞΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ

ISBN: 978-9963-54-261-1



Στο εξώφυλλο χρησιμοποιήθηκε ανακυκλωμένο χαρτί σε ποσοστό τουλάχιστον 50%, προερχόμενο από διαχείριση απορριμμάτων χαρτιού. Το υπόλοιπο ποσοστό προέρχεται από υπεύθυνη διαχείριση δασών.

Πρόλογος

Με ιδιαίτερη χαρά και ικανοποίηση προλογίζω το έντυπο υλικό για την υποστήριξη της διδασκαλίας του μαθήματος της Πληροφορικής και της Επιστήμης των Ηλεκτρονικών Υπολογιστών στη Β' Λυκείου.

Μέσα από την προσπάθεια διαμόρφωσης ενός νέου αναλυτικού προγράμματος, δρομολογήθηκαν και εφαρμόζονται αλλαγές με στόχο την αναβάθμιση και τον εκσυγχρονισμό του μαθήματος τόσο στο Γυμνάσιο όσο και στο Λύκειο. Το έντυπο αυτό υλικό δημιουργήθηκε ακριβώς, για να υποστηρίξει τη διδασκαλία και τη μάθηση στο πλαίσιο του ριζικά αναθεωρημένου σε περιεχόμενο και μεθοδολογική προσέγγιση μαθήματος της Πληροφορικής και της Επιστήμης Ηλεκτρονικών Υπολογιστών στη Β' Λυκείου.

Η δομή του μαθησιακού υλικού βασίζεται στις επτά ενότητες του αναλυτικού προγράμματος, καλύπτοντας το φάσμα των γνώσεων, των ικανοτήτων, των δεξιοτήτων, των στάσεων και των αξιών που προκύπτουν από τον γενικό σκοπό του μαθήματος, να προετοιμάσει δηλαδή τους/τις μαθητές/τριες για την ένταξή τους στην Κοινωνία της Πληροφορίας. Τα εφόδια αυτά θα τους επιτρέψουν την υπεύθυνη, την ενσυνείδητη, την ασφαλή, την αποδοτική και τη δημιουργική χρήση σύγχρονων τεχνολογιών της Πληροφορικής και της Επιστήμης των Ηλεκτρονικών Υπολογιστών στον επαγγελματικό τομέα που θα επιλέξουν, στον κλάδο σπουδών που πιθανό να ακολουθήσουν αλλά και στην καθημερινότητά τους.

Το διδακτικό βιβλίο *Σημειώσεις και Ασκήσεις* δεν αποτελεί διδακτικό εγχειρίδιο αλλά υλικό αναφοράς, το οποίο περιέχει παραδείγματα, επεξηγήσεις και επιπρόσθετες πληροφορίες και χρησιμοποιείται ως πηγή άντλησης πληροφοριών και είναι στενά συνδεδεμένο με το αναλυτικό πρόγραμμα. Στην αρχή του κάθε κεφαλαίου διατυπώνονται οι διδακτικοί στόχοι, βασισμένοι στους δείκτες επιτυχίας και το περιεχόμενο αναπτύχθηκε με βάση τους δείκτες επάρκειας. Στο τέλος κάθε κεφαλαίου περιλαμβάνονται ποικίλες δραστηριότητες με διαφοροποιημένο δείκτη δυσκολίας, ώστε να είναι δυνατή η προσαρμογή του μαθήματος στις ικανότητες, στις απαιτήσεις και στις ανάγκες όλων των μαθητών/τριών. Υιοθετούνται σύγχρονα διδακτικά μοντέλα, που στηρίζονται στην προώθηση εξελιγμένων μεθόδων οικοδόμησης της γνώσης και συστηματικών τρόπων προσέγγισης για την επίλυση προβλημάτων, ώστε οι μαθητές/τριες να προετοιμάζονται για να λειτουργήσουν στο συνεχώς μεταβαλλόμενο περιβάλλον των τεχνολογιών της πληροφορίας και των επικοινωνιών.

Ευχαριστώ θερμά όλους τους συντελεστές της προσπάθειας αυτής, ιδιαίτερα την ομάδα υποστήριξης της εφαρμογής του αναλυτικού προγράμματος και των συμβούλων καθηγητών της Πληροφορικής και της Επιστήμης των Ηλεκτρονικών Υπολογιστών για τη δημιουργία του έντυπου υλικού αλλά και του ηλεκτρονικού υλικού που το συνοδεύει, του οικείου επιθεωρητές, τους λειτουργούς της Υπηρεσίας Ανάπτυξης Προγραμμάτων και ιδιαίτερα τους επιστημονικούς συνεργάτες, τον Δρα Χρύση Γεωργίου (Τμήμα Πληροφορικής του Πανεπιστημίου Κύπρου), τη Δρα Χαρούλα Αγγελή (Τμήμα Επιστημών της Αγωγής του Πανεπιστημίου Κύπρου) και τον Δρα Ανδρέα Ανδρέου (Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής του Τεχνολογικού Πανεπιστημίου Κύπρου) για τη συμβολή τους στην ανάπτυξη του υλικού αυτού.

Δρ Κυπριανός Δ. Λούης

Διευθυντής Μέσης Εκπαίδευσης

ΠΕΡΙΕΧΟΜΕΝΑ

Εισαγωγή	9
ΕΝΟΤΗΤΑ Β1 Βασικές Έννοιες της Πληροφορικής και της Επιστήμης Ηλεκτρονικών Υπολογιστών	11
B1.1 Αναπαράσταση Δεδομένων	13
1. Αναπαράσταση Δεδομένων – Δυαδικό Σύστημα	13
2. Αναπαράσταση Χαρακτήρων με Δυαδικά Ψηφία	13
3. Αναπαράσταση Αριθμών με Δυαδικά Ψηφία	13
4. Αναπαράσταση Χαρακτήρων με Δεκαεξαδικά Ψηφία	14
5. Διεθνές Πρότυπο Unicode	15
ΕΝΟΤΗΤΑ Β6 Βάσεις Δεδομένων και Ανάπτυξη Πληροφοριακών Συστημάτων	21
B6.1 Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων	23
1. Εισαγωγή	23
2. Σύστημα, Υποσύστημα και Περιβάλλον	23
3. Εσωτερικό και Εξωτερικό Περιβάλλον Συστήματος	24
4. Πληροφοριακά Συστήματα (ΠΣ)	25
5. Πληροφοριακό Σύστημα	25
6. Τύποι πληροφοριακών συστημάτων	26
7. Κύκλος ζωής και ανάπτυξης πληροφοριακού συστήματος	26
8. Μοντέλα κύκλου ζωής και ανάπτυξης πληροφοριακών συστημάτων	27
9. Διάφορα Μοντέλα:	27
10. Σύγχρονα μοντέλα κύκλου ζωής λογισμικού	28
11. Αναλυτής συστημάτων	28
B6.2 Μοντέλο Καταρράκτη	33
1. Κύκλος ζωής και Ανάπτυξης Πληροφοριακού Συστήματος - Μοντέλο Καταρράκτη	33
2. Φάσεις Κύκλου Ζωής και Ανάπτυξης Πληροφοριακού Συστήματος:	33
3. Κύκλος Ζωής και Ανάπτυξης Πληροφοριακού Συστήματος	33
4. Προκαταρκτική Έρευνα - Σύνταξη μελέτης σκοπιμότητας	34
5. Εξακρίβωση Αναγκών και Καθορισμός Απαιτήσεων	34
6. Καθορισμός Προδιαγραφών	35
7. Σχεδιασμός συστήματος	35
8. Υλοποίηση – Συνένωση Κώδικα και Έλεγχος συστήματος	35
9. Συντήρηση συστήματος	35
B6.3 Μέθοδοι Καταγραφής Προδιαγραφών	39
1. Μέθοδοι καταγραφής προδιαγραφών:	39
2. Λογικά και Φυσικά Μοντέλα	39
3. Διαγράμματα Ροής Δεδομένων (ΔΡΔ) (Data Flow Diagrams)	40
4. Σύμβολα Διαγραμμάτων Ροής Δεδομένων	40
5. Παράδειγμα Διαγράμματος Ροής Δεδομένων	41
6. Βασικοί Κανόνες που διέπουν τον σχεδιασμό Διαγραμμάτων Ροής Δεδομένων (ΔΡΔ)	42
7. Λάθη σε ΔΡΔ	42

B6.4	Διαγράμματα Οντοτήτων-Συσχετίσεων (ΔΟΣ) – Entity Relationship Diagrams (ERDs)	59
1.	Οντότητα (Αρχείο)	59
2.	Σχέσεις (Relationships)	59
3.	Πληθυκότητα (Cardinality)	60
4.	Παραδείγματα Διαγραμμάτων Οντοτήτων-Συσχετίσεων	60
B6.5	Σχεδίαση Συστήματος	65
1.	Σχεδιασμός Οθονών (Forms) και Αναφορών (Reports)	65
B6.6	Υλοποίηση – Συνένωση Κώδικα και Έλεγχος συστήματος	71
1.	Υλοποίηση – Συνένωση Κώδικα και Έλεγχος συστήματος	71
2.	Εισαγωγή στη Microsoft Access	71
3.	Δημιουργία και Αποθήκευση Βάσης Δεδομένων	72
4.	Δημιουργία και Αποθήκευση Πίνακα (Table)	72
5.	Εισαγωγή εγγραφών στον πίνακα από αρχείο	73
6.	Δημιουργία Σχέσεων μεταξύ των Πινάκων (Relationships)	74
7.	Δημιουργία Φόρμας Εισαγωγής Δεδομένων (Form)	75
8.	Δημιουργία Ερωτημάτων (Queries)	76
9.	Δημιουργία Αναφορών (Reports)	77
10.	Δημιουργία Switchboard	78
11.	Εμφάνιση του Switchboard Manager	78
12.	Δημιουργία Switchboard	79
13.	Πρόσβαση στη Βάση Δεδομένων από πολλούς χρήστες	80
ΕΝΟΤΗΤΑ B7	Αλγοριθμική Σκέψη, Προγραμματισμός και Σύγχρονες Εφαρμογές Πληροφορικής	85
B7.1	Προγραμματισμός στη C++	87
B7.1.1	Ξεκινώντας...	89
1.	Εισαγωγή	89
2.	Λίγα λόγια για τη C++	90
3.	Χρησιμοποιώντας το Code::Blocks	90
4.	Δημιουργία νέου προγράμματος:	91
5.	Μεταγλώττιση και Εκτέλεση προγράμματος:	91
6.	Επαναφορά προγράμματος:	91
7.	Παραδείγματα	91
B7.1.2	Τα Πρώτα Προγράμματα σε C++	93
1.	Δομή προγράμματος	94
2.	Επεξήγηση της δομής του προγράμματος	94
3.	Εντολή εξόδου - cout	96
4.	Μορφοποίηση εξόδου	97
5.	Μεταβλητές και Σταθερές	99
6.	Εντολή εισόδου - cin	100
7.	Τελεστές	101
8.	Συναρτήσεις της C++	104
9.	Παραδείγματα απλών προγραμμάτων	104

B7.1.3	Δομές Διακλάδωσης.....	127
1.	Εισαγωγή	127
2.	Απλή δομή διακλάδωσης	128
3.	Συγκριτικοί τελεστές (== , != , > , < , >= , <=)	128
4.	Λογικοί Τελεστές.....	132
5.	Ένθετη δομή διακλάδωσης – Nested if	134
7.	Εισαγωγή στα strings	139
B7.1.4	Δομές Επανάληψης.....	165
1.	Εισαγωγή	165
2.	Τελεστές αύξησης/μείωσης (++, --).....	165
3.	Ο βρόχος for	166
4.	Φωλιασμένες εντολές for – Nested for loops.....	169
5.	Ο βρόχος while.....	170
6.	Ο βρόχος do.....	174
7.	Η Βιβλιοθήκη <climits>	177
8.	Έλεγχος ορθότητας δεδομένων	178
9.	Έλεγχος Προγράμματος	178
B7.1.5	Πίνακες.....	213
1.	Εισαγωγή	213
2.	Δήλωση πινάκων	214
3.	Απόδοση αρχικών τιμών σε πίνακες.....	214
4.	Εισαγωγή στοιχείων σε πίνακα	215
5.	Εμφάνιση στοιχείων του πίνακα	215
6.	Υπολογισμός του μέσου όρου των στοιχείων του πίνακα	216
7.	Παράλληλοι πίνακες	218
B7.2	Δημιουργία Παιγνιδιών για Ηλεκτρονικούς Υπολογιστές.....	241
B7.2.1	Δημιουργία παιγνιδιών για Ηλεκτρονικούς Υπολογιστές.....	243
1.	Εισαγωγή	243
2.	Παιγνίδια και εκπαίδευση	243
3.	Είδη παιγνιδιών.....	244
4.	Σχετικά με το σύστημα ταξινόμησης PEGI	245
5.	Η ιστορία των ψηφιακών παιγνιδιών	248
6.	Ο κύκλος ζωής ενός βιντεοπαιγνιδιού	253
B7.2.2	Εισαγωγή στο GameMaker – Τα Δομικά Στοιχεία ενός Παιγνιδιού ...	259
1.	Εισαγωγή	259
2.	Ενεργοποίηση του GameMaker, άνοιγμα έτοιμου παιγνιδιού και αποθήκευση παιγνιδιού	259
3.	Διαχείριση της εκτέλεσης ενός παιγνιδιού	259
4.	Τα βασικά δομικά στοιχεία (οντότητες) ενός παιγνιδιού στο GameMaker	260
5.	Sprites	260
6.	Εικόνες Φόντου (Backgrounds)	262
7.	Objects	262
8.	Πίστα (Room).....	263
9.	Sounds	264
10.	Paths	264

11. Scripts.....	264
12. Fonts.....	264
13. Time Lines.....	264
B7.2.3 Συμβάντα (Events) και Ενέργειες (Actions)	269
1. Εισαγωγή.....	269
2. Συμβάντα (Events)	269
3. Ενέργειες (Actions).....	270
B7.2.4 Ζωές, Σκορ και ο Ελεγκτής του παιχνιδιού	283
1. Εισαγωγή.....	283
2. Υλοποίηση του ελεγκτή (controller) του παιχνιδιού	283
B7.2.5 Δομή διακλάδωσης	289
1. Εισαγωγή.....	289
2. Η δομή διακλάδωσης.....	289
B7.2.6 Προγραμματισμός με GML	299
1. Εισαγωγή.....	299
2. Προγραμματίζοντας με τη GameMaker Language (GML)	299
B7.2.7 Η διαδικασία Ανάπτυξης Ενός Ηλεκτρονικού Παιχνιδιού	307
1. Η διαδικασία ανάπτυξης ενός ηλεκτρονικού παιχνιδιού	307
ΠΑΡΑΡΤΗΜΑ Α. Λογικά Διαγράμματα και Προκαταρτική Εκτέλεση	309

Εισαγωγή

Όταν τον Μάρτιο του 2009 ξεκίνησε η προσπάθεια για εκπόνηση του Νέου Αναλυτικού Προγράμματος του μαθήματος της Πληροφορικής και της Επιστήμης των Ηλεκτρονικών Υπολογιστών, τέθηκε ως προτεραιότητα η θεώρηση του μαθήματος μέσα από το πρίσμα των τριών πυλώνων της εκπαιδευτικής μεταρρύθμισης. Μέσα από το πρίσμα, δηλαδή, της απόκτησης ενός επαρκούς σώματος γνώσεων από τους μαθητές και τις μαθήτριες, την καλλιέργεια αξιών και την εκδήλωση συμπεριφορών που να συνάδουν με τη σύγχρονη έννοια της δημοκρατικής πολιτότητας και την καλλιέργεια κομβικών ικανοτήτων και δεξιοτήτων από τους μαθητές και τις μαθήτριες, ώστε να μπορούν να λειτουργούν στη διαμορφούμενη κοινωνία του 21^{ου} αιώνα.

Στην επιτροπή για την εκπόνηση του Αναλυτικού Προγράμματος τότε συμμετείχαν ο ακαδημαϊκός Ανδρέας Ανδρέου (Αναπληρωτής Καθηγητής στο Τεχνολογικό Πανεπιστήμιο Κύπρου), οι επιθεωρητές Πληροφορικής Μάριος Μιλτιάδου και Μιχάλης Τορτούρης και οι καθηγητές Πληροφορικής Νικόλαος Ζάγγουλος, Ιωάννης Ιωάννου, Μηνάς Καραολής, Θεόδουλος Κωνσταντίνου, Μαρία Μαρδά, Σωκράτης Μυλωνάς και Παναγιώτης Παπέττας. Παρόλο που είχε προταθεί ένα ολοκληρωμένο αναλυτικό πρόγραμμα από την Α΄ Γυμνασίου μέχρι και την Γ΄ Λυκείου, σε πρώτη φάση εφαρμόστηκε μόνο μέχρι την Γ΄ Γυμνασίου.

Το 2014 η διαδικασία εφαρμογής του Νέου Αναλυτικού Προγράμματος αναστάλθηκε για ένα έτος με σκοπό την αξιολόγησή του. Η ανεξάρτητη επιτροπή αξιολόγησης η οποία εξέτασε το περιεχόμενο του Νέου Αναλυτικού Προγράμματος για το μάθημα της Πληροφορικής και της Επιστήμης των Ηλεκτρονικών Υπολογιστών δεν εντόπισε παραλείψεις αλλά επισήμανε ότι αυτό ήταν βασισμένο σε διεθνή πρότυπα και ότι ήταν επιστημονικά επαρκές.

Τον Φεβρουάριο του 2015 διορίστηκε νέα επιτροπή για τη βελτίωση του Νέου Αναλυτικού Προγράμματος και την αναδιτύπωση του με συγκεκριμένο τρόπο (με τη μορφή δεικτών επιτυχίας και δεικτών επάρκειας), καθώς και την ετοιμασία Αναλυτικού Προγράμματος για τα μαθήματα που θα διδάσκονται στο Λύκειο, λαμβάνοντας υπόψη και τις παραμέτρους του Νέου Ωρολογίου Προγράμματος. Στην επιτροπή αυτή συμμετείχαν οι ακαδημαϊκοί Δρ Χρύσης Γεωργίου (Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής του Πανεπιστημίου Κύπρου), η Δρ Χαρούλα Αγγελή (Αναπληρώτρια Καθηγήτρια, Τμήμα Επιστημών της Αγωγής του Πανεπιστημίου Κύπρου) και Δρ Ανδρέας Ανδρέου (Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής του Τεχνολογικού Πανεπιστημίου Κύπρου), οι Επιθεωρητές Πληροφορικής Μάριος Μιλτιάδου και Μιχάλης Τορτούρης, οι Βοηθοί Διευθυντές Πληροφορικής Σωκράτης Μυλωνάς και Νικόλαος Ζάγγουλος και οι Καθηγητές/τριες Πληροφορικής, Παύλος Παυλικκάς, Πόλα Μάκκουλα, Παναγιώτης Ηρακλέους, Κωνσταντίνος Σωφρονίου, Μιλτιάδης Χαριλάου, Ξένιος Ξενοφώντος, Σοφία Καζέλη, Ηλίας Θεοδώρου, Δημήτρης Μαυροβουνιώτης, Δημήτρης Χατζηπαντελής, Νικόλας Στρατής και Θωμάς Ιωάννου.

Η διατύπωση του Αναλυτικού Προγράμματος για τη Β και τη Γ Λυκείου ολοκληρώθηκε τον Ιούλιο του 2016. Παράλληλα αναπτύχθηκε και το υποστηρικτικό υλικό (έντυπο και ηλεκτρονικό) το οποίο έχετε στα χέρια σας.

Σε επίπεδο περιεχομένου, το Νέο Αναλυτικό Πρόγραμμα περιέχει σημαντικές αλλαγές από το αντίστοιχο παλαιότερο, αλλά δεν περιέχει μεγάλο αριθμό νέων θεμάτων. Τόσο το Νέο Αναλυτικό Πρόγραμμα, όσο και το υποστηρικτικό υλικό είναι οργανωμένο με βάση επτά ενότητες, οι οποίες αναπτύσσονται παράλληλα από τη μία τάξη στην επόμενη και αντικατοπτρίζουν βασικούς θεματικούς άξονες της Πληροφορικής και Επιστήμης των Ηλεκτρονικών Υπολογιστών:

- Βασικές Έννοιες
- Το Υλικό/Αρχιτεκτονική Υπολογιστών
- Λειτουργικά Συστήματα
- Λογισμικό Εφαρμογών
- Δίκτυα και Διαδίκτυο
- Βάσεις Δεδομένων και Ανάπτυξη Πληροφοριακών Συστημάτων
- Αλγοριθμική Σκέψη, Προγραμματισμός και Σύγχρονες Εφαρμογές Πληροφορικής.

Στα μαθήματα της Β' και Γ' Λυκείου υπάρχει εξειδίκευση στα προσφερόμενα μαθήματα. Στο μάθημα *Πληροφορική και Επιστήμη Ηλεκτρονικών Υπολογιστών* δίνεται έμφαση στις Βασικές Έννοιες, στην Αρχιτεκτονική Υπολογιστών, στα Λειτουργικά Συστήματα, στις Βάσεις Δεδομένων και στην Ανάπτυξη Πληροφοριακών Συστημάτων και κυρίως στην Αλγοριθμική Σκέψη, τον Προγραμματισμό και στις Σύγχρονες Εφαρμογές Πληροφορικής όπου παρατηρούνται οι κύριες αλλαγές, με την εισαγωγή της γλώσσας προγραμματισμού C++ και τις ενότητες για ανάπτυξη παιγνιδιών και εφαρμογών για φορητές συσκευές. Το Λογισμικό Εφαρμογών και τα Δίκτυα Ηλεκτρονικών Υπολογιστών αναπτύσσονται σε άλλα εξειδικευμένα μαθήματα του κλάδου της Πληροφορικής και της Επιστήμης των Ηλεκτρονικών Υπολογιστών.

Οι ενότητες αυτές αναπτύσσονται σταδιακά μέσα από διαβαθμισμένους στόχους/δείκτες επιτυχίας από τη μία τάξη στην επόμενη, παρέχοντας επιπρόσθετες ευκαιρίες στους/στις μαθητές/τριες που δεν μπόρεσαν να τους κατακτήσουν σε μία τάξη να το πράξουν σε επόμενη, με παράλληλη εμβάθυνση. Σημαντικό στοιχείο της νέας προσέγγισης είναι και η αλληλεξάρτηση των ενοτήτων, αλλά και η συγκεκριμενοποίηση και ο περιορισμός των εννοιών που θα πρέπει να γνωρίζουν οι μαθητές/τριες, με έμφαση στην εφαρμογή τους στην πράξη.

Οι σημειώσεις και οι δραστηριότητες που ακολουθούν ετοιμάστηκαν από τους υποστηρικτές της εφαρμογής του νέου αναλυτικού προγράμματος και από τους συμβούλους της Πληροφορικής και της Επιστήμης των Ηλεκτρονικών Υπολογιστών. Μέσα από αυτές παρέχεται καθοδήγηση για το εύρος και το βάθος των γνώσεων και των δεξιοτήτων που χρειάζεται να αναπτύξουν οι μαθητές/τριες. Τονίζεται ότι οι σημειώσεις δεν αποτελούν διδακτικό εγχειρίδιο, αλλά αποτελούν σημείο αναφοράς περιλαμβάνοντας γνώσεις, επεξήγηση και παραδείγματα. Οι καθηγητές/τριες ενθαρρύνονται να στηρίξουν τη διδασκαλία τους στις σημειώσεις αυτές, αποφεύγοντας τη συστηματική αποστήθιση και ανάκληση ορισμών, αλλά με στόχο την εκμάθηση των βασικών γνώσεων και την εφαρμογή τους από τους μαθητές/τις μαθήτριες.

Οι δραστηριότητες στο τέλος κάθε κεφαλαίου είναι διαβαθμισμένες, ώστε να δίνεται η ευκαιρία εφαρμογής της νέας γνώσης σε όλους/ες τους/τις μαθητές/τριες ανάλογα με το βαθμό ετοιμότητάς τους, αρχίζοντας από το απλό και προχωρώντας στο σύνθετο. Μέσα από αυτές παρέχεται καθοδήγηση τόσο για το εύρος και το βάθος των γνώσεων και των δεξιοτήτων που χρειάζεται να αναπτύξουν οι μαθητές/τριες, όσο και για τις προτεινόμενες μεθοδολογικές προσεγγίσεις. Έμφαση δίνεται στην πρακτική εφαρμογή των γνώσεων και των δεξιοτήτων και στη χρήση εργαλείων και λογισμικού από τους/τις μαθητές/τριες στους ηλεκτρονικούς υπολογιστές, προάγοντας μαθητοκεντρικές προσεγγίσεις και περιορίζοντας τη μετωπική διδασκαλία. Οι καθηγητές/τριες ενθαρρύνονται να τα προσαρμόσουν ή και να δημιουργήσουν και άλλες δραστηριότητες, οι οποίες να ανταποκρίνονται καλύτερα, τόσο στα ιδιαίτερα χαρακτηριστικά των μαθητών/τριών τους, όσο και στη δική τους διδακτική προσέγγιση.

ΕΝΟΤΗΤΑ Β1 Βασικές Έννοιες της Πληροφορικής και της Επιστήμης Ηλεκτρονικών Υπολογιστών

B1.1 Αναπαράσταση Δεδομένων

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ❖ Να αναφέρουμε τον τρόπο αναπαράστασης των δεδομένων (δυναμικό σύστημα)
- ❖ Να αναγνωρίζουμε πώς γράμματα και σύμβολα από το πληκτρολόγιο αντιστοιχούν σε δυαδικούς αριθμούς (με τη βοήθεια του πίνακα ASCII)
- ❖ Να αναγνωρίζουμε πώς ακέραιοι αριθμοί αναπαρίστανται στο δυαδικό σύστημα και να μετατρέπουμε δυαδικούς αριθμούς σε δεκαδικούς
- ❖ Να αναγνωρίζουμε πώς ακέραιοι αριθμοί αναπαρίστανται στο δεκαεξαδικό σύστημα και να μετατρέπουμε δεκαεξαδικούς αριθμούς σε δεκαδικούς
- ❖ Να αναγνωρίζουμε πώς γράμματα και σύμβολα από το πληκτρολόγιο αντιστοιχούν σε δεκαεξαδικούς αριθμούς (με τη βοήθεια του πίνακα UNICODE).

1. Αναπαράσταση Δεδομένων – Δυναμικό Σύστημα

Η γλώσσα του ηλεκτρονικού υπολογιστή αποτελείται από μόνο δύο σύμβολα, το 0 και το 1, τα οποία ονομάζονται δυαδικά ψηφία - **bits** (binary digits). Το δυαδικό ψηφίο 1 αντιπροσωπεύει την ύπαρξη μίας ψηλότερης τάσης ρεύματος, ενώ το δυαδικό ψηφίο 0 αντιπροσωπεύει μία χαμηλότερη τάση ρεύματος. Το bit είναι η μικρότερη ποσότητα πληροφορίας που μπορεί να διαχειριστεί ο ηλεκτρονικός υπολογιστής. Για την παράσταση ενός χαρακτήρα στον υπολογιστή απαιτούνται, συνήθως, 8 bits τα οποία αποτελούν και τη στοιχειώδη μονάδα αποθήκευσης, τη ψηφιολέξη (**byte**).

2. Αναπαράσταση Χαρακτήρων με Δυναμικά Ψηφία

Οι περισσότερες από τις πληροφορίες που χρειάζεται να αποθηκεύσουμε είναι, κυρίως, **χαρακτήρες**, δηλαδή, γράμματα και σύμβολα από το πληκτρολόγιο. Αυτοί, όπως μάθαμε προηγουμένως, αναπαρίστανται στο εσωτερικό του υπολογιστή με δυαδικά ψηφία, 0 και 1, ως εξής:

- (1) Κάθε χαρακτήρας αποτελείται από έναν μοναδικό συνδυασμό 8 δυαδικών ψηφίων, για να ξεχωρίζει από τους υπόλοιπους χαρακτήρες. Με 8 δυαδικά ψηφία μπορούν να δημιουργηθούν $2^8 = 256$ διαφορετικοί συνδυασμοί ψηφίων, ο καθένας από τους οποίους αντιστοιχεί με ένα διαφορετικό γράμμα ή σύμβολο.
- (2) Ένας διαφορετικός συνδυασμός (κωδικός) αντιστοιχεί σε καθένα από τα γράμματα του αγγλικού αλφαβήτου (κεφαλαία και μικρά), τα 10 αριθμητικά ψηφία (0 - 9) και τους άλλους ειδικούς χαρακτήρες (+, -, ?, @ κ.α.).
- (3) Οι πληροφορίες αναπαρίστανται στον υπολογιστή σαν μία σειρά από τέτοιους οκταψήφιους δυαδικούς κωδικούς.
- (4) Οι περισσότερες σύγχρονες κωδικοποιήσεις χαρακτήρων βασίζονται στον Πίνακα Κωδικοποίησης ASCII (American Standard Code for Information Interchange).

3. Αναπαράσταση Αριθμών με Δυναμικά Ψηφία

Στον πραγματικό κόσμο είμαστε συνηθισμένοι να χρησιμοποιούμε το δεκαδικό σύστημα αρίθμησης, το οποίο αποτελείται από δέκα αριθμητικά ψηφία (0,1,2,3,...,9) και συνδυασμούς αυτών. Οι υπολογιστές, όμως, όπως έχουμε αναφέρει και προηγουμένως, αναγνωρίζουν μόνο τα ψηφία 0 και 1, έτσι χρησιμοποιούν το δυαδικό σύστημα αρίθμησης για ό,τι έχει να κάνει με αριθμούς και πράξεις. Σε αυτό, όλοι οι αριθμοί (όπως και οι χαρακτήρες) γράφονται χρησιμοποιώντας μόνο τα ψηφία 0 και 1. Σε αντίθεση, όμως, με τους χαρακτήρες, οι οποίοι είναι συγκεκριμένοι και μπορούν να αναπαρασταθούν με συγκεκριμένο αριθμό bit, οι αριθμοί

είναι άπειροι, έτσι χρειάζονται άπειρα bit για την αναπαράστασή τους. Στην πράξη, όμως, είναι σπάνιες οι περιπτώσεις όπου απαιτείται η χρήση είτε εξαιρετικά μεγάλων αριθμών είτε αριθμών με μεγάλη ακρίβεια (πολλά δεκαδικά ψηφία). Έτσι προσφέρονται διαφορετικές κατηγορίες αριθμών με διαφορετικό αριθμό bit η καθεμιά, ανάλογα με τις ανάγκες του προβλήματος. Οι δύο βασικές κατηγορίες είναι οι ακέραιοι αριθμοί και οι πραγματικοί αριθμοί. Για την αναπαράσταση ακεραίων αριθμών συνήθως χρησιμοποιούνται 8, 16, 32 ή 64 bit (1, 2, 4 ή 8 byte αντίστοιχα). Με 8 bit μπορούν να αναπαρασταθούν οι ακέραιοι αριθμοί από το 0 μέχρι το 2^8-1 , αλλά συνήθως ένα bit χρησιμοποιείται για να δηλώνει το πρόσημο του αριθμού (0=θετικός, 1=αρνητικός), με αποτέλεσμα να μπορούν να αναπαρασταθούν μόνο οι ακέραιοι αριθμοί από το $-2^7=-128$ μέχρι το $2^7-1=+127$. Με 16 bit διευρύνεται το μέγεθος των αριθμών από $-2^{15}=-32768$ μέχρι $2^{15}-1=+32767$, ενώ με 32 bit έχουμε αριθμούς από $-2^{31}=-2147483648$ μέχρι $2^{31}-1=+2147483647$ και με 64 bit από $-2^{63}=-9223372036854775808$ μέχρι $2^{63}-1=9223372036854775807$. Στις περισσότερες περιπτώσεις σήμερα χρησιμοποιούνται 32 bit, τα οποία μπορούν να αναπαριστούν αριθμούς ικανοποιητικού μεγέθους για τα περισσότερα προβλήματα.

Η αναπαράσταση πραγματικών αριθμών είναι πιο πολύπλοκη και δεν θα την περιγράψουμε με ακρίβεια, αλλά με ένα ανάλογο παράδειγμα στο δεκαδικό σύστημα. Έστω ότι θέλουμε να αναπαραστήσουμε τον αριθμό 12.3456. Ο αριθμός αυτός μπορεί να γραφεί ως 123456×10^{-4} , δηλαδή ως ένας ακέραιος αριθμός (x) επί κάποια δύναμη του 10 με ακέραιο εκθέτη (d). Τόσο το x όσο και το d μπορούν να αναπαρασταθούν από τον ηλεκτρονικό υπολογιστή ως ακέραιοι αριθμοί με περιορισμένο αριθμό bit, ανάλογα με τις ανάγκες του προβλήματος (στον ηλεκτρονικό υπολογιστή ο πραγματικός αριθμός εκφράζεται ως ένας ακέραιος αριθμός x επί κάποια δύναμη του 2 με ακέραιο εκθέτη d).

Παραδείγματα:

1. Να μετατραπεί ο δυαδικός αριθμός $(11010101)_2$ στο δεκαδικό σύστημα αρίθμησης.

$$(11010101)_2 = 1 \times \underline{128} + 1 \times \underline{64} + 1 \times \underline{16} + 1 \times \underline{4} + 1 \times \underline{1} = 213$$

$$(11010101)_2 = (213)_{10}$$

2. Να μετατραπεί ο αριθμός $(30)_{10}$ σε δυαδικό σύστημα αρίθμησης:

Βήμα	Διαίρεση	Υπόλοιπο	Δυαδική μορφή							
1	$30/2=15$	0								0
2	$15/2=7$	1							1	1
3	$7/2=3$	1						1	1	1
4	$3/2=1$	1					1	1	1	1
5	$1/2=0$	1				1	1	1	1	1
6	Συμπληρώνω με 0		0	0	0	1	1	1	1	0

$$(30)_{10} = (00011110)_2$$

4. Αναπαράσταση Χαρακτήρων με Δεκαεξαδικά Ψηφία

Για την αναπαράστασή του, το δεκαεξαδικό σύστημα χρησιμοποιεί 16 ψηφία. Για τα πρώτα δέκα χρησιμοποιούνται τα ψηφία 0 - 9 του δεκαδικού συστήματος. Για την αναπαράσταση

των τιμών από το 10 έως και το 15, χρησιμοποιεί τα πρώτα 6 κεφαλαία γράμματα του λατινικού αλφαβήτου: A, B, C, D, E και F.

Παραδείγματα:

1. Να μετατραπεί ο δεκαεξαδικός αριθμός $(23B)_{16}$ στο δεκαδικό σύστημα αρίθμησης.

$$(23B)_{16} = 2 \times \underline{256} + 3 \times \underline{16} + 11 \times \underline{1} = 571$$

2. Να μετατραπεί ο δεκαεξαδικός αριθμός $(ABCD)_{16}$ στο δεκαδικό σύστημα αρίθμησης.

$$(ABCD)_{16} = 10 \times \underline{4096} + 11 \times \underline{256} + 12 \times \underline{16} + 13 \times \underline{1} = 43981$$

5. Διεθνές Πρότυπο Unicode

Το διεθνές πρότυπο Unicode επιτρέπει την κωδικοποίηση όλων των συστημάτων γραφής, ώστε να επιτυγχάνεται η αποθήκευση, στη μνήμη ενός υπολογιστή, γραπτού κειμένου σε όλες τις γλώσσες, συμπεριλαμβανομένων και συμβόλων επιστημών, όπως μαθηματικά, φυσική κ.λπ. Το πρότυπο Unicode κωδικοποιεί οποιονδήποτε χαρακτήρα με μοναδικό τρόπο, επιτρέποντας να υπάρχουν στο ίδιο κείμενο χαρακτήρες από διάφορες γλώσσες, το οποίο εξασφαλίζει ότι όλοι οι χαρακτήρες μπορούν να υποστηρίζονται από όλα τα υπολογιστικά συστήματα που το υιοθετούν. Επιπρόσθετα, επιτρέπει στο ανάλογο λογισμικό (φυλλομετρητής, επεξεργαστής κειμένου) να επιλέξει αυτό την οπτική αναπαράσταση (στυλ, μέγεθος, γραμματοσειρά) των χαρακτήρων.

5.1 Το πρότυπο

Η κοινοπραξία Unicode, που αναπτύσσει το πρότυπο, επιτρέπει σε οποιαδήποτε οργάνωση να γίνει μέλος της εφόσον πληρώσει συνδρομή. Στα μέλη της συμπεριλαμβάνονται σχεδόν όλες οι μεγάλες εταιρείες λογισμικού και υλικού που όπως οι Apple, Microsoft, IBM, Xerox, HP, Adobe Systems και πολλές άλλες. Η κοινοπραξία δημοσίευσε για πρώτη φορά το πρότυπο Unicode το 1991 και συνεχίζει να αναπτύσσει πρότυπα βασισμένα στην αρχική αυτή εργασία. Η κοινοπραξία Unicode αναπτύχθηκε σε συνδυασμό με τον Διεθνή Οργανισμό Τυποποίησης ISO, και το πρότυπό της μοιράζεται το σύνολο χαρακτήρων της με το πρότυπο ISO/IEC 10646. Το Unicode και το ISO/IEC 10646 είναι ισοδύναμα ως κωδικοποιήσεις χαρακτήρων, αλλά το Unicode περιέχει πολύ περισσότερες πληροφορίες για προγραμματιστές. Η πιο πρόσφατη έκδοση (Unicode 9.0) περιλαμβάνει 128,172 χαρακτήρες και 72 νέα emoji.

5.2 Αναπαράσταση χαρακτήρων

Στο σύστημα Unicode, κάθε χαρακτήρας έχει τη δική του μοναδική αναπαράσταση ως μίας σειράς από bit και αναφέρεται ως κωδικό σημείο (codepoint). Συχνά ο τρόπος αναφοράς εκφράζεται με τη μορφή U+xxxx, όπου τα xxxx αναπαριστούν τη σειρά από bit ως δεκαεξαδικά ψηφία. Για παράδειγμα, το πεζό λατινικό γράμμα b εκφράζεται ως U+0062 ενώ το κεφαλαίο ελληνικό γράμμα Ω εκφράζεται ως U+03A9. Συνήθως αναγράφονται 4 δεκαεξαδικά ψηφία, τα οποία αντιστοιχούν με 16 bit, αλλά υπάρχουν εξαιρέσεις για κάποιες κατηγορίες χαρακτήρων, όπως το μουσικό σύμβολο B^\flat το οποίο εκφράζεται ως U+1D122 και έχει 5 δεκαεξαδικά ψηφία.

5.3 Αποθήκευση, μεταφορά και επεξεργασία

Το λογισμικό που χρησιμοποιείται στον δυτικό κόσμο χειρίζεται μόνο κωδικοποιήσεις 8-bit. Είναι αδύνατη η χρήση περισσότερων από 256 κωδικών σημείων χωρίς ειδική επεξεργασία. Η αναπαράσταση των χαρακτήρων στο σύστημα Unicode επιλύει κάποια από αυτά τα προβλήματα, αλλά συνοδεύεται με νέες προκλήσεις:

- Συμβατότητα με δεδομένα που έχουν συλλεγεί για χρόνια και είναι κωδικοποιημένα σε ASCII.
- Σημαντική αύξηση των απαιτήσεων για αποθήκευση χαρακτήρων κωδικοποιημένων σε Unicode, αφού χρειάζονται τουλάχιστο 16 bit, αντί 8 bit του ASCII, ιδιαίτερα εάν το κείμενο αποτελείται κατά κύριο λόγο από λατινικούς χαρακτήρες, όπου τα 8 bit σε ASCII θα ήταν καλύτερη επιλογή, αφού εξοικονομεί τον μισό χώρο για αποθήκευση των πληροφοριών.

Για να αντιμετωπίσει τις προκλήσεις αυτές, το σύστημα Unicode ορίζει δύο τρόπους αναπαράστασης και αποθήκευσης των χαρακτήρων. Πρόκειται για πρότυπα, ή καλύτερα οικογένειες προτύπων. Αυτοί είναι οι ακόλουθοι:

- Κωδικοποιήσεις UTF (Unicode Transformation Format).
- Κωδικοποιήσεις UCS (Universal Character Set).

Οι τρόποι απεικόνισης περιλαμβάνουν τις εξής κύριες κωδικοποιήσεις: UTF-32, UCS-4, UTF-16, UCS-2, UTF-8, UTF-EBCDIC.

Ο αριθμός μετά από την οικογένεια UTF αναφέρεται στον αριθμό των bit για τη βασική κωδικοποίηση (π.χ. η UTF-16 χρησιμοποιεί 16 bit), ενώ για την οικογένεια UCS εκφράζει τον αριθμό των byte (π.χ. η UCS-2 χρησιμοποιεί 2 byte=16 bit). Η κύρια διαφορά των δύο οικογενειών είναι ότι η UTF επιτρέπει την κωδικοποίηση με πολλαπλάσια της βασικής μονάδας ενώ η UCS έχει συγκεκριμένο μέγεθος.

Παράδειγμα:

Στο UTF-8, η αναπαράσταση του πεζού λατινικού γράμματος b (U+0062) χρειάζεται 8 bit, ενώ του κεφαλαίου ελληνικού γράμματος Ω (U+03A9) χρειάζεται 16 bit και του μουσικού συμβόλου ♪ (U+1D122) χρειάζεται 32 bit.

Στο UTF-16, η αναπαράσταση του πεζού λατινικού γράμματος b (U+0062) χρειάζεται 16 bit, όπως και του κεφαλαίου ελληνικού γράμματος Ω (U+03A9), ενώ του μουσικού συμβόλου ♪ (U+1D122) χρειάζεται 32 bit.

Στο UCS-2, η αναπαράσταση του πεζού λατινικού γράμματος b (U+0062) χρειάζεται 16 bit, όπως και του κεφαλαίου ελληνικού γράμματος Ω (U+03A9), ενώ το μουσικό σύμβολο ♪ (U+1D122) δε μπορεί να αναπαρασταθεί παρά μόνο στο UCS-4.

Η μέθοδος κωδικοποίησης UCS είναι παλαιότερη από την UTF, η οποία προσφέρει περισσότερα πλεονεκτήματα και σταδιακά εγκαταλείπεται. Για παράδειγμα, η UTF-16 παρέχει τα ίδια πλεονεκτήματα και είναι αντίστοιχη με την UCS-2 (για κωδικά σημεία μέχρι 16 bit), αλλά είναι παράλληλα πιο ευέλικτη.

Η UTF-8 παρέχει αρκετά πλεονεκτήματα και είναι η πιο συχνή κωδικοποίηση πληροφοριών στο Διαδίκτυο μετά από την ASCII. Τα κύρια πλεονεκτήματά της είναι η συμβατότητα της με την κωδικοποίηση ASCII (το περιεχόμενο ενός αρχείου με UTF-8 κωδικοποίηση το οποίο περιέχει μόνο κωδικά σημεία από το διάστημα του Unicode U+0000 μέχρι U+007F, τα οποία αντιστοιχούν στους 128 ASCII χαρακτήρες είναι ακριβώς το ίδιο με το αρχείο κωδικοποιημένο σε ASCII) καθώς και η εξοικονόμηση χώρου, αφού οι περισσότερες πληροφορίες γενικά αποτελούνται κυρίως από σύμβολα από το διάστημα του Unicode U+0000 μέχρι U+007F τα οποία χρειάζονται μόνο 8 bit για αποθήκευση.

Το κύριο μειονέκτημα της κωδικοποίησης UTF-8 είναι η επιπρόσθετη (πιο σύνθετη) επεξεργασία για αποκωδικοποίηση, αφού ο αριθμός των bit διαφέρει από χαρακτήρα σε χαρακτήρα. Όπου αυτό αποτελεί σοβαρό πρόβλημα, μπορεί να χρησιμοποιηθεί κάποιος εναλλακτικός τρόπος κωδικοποίησης, π.χ. UTF-16.

Στον πίνακα που ακολουθεί δίνονται παραδείγματα χαρακτήρων και η αντίστοιχη κωδικοποίηση σε ASCII και σε UTF-16 (UCS-2). Η κωδικοποίηση με UTF-8 θα ήταν πανομοιότυπη με την ASCII για τους χαρακτήρες που υποστηρίζονται, αλλά θα είχε διαφορετική δυαδική μορφή για τους υπόλοιπους χαρακτήρες (π.χ. το Ω σε UTF-8 κωδικοποιείται ως 11001110 10101001).

Πίνακας Κωδικοποίησης ASCII και Unicode

Χαρακτήρας	Περιγραφή	ASCII	UNICODE	UTF-16/UCS-2
0	Ψηφίο Μηδέν	00110000	U+0030	00000000 00110000
1	Ψηφίο Ένα	00110001	U+0031	00000000 00110001
2	Ψηφίο Δύο	00110010	U+0032	00000000 00110010
3	Ψηφίο Τρία	00110011	U+0033	00000000 00110011
4	Ψηφίο Τέσσερα	00110100	U+0034	00000000 00110100
5	Ψηφίο Πέντε	00110101	U+0035	00000000 00110101
6	Ψηφίο Έξι	00110110	U+0036	00000000 00110110
7	Ψηφίο Επτά	00110111	U+0037	00000000 00110111
8	Ψηφίο Οκτώ	00111000	U+0038	00000000 00111000
9	Ψηφίο Εννέα	00111001	U+0039	00000000 00111001
A	Κεφαλαίο Α (Λατ.)	01000001	U+0041	00000000 01000001
B	Κεφαλαίο Β (Λατ.)	01000010	U+0042	00000000 01000010
C	Κεφαλαίο C (Λατ.)	01000011	U+0043	00000000 01000011
D	Κεφαλαίο D (Λατ.)	01000100	U+0044	00000000 01000100
E	Κεφαλαίο E (Λατ.)	01000101	U+0045	00000000 01000101
F	Κεφαλαίο F (Λατ.)	01000110	U+0046	00000000 01000110
G	Κεφαλαίο G (Λατ.)	01000111	U+0047	00000000 01000111
H	Κεφαλαίο H (Λατ.)	01001000	U+0048	00000000 01001000
I	Κεφαλαίο I (Λατ.)	01001001	U+0049	00000000 01001001
J	Κεφαλαίο J (Λατ.)	01001010	U+004A	00000000 01001010
K	Κεφαλαίο K (Λατ.)	01001011	U+004B	00000000 01001011
L	Κεφαλαίο L (Λατ.)	01001100	U+004C	00000000 01001100
M	Κεφαλαίο M (Λατ.)	01001101	U+004D	00000000 01001101
N	Κεφαλαίο N (Λατ.)	01001110	U+004E	00000000 01001110

Χαρακτήρας	Περιγραφή	ASCII	UNICODE	UTF-16/UCS-2
O	Κεφαλαίο O (Λατ.)	01001111	U+004F	00000000 01001111
P	Κεφαλαίο P (Λατ.)	01010000	U+0050	00000000 01010000
Q	Κεφαλαίο Q (Λατ.)	01010001	U+0051	00000000 01010001
R	Κεφαλαίο R (Λατ.)	01010010	U+0052	00000000 01010010
S	Κεφαλαίο S (Λατ.)	01010011	U+0053	00000000 01010011
T	Κεφαλαίο T (Λατ.)	01010100	U+0054	00000000 01010100
U	Κεφαλαίο U (Λατ.)	01010101	U+0055	00000000 01010101
V	Κεφαλαίο V (Λατ.)	01010110	U+0056	00000000 01010110
W	Κεφαλαίο W (Λατ.)	01010111	U+0057	00000000 01010111
X	Κεφαλαίο X (Λατ.)	01011000	U+0058	00000000 01011000
Y	Κεφαλαίο Y (Λατ.)	01011001	U+0059	00000000 01011001
Z	Κεφαλαίο Z (Λατ.)	01011010	U+005A	00000000 01011010
a	Πεζό a (Λατ.)	01100001	U+0061	00000000 01100001
b	Πεζό b (Λατ.)	01100010	U+0062	00000000 01100010
z	Πεζό z (Λατ.)	01111010	U+007A	00000000 01111010
A	Κεφαλαίο A (Ελλ.)	-	U+0391	00000011 10010001
B	Κεφαλαίο B (Ελλ.)	-	U+0392	00000011 10010010
Ω	Κεφαλαίο Ω (Ελλ.)	-	U+03A9	00000011 10101001
α	Πεζό α (Ελλ.)	-	U+03B1	00000011 10110001
β	Πεζό β (Ελλ.)	-	U+03B2	00000011 10110010
ω	Πεζό ω (Ελλ.)	-	U+03C9	00000011 11001001
ά	α με τόνο (Ελλ.)	-	U+03AC	00000011 10101100
ώ	ω με τόνο (Ελλ.)	-	U+03CE	00000011 11001110



Ασκήσεις Κεφαλαίου

Άσκηση 1.1

Να συμπληρώσετε τους αριθμούς ώστε να ισχύουν οι ισότητες:

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
$(24)_{10} =$	0	0	0		1	0		

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
$(65)_{10} =$	0	1	0		0	0		

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
$(193)_{10} =$				0	0	0		

Άσκηση 1.2

Να αντιστοιχίσετε τους δυαδικούς αριθμούς της **Στήλης Α** με τους δεκαδικούς αριθμούς της **Στήλης Β**.

Στήλη Α		Στήλη Β	
1.	00001010	α.	9
2.	00000011	β.	140
3.	00001001	γ.	128
4.	10000000	δ.	3
5.	01010000	ε.	10
6.	10001100	στ.	80

Άσκηση 1.3

Να μετατρέψετε τους πιο κάτω αριθμούς από το δεκαδικό στο δυαδικό σύστημα αρίθμησης.

(α) 123 (β) 50 (γ) 125 (δ) 138

Άσκηση 1.4

Να μετατρέψετε τους πιο κάτω αριθμούς από το δυαδικό στο δεκαδικό σύστημα αρίθμησης.

(α) 1010 (β) 10111010 (γ) 1110001 (δ) 11001101

Άσκηση 1.5

Να μετατρέψετε τους πιο κάτω αριθμούς από το δεκαεξαδικό στο δεκαδικό σύστημα αρίθμησης.

(α) 8AF2

(β) DC

(γ) 123AB

(δ) 3E4

Άσκηση 1.6

Γιατί δημιουργήθηκε το σύστημα Unicode; Να αναφέρετε τρεις κύριες κωδικοποιήσεις του.

Άσκηση 1.7

Μέχρι πόσα byte ανά χαρακτήρα επιτρέπει η απεικόνιση UTF-8; Πόσα byte χρειάζονται για την κωδικοποίηση και των 128 χαρακτήρων ASCII;

Άσκηση 1.8

Χρησιμοποιώντας το φυλλομετρητή Google Chrome, να μεταβείτε στην ιστοσελίδα <http://www.columbia.edu/~fdc/utf8/>. Από τις ρυθμίσεις, να επιλέξετε More tools->Encoding και αφού από-επιλέξετε το Auto-Detect στο πάνω μέρος, να επιλέξετε το Korean. Τι παρατηρείτε;

Άσκηση 1.9

Σε ένα νέο αρχείο κειμένου, να πληκτρολογήσετε τον κώδικα UNICODE ενός χαρακτήρα (π.χ. U+03CE). Στη συνέχεια να πατήσετε ταυτόχρονα τα πλήκτρα ALT και X για να τον μετατρέψετε στον χαρακτήρα που αντιστοιχεί. Να δοκιμάσετε και το αντίθετο, πληκτρολογώντας αρχικά έναν χαρακτήρα και ακολούθως πατώντας τα πλήκτρα ALT και X.

Άσκηση 1.10

Χρησιμοποιώντας κατάλληλο εργαλείο (π.χ. character_repr.html από το μαθησιακό υλικό) να κωδικοποιήσετε ένα σύντομο μήνυμα (μέχρι 20 χαρακτήρες) σε Unicode (UTF-16). Στη συνέχεια να ανταλλάξετε κωδικοποιημένα μηνύματα με κάποιον/α συμμαθητή/τρια σας και να αποκωδικοποιήσετε το μήνυμα. Να επιβεβαιώσετε ότι η κωδικοποίηση/αποκωδικοποίηση έγινε με ορθό τρόπο, συγκρίνοντας με το αρχικό μήνυμα.

ΕΝΟΤΗΤΑ Β6 Βάσεις Δεδομένων και Ανάπτυξη Πληροφοριακών Συστημάτων

B6.1 Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ❖ Να διακρίνουμε ανάμεσα στο σύστημα, υποσύστημα, το εσωτερικό και το εξωτερικό περιβάλλον ενός συστήματος σε παραδείγματα
- ❖ Να αναφέρουμε τα βασικά συστατικά στοιχεία ενός πληροφοριακού συστήματος και να τα διακρίνουμε σε παραδείγματα
- ❖ Να αναφέρουμε τις κατηγορίες Πληροφοριακών Συστημάτων
- ❖ Να εντοπίζουμε σε εργασιακά παραδείγματα την κατηγορία (τύπο) ενός πληροφοριακού συστήματος
- ❖ Να αναφέρουμε τις φάσεις του Κύκλου Ζωής και Ανάπτυξης ενός Πληροφοριακού Συστήματος
- ❖ Να αναφέρουμε και να διακρίνουμε ανάμεσα στα διάφορα μοντέλα ανάπτυξης ενός Πληροφοριακού Συστήματος
- ❖ Να εντοπίζουμε το μοντέλο ανάπτυξης του πληροφοριακού συστήματος σε εργασιακά παραδείγματα
- ❖ Να περιγράφουμε σύντομα τον ρόλο (τα καθήκοντα και τις ευθύνες) του αναλυτή συστημάτων και να αναφέρουμε τα αναγκαία του προσόντα
- ❖ Να περιγράφουμε σύντομα τον ρόλο άλλου ανθρώπινου δυναμικού που εμπλέκεται στην ανάπτυξη ενός πληροφοριακού συστήματος (Ο ιδιοκτήτης / πελάτης, οι χρήστες, οι προγραμματιστές, οι διαχειριστές βάσεων δεδομένων, οι τεχνικοί/διαχειριστές δικτύων, οι εκπαιδευτές)
- ❖ Να διακρίνουμε σε εργασιακά παραδείγματα ρόλους που επιτελεί ο αναλυτής και το άλλο ανθρώπινο δυναμικό.

1. Εισαγωγή

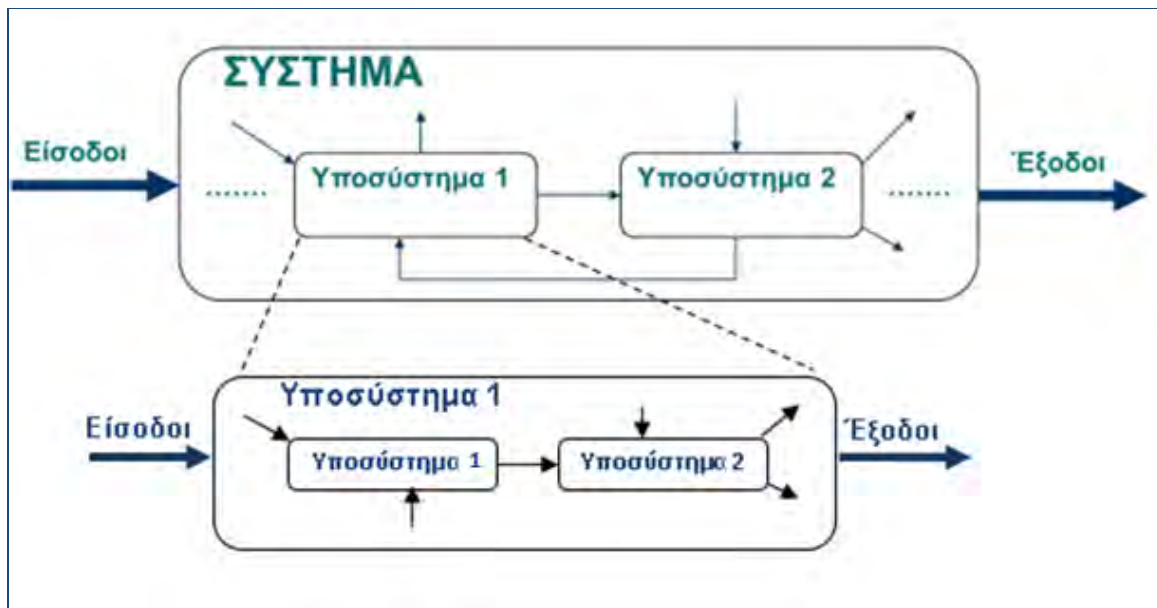
Η Ανάλυση και Σχεδίαση πληροφοριακών συστημάτων έχει να κάνει με την επιλογή, σχεδίαση και υλοποίηση κατάλληλων πληροφοριακών συστημάτων για την αντιμετώπιση των αναγκών και προβλημάτων ενός οργανισμού/επιχείρησης.

Οι σύγχρονες επιχειρήσεις/οργανισμοί στηρίζονται σε πολύ μεγάλο βαθμό στις τεχνολογίες πληροφορίας και επικοινωνιών και σε εξελιγμένα πληροφοριακά συστήματα, για να είναι ανταγωνιστικές και να προσφέρουν στην εγχώρια και διεθνή αγορά τα προϊόντα και τις υπηρεσίες τους. Για την ανάπτυξη ενός επιτυχημένου πληροφοριακού συστήματος επιβάλλεται η ενεργός συμμετοχή όλων των εργαζομένων στην επιχείρηση/οργανισμό και όχι μόνο των εμπειρογνομόνων της Πληροφορικής και των υπολογιστών.

2. Σύστημα, Υποσύστημα και Περιβάλλον

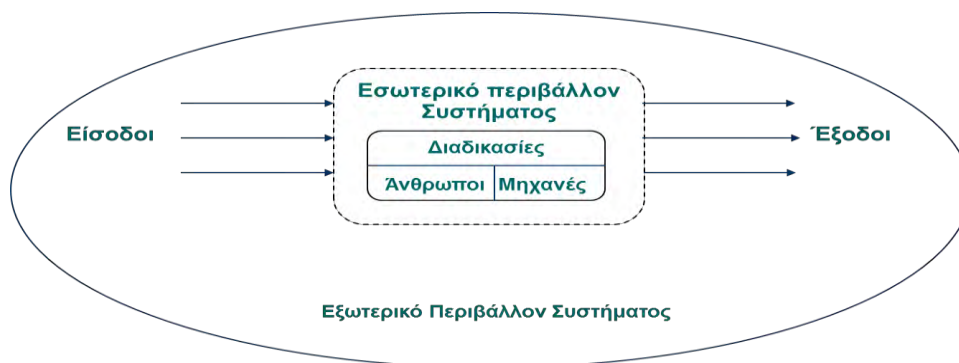
Σύστημα είναι ένα σύνολο συστατικών στοιχείων, άνθρωποι – μηχανές – διαδικασίες, που διέπονται από συγκεκριμένους τρόπους λειτουργίας και τα οποία επιδρούν μεταξύ τους για την διεκπεραίωση κάποιου έργου.

Κάθε σύστημα μπορεί να περιλαμβάνει έναν αριθμό υποσυστημάτων, που και αυτά με τη σειρά τους μπορούν να περιλαμβάνουν άλλα μικρότερα υποσυστήματα. Όλα αυτά τα υποσυστήματα συνδέονται και συνεργάζονται μεταξύ τους, ανταλλάζοντας δεδομένα για την επίτευξη των στόχων και των σκοπών του συστήματος το οποίο απαρτίζουν.



3. Εσωτερικό και Εξωτερικό Περιβάλλον Συστήματος

Κάθε σύστημα αποτελείται από δύο περιβάλλοντα, το εσωτερικό και το εξωτερικό. Τα συστατικά στοιχεία του συστήματος (άνθρωποι – μηχανές – διαδικασίες) καθορίζουν το εσωτερικό του περιβάλλον. Ό,τι βρίσκεται εκτός από το εσωτερικό περιβάλλον ονομάζεται εξωτερικό περιβάλλον. Αυτά τα δύο περιβάλλοντα βρίσκονται σε συνεχή επικοινωνία, ανταλλάσσοντας δεδομένα (Είσοδοι / Έξοδοι).



Παράδειγμα Συστήματος

Η γραμματεία ενός σχολείου είναι ένα σύστημα με τα πιο κάτω χαρακτηριστικά:

Εσωτερικό Περιβάλλον:

Άνθρωποι (γραμματείς, κλητήρες κ.λπ.)
 Διαδικασίες (Π.χ. εγγραφή μαθητών, διαχωρισμός τους σε τάξεις, ετοιμασία ενδεικτικών κ.λπ.)
 Μηχανές (Η.Υ., Φωτοτυπικές κ.λπ.)

Εξωτερικό Περιβάλλον:

Μαθητές, Γονείς, Υπουργείο Παιδείας κ.λπ.

Είσοδοι:

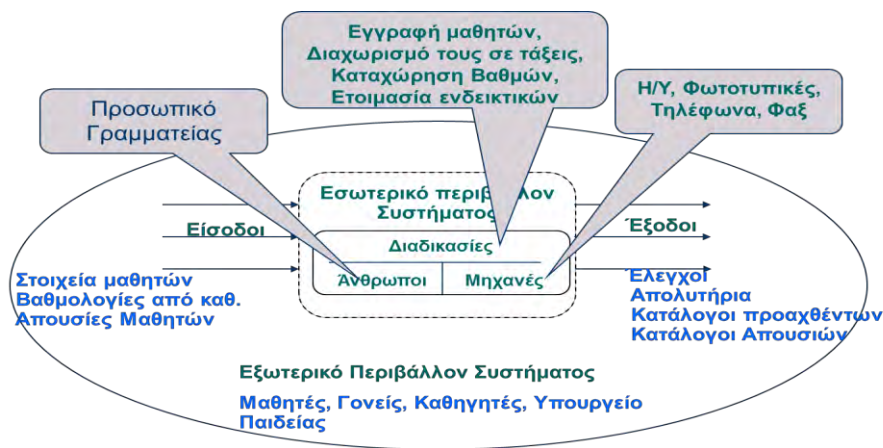
Στοιχεία μαθητών, Βαθμολογίες καθηγητών κ.λπ.

Έξοδοι:

Ενδεικτικοί έλεγχοι, Απολυτήρια, Βεβαιώσεις κ.λπ.

Υποσυστήματα:

Τμήμα δακτυλογράφησης, Τμήμα φωτοτυπιών, κ.λπ.



4. Πληροφοριακά Συστήματα (ΠΣ)

Πληροφοριακό Σύστημα (Information System) είναι ένα οργανωμένο σύνολο, το οποίο αποτελείται από:

- ανθρώπινο δυναμικό (το σύνολο των ανθρώπων που εργάζονται με το πληροφοριακό σύστημα σε διάφορους ρόλους όπως χρήστες, διαχειριστές κ.λπ.)
- διαδικασίες (το σύνολο των οδηγιών για τη χρήση και τον συνδυασμό όλων των στοιχείων υποδομής ενός ΠΣ)
- δεδομένα (βάση δεδομένων)
- τεχνολογία της Πληροφορικής - λογισμικό
- υλικό εξοπλισμό (Υπολογιστές, εκτυπωτές, servers, δίκτυα κ.λπ.)

Αυτό το οργανωμένο σύνολο αποσκοπεί στη συλλογή, στον έλεγχο, στην αποθήκευση, στη μετάδοση και στην επεξεργασία δεδομένων για την παροχή όπου αυτό είναι απαραίτητο, χρήσιμων, ολοκληρωμένων, έγκυρων και έγκαιρων πληροφοριών.

5. Πληροφοριακό Σύστημα

Ένα πληροφοριακό σύστημα μετατρέπει τα δεδομένα σε πληροφορίες.

- Δεδομένα (είσοδοι) είναι ακατέργαστα γεγονότα.
- Πληροφορίες (έξοδοι) είναι τα δεδομένα τα οποία έχουν υποστεί επεξεργασία, έτσι ώστε να γίνουν χρήσιμα σε κάποιον.

Το άτομο ή ο οργανισμός/επιχείρηση που πληρώνει για την ανάπτυξη ενός πληροφοριακού συστήματος ονομάζεται πελάτης, ενώ το άτομο ή το σύνολο των ατόμων που αλληλεπιδρούν άμεσα με το σύστημα ονομάζονται χρήστες. Πολλές φορές, χρήστες και πελάτες είναι το ίδιο πρόσωπο.

Όπως κάθε σύστημα έτσι και το πληροφοριακό σύστημα μπορεί να αποτελείται από έναν αριθμό υποσυστημάτων, τα οποία συνδέονται και συνεργάζονται μεταξύ τους για τη διεκπεραίωση του όλου έργου.



6. Τύποι πληροφοριακών συστημάτων

- **Πληροφοριακά συστήματα διοίκησης** (Management Information Systems - MIS). Βασικό εργαλείο για την οργάνωση και έλεγχο της διοικητικής λειτουργίας ενός οργανισμού/επιχείρησης.

Στην περίπτωση που ένα πληροφοριακό σύστημα διοίκησης καλύπτει και υποστηρίζει (αυτοματοποιεί) όλες τις λειτουργίες και δραστηριότητες ενός/μίας οργανισμού/επιχείρησης, χαρακτηρίζεται ως Ολοκληρωμένο Πληροφοριακό Σύστημα Διοίκησης (Integrated Management Information Systems - IMIS)

- **Συστήματα επεξεργασίας συναλλαγών** (Συστήματα τραπεζών, παραγγελιών, διαχείρισης αποθήκης, κράτησης αεροπορικών εισιτηρίων, κ.λπ.)
- **Συστήματα υποστήριξης αποφάσεων** (Σύστημα χρηματιστηρίου - επενδύσεων, σύστημα διανομής προϊόντων - καταλληλότερη πορεία, κ.λπ.)
- **Συστήματα εμπειρογνωμοσύνης** (Ιατρικά Συστήματα, κ.λπ.)
- **Συστήματα αυτοματοποίησης γραφείου** (Συστήματα επεξεργασία κειμένου, λογιστικών φύλλων, ηλεκτρονικού ημερολογίου κ.λπ.)

7. Κύκλος ζωής και ανάπτυξης πληροφοριακού συστήματος

Κάθε πληροφοριακό σύστημα, από τη σύλληψη μέχρι την απόσυρσή του, διέρχεται από διάφορες φάσεις, οι οποίες συνδέονται και εξαρτώνται μεταξύ τους. Σε καθεμιά από τις φάσεις αυτές, πρέπει να γίνονται ορισμένες εργασίες, ώστε να επιτυγχάνεται το επιθυμητό αποτέλεσμα. Οι πολύ γενικές φάσεις με την σειρά που λαμβάνουν χώρα είναι: σύλληψη, κατασκευή, χρήση/συντήρηση και απόσυρση.

8. Μοντέλα κύκλου ζωής και ανάπτυξης πληροφοριακών συστημάτων

Ένα μοντέλο κύκλου ζωής και ανάπτυξης του πληροφοριακού συστήματος είναι μία περιγραφή των δραστηριοτήτων και των επιμέρους φάσεων, από τις οποίες διέρχεται ένα πληροφοριακό σύστημα από τη σύλληψη μέχρι την απόσυρσή του, καθώς και των εργασιών που λαμβάνουν χώρα σε καθεμιά από τις φάσεις αυτές.

Οι δραστηριότητες (διαδικασίες) ανάπτυξης του πληροφοριακού συστήματος καθορίζουν ποιες ενέργειες πρέπει να γίνουν, για να επιτευχθεί ένα επιθυμητό αποτέλεσμα σε κάποια από τις φάσεις του κύκλου ζωής. Μία δραστηριότητα μπορεί να αναλύεται σε περισσότερες από μία επιμέρους φάσεις.

Η μεθοδολογία ανάπτυξης καθορίζει το πώς θα πρέπει να εκτελούνται οι δραστηριότητες ανάπτυξης του πληροφοριακού συστήματος, δηλαδή ποιες επιμέρους ενέργειες περιλαμβάνει, ποια βήματα γίνονται σε καθεμιά από τις φάσεις, ποια προϊόντα παράγονται, καθώς και πότε αυτές θεωρούνται περατωθείσες.

Υπάρχουν αρκετά μοντέλα κύκλου ζωής και ανάπτυξης πληροφοριακού συστήματος, τα οποία διαφοροποιούνται ως προς τη σύλληψη της ιδέας του τρόπου κατασκευής, αλλά και ως προς τις επιμέρους φάσεις που προτείνουν, την επαναληπτικότητα και την εμβέλεια των εργασιών αυτών, τα ενδιάμεσα προϊόντα – συστατικά λογισμικού και την περιγραφή τους, τις οικονομικές και επιχειρηματικές πλευρές της χρήσης τους κ.ά.

9. Διάφορα Μοντέλα:

- Το μοντέλο του καταρράκτη
- Το μοντέλο πρωτοτυποποίησης
- Το μοντέλο λειτουργικής επαύξησης
- Το σπειροειδές μοντέλο
- Σύγχρονα μοντέλα κύκλου ζωής λογισμικού

9.1 Μοντέλο του καταρράκτη

Ένα από τα πιο διαδεδομένα μοντέλα κύκλου ζωής και ανάπτυξης πληροφοριακού συστήματος είναι αυτό του καταρράκτη (waterfall). Η κεντρική ιδέα του μοντέλου του καταρράκτη είναι ότι το πληροφοριακό σύστημα αναπτύσσεται, περνώντας ολόκληρο από διαδοχικές επιμέρους φάσεις, καθεμιά από τις οποίες θεωρείται περατωμένη με την παραγωγή ορισμένων συστατικών του συστήματος. Κάθε επιμέρους φάση ολοκληρώνεται με μία εργασία επαλήθευσης/επικύρωσης των προϊόντων της, κατά την οποία αποφασίζεται η μετάβαση ή όχι στην επόμενη. Χαρακτηριστικό του μοντέλου του καταρράκτη είναι ότι, για να ξεκινήσει μία φάση, πρέπει να έχει ολοκληρωθεί πλήρως η προηγούμενη. Το σύστημα είναι πλήρες, δηλαδή περιλαμβάνει όλα τα λειτουργικά του χαρακτηριστικά μετά από την τελευταία φάση του συγκεκριμένου μοντέλου.

9.2 Μοντέλο πρωτοτυποποίησης

Η κεντρική ιδέα του μοντέλου πρωτοτυποποίησης (prototyping model) είναι η ανάπτυξη του πληροφοριακού συστήματος όχι εξολοκλήρου, αλλά σε τμήματα, που ονομάζονται «πρωτότυπα». Οι διαδικασίες ανάπτυξης επαναλαμβάνονται για ένα τμήμα του συστήματος κάθε φορά και, για το λόγο αυτό, το μοντέλο χαρακτηρίζεται ως επαναληπτικό. Κάθε πρωτότυπο περιλαμβάνει τις βασικές από τις λειτουργίες που προορίζεται να εκτελεί το σύστημα και τίθεται σε δοκιμασία από τον πελάτη. Από εκεί συλλέγονται παρατηρήσεις και η

διαδικασία κατασκευής νέου πρωτοτύπου επαναλαμβάνεται μέχρις ότου ένα πρωτότυπο να ικανοποιεί τις απαιτήσεις, δηλαδή να εκτελεί τις επιθυμητές λειτουργίες του συστήματος με τρόπο ικανοποιητικό και να γίνεται αποδεκτό από τον πελάτη. Από το σημείο αυτό και μετά μπορούν να προστεθούν και οι υπόλοιπες λειτουργίες, ώστε το σύστημα να ολοκληρωθεί.

9.3 Μοντέλο λειτουργικής επαύξησης

Το μοντέλο της λειτουργικής επαύξησης (incremental model) συνδυάζει την ακολουθιακή ανάπτυξη του μοντέλου του καταρράκτη με την τμηματική ανάπτυξη του μοντέλου της πρωτοτυποποίησης. Κεντρική ιδέα, είναι ο διαχωρισμός του υπό κατασκευή συστήματος σε τμήματα που αναπτύσσονται ανεξάρτητα, ακολουθώντας το καθένα ακολουθιακή ανάπτυξη σύμφωνα με το μοντέλο του καταρράκτη. Κατά την αρχική φάση ανάλυσης και σχεδίασης αποφασίζονται τα τμήματα στα οποία θα «διαχωριστεί» το σύστημα, η ανάπτυξη των οποίων γίνεται στη συνέχεια ανεξάρτητα και παράλληλα. Όταν ολοκληρώνεται η ανάπτυξη κάθε τμήματος, αυτό ενσωματώνεται στο σύνολο του συστήματος, για αυτό και ονομάζεται «μοντέλο λειτουργικής επαύξησης».

9.4 Σπειροειδές μοντέλο

Το σπειροειδές μοντέλο είναι μία γενίκευση των μοντέλων της λειτουργικής επαύξησης και της πρωτοτυποποίησης, με τα εξής νέα σημαντικά στοιχεία:

- Οι φάσεις και οι διαδικασίες ανάπτυξης του συστήματος δεν είναι προκαθορισμένες από το μοντέλο, αλλά εξειδικεύονται στον χώρο της εφαρμογής του.
- Η ανάπτυξη ολόκληρου του συστήματος χωρίζεται σε πολλούς κύκλους, σε καθέναν από τους οποίους προστίθενται νέα λειτουργικά χαρακτηριστικά στο σύστημα.
- Πριν από την έναρξη κάθε κύκλου γίνεται μία μελέτη σκοπιμότητας και ανάλυση κινδύνων, από την οποία προκύπτουν, αφενός, οι συγκεκριμένες εργασίες που θα εκτελεστούν μέσα στον κύκλο, αφετέρου, το πόσο εφικτή είναι η εκτέλεση του κύκλου αυτού.

10. Σύγχρονα μοντέλα κύκλου ζωής λογισμικού

Μεταγενέστερα μοντέλα κύκλου ζωής και ανάπτυξης πληροφοριακού συστήματος προσπαθούν να δώσουν μία γενική κατεύθυνση εφαρμογής των υπαρχουσών ιδεών, αφήνοντας όμως σημαντικούς βαθμούς ελευθερίας στον κατασκευαστή που τα ακολουθεί. Αυτό είναι αποτέλεσμα των υψηλών ρυθμών εξέλιξης στον χώρο της Πληροφορικής.

Μία περιγραφή ενός σύγχρονου μοντέλου κύκλου ζωής και ανάπτυξης πληροφοριακού συστήματος περιέχει μόνο γενικές κατευθύνσεις, οι οποίες εξειδικεύονται στο εκάστοτε περιβάλλον ανάπτυξης, πρόβλημα κ.λπ.

11. Αναλυτής συστημάτων

Ο αναλυτής συστημάτων είναι το άτομο που μελετά τα προβλήματα και τις ανάγκες ενός οργανισμού/ επιχείρησης, με σκοπό να καθορίσει τον τρόπο με τον οποίο η τεχνολογία της Πληροφορικής (υλικό, λογισμικό) μπορεί να δώσει λύσεις στα προβλήματα του οργανισμού/επιχείρησης και να υποστηρίξει/βελτιώσει τις καθημερινές λειτουργίες του. Ένα μεγάλο μέρος της εργασίας του είναι η Ανάλυση και Σχεδίαση Συστημάτων.

Ανάλυση Συστημάτων είναι η μελέτη των προβλημάτων/αναγκών του/της οργανισμού/επιχείρησης, με σκοπό την εισήγηση βελτιώσεων και τον καθορισμό μίας λύσης, η οποία θα κάνει χρήση πληροφοριακών συστημάτων.

Σχεδίαση Συστημάτων είναι η κατασκευή (σχεδιασμός) μίας τεχνικής λύσης, βασισμένη στη χρήση ηλεκτρονικών υπολογιστών, με βάση αυτά που καθορίστηκαν στην Ανάλυση Συστημάτων.

11.1 Αναγκαία προσόντα αναλυτή

Ο αναλυτής συστημάτων πρέπει να διαθέτει τα πιο κάτω προσόντα, για να ανταποκρίνεται στις υποχρεώσεις του:

- **Γνώση πληροφοριακών συστημάτων και τεχνολογίας.** Πρέπει να γνωρίζει τις υφιστάμενες μεθόδους και τεχνικές, καθώς και τις νέες τάσεις της τεχνολογίας, ώστε να τις προωθεί ως λύσεις προβλημάτων.
- **Προγραμματιστική εμπειρία.** Ο αναλυτής είναι ο συνδετικός κρίκος μεταξύ των χρηστών και των προγραμματιστών, για αυτό και είναι απαραίτητη η προγραμματιστική εμπειρία.
- **Γενική γνώση επιχειρήσεων.** Η γενική γνώση επιχειρήσεων βοηθά τον αναλυτή στην καλύτερη κατανόηση των επιχειρησιακών προβλημάτων και αναγκών.
- **Ικανότητα επίλυσης προβλημάτων.** Π.χ. Η επίλυση ενός μεγάλου προβλήματος απαιτεί τον διαχωρισμό του σε μικρότερα μέρη με τρόπο ώστε η ανάλυση και η σύνθεση τους να δώσει τη λύση του προβλήματος.
- **Διαπροσωπικές ικανότητες επικοινωνίας.** Η καλή χρήση του γραπτού και προφορικού λόγου βοηθά τον αναλυτή στην επικοινωνία του με τους χρήστες και τους προγραμματιστές.
- **Διαπροσωπικές ικανότητες σχέσεων.** Ο αναλυτής είναι αναγκαίο να λειτουργεί ενεργά σε μία ομάδα και να πείθει χωρίς να επιβάλλεται.
- **Διαλλακτικότητα και προσαρμογή.** Ο αναλυτής πρέπει να προσαρμόζεται με βάση την ιδιαιτερότητα του κάθε προβλήματος και να ορίζει εναλλακτικές λύσεις.
- **Χαρακτήρας και ηθική.** Ο αναλυτής πρέπει να είναι δυνατός χαρακτήρας και να έχει ηθική. Πρέπει να έχει εχεμύθεια και σεβασμό προς τους συνεργάτες του.
- **Ικανότητες ανάλυσης και σχεδίασης συστημάτων.** Να έχει συνεχή επιμόρφωση και διατήρηση επαφής με τις νέες τεχνολογίες στον τομέα της Πληροφορικής.

**Ασκήσεις Κεφαλαίου****Άσκηση 1.1**

Ποια είναι τα συστατικά στοιχεία ενός συστήματος;

Άσκηση 1.2

Ποια είναι τα δύο περιβάλλοντα ενός συστήματος; Πώς και γιατί επικοινωνούν μεταξύ τους τα δύο αυτά περιβάλλοντα; Δώστε ένα αναλυτικό παράδειγμα συστήματος, στο οποίο να επεξηγούνται τα πιο πάνω.

Άσκηση 1.3

Να αναφέρετε τα βασικά συστατικά στοιχεία και τον ρόλο του Πληροφοριακού Συστήματος.

Άσκηση 1.4

Να εξηγήσετε τη διαφορά ανάμεσα στον χρήστη και στον πελάτη ενός πληροφοριακού συστήματος. Μπορεί το ίδιο άτομο να είναι ταυτόχρονα χρήστης και πελάτης σε ένα πληροφοριακό σύστημα;

Άσκηση 1.5

Ένα πληροφοριακό σύστημα μετατρέπει τα δεδομένα σε πληροφορίες. Εξηγήστε τη διαφορά μεταξύ δεδομένων και πληροφοριών.

Άσκηση 1.6

Να αναφέρετε τρία διαφορετικά είδη συστημάτων, τα οποία ανήκουν στους άλλους τύπους πληροφοριακών συστημάτων.

Άσκηση 1.7

Δύο οργανισμοί, ο ΟΡΓ1 και ο ΟΡΓ2, χρησιμοποιούν ο καθένας από ένα πληροφοριακό σύστημα. Το σύστημα που χρησιμοποιεί ο ΟΡΓ1 παρέχει υποστήριξη μόνο για διοικητικές λειτουργίες και αποφάσεις του οργανισμού. Το σύστημα που χρησιμοποιεί ο ΟΡΓ2 παρέχει υποστήριξη για διοικητικές λειτουργίες και αποφάσεις του οργανισμού, αλλά ταυτόχρονα αυτοματοποιεί και όλες τις υπόλοιπες λειτουργίες και δραστηριότητες του οργανισμού.

- (1) Ποιον τύπο πληροφοριακού συστήματος χρησιμοποιεί ο οργανισμός ΟΡΓ1;
- (2) Ποιον τύπο πληροφοριακού συστήματος χρησιμοποιεί ο οργανισμός ΟΡΓ2;

Άσκηση 1.8

Να αναφέρετε δύο διαφορετικά μοντέλα Κύκλου Ζωής και Ανάπτυξης Πληροφοριακών Συστημάτων.

Άσκηση 1.9

Ποια είναι η κεντρική ιδέα του μοντέλου του καταρράκτη;

Άσκηση 1.10

Ποιο μοντέλο συνδυάζει την ακολουθιακή ανάπτυξη του μοντέλου του καταρράκτη με την τμηματική ανάπτυξη του μοντέλου της πρωτοτυποποίησης.

Άσκηση 1.11

Ποιος είναι ο ρόλος του αναλυτή συστημάτων στην ανάπτυξη ενός πληροφοριακού συστήματος;

Άσκηση 1.12

Να αναφέρετε τρία προσόντα που πρέπει να διαθέτει ένας καλός αναλυτής πληροφοριακών συστημάτων.

Άσκηση 1.13

Ο διευθυντής του Τμήματος Ανάπτυξης Πληροφοριακών Συστημάτων ενός οργανισμού, που αναπτύσσει πληροφοριακά συστήματα, ζητά να προσλάβει έναν αναλυτή συστημάτων για το τμήμα του. Μία από τις ερωτήσεις που κάνει στους υποψήφιους για τη θέση, στοχεύει στο να διαπιστώσει κατά πόσο οι υποψήφιοι έχουν προγραμματιστική εμπειρία. Γιατί επιβάλλεται ένας καλός αναλυτής συστημάτων να κατέχει αυτό το προσόν;

B6.2 Μοντέλο Καταρράκτη

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ❖ Να αναφέρουμε τις φάσεις του κύκλου ζωής και ανάπτυξης πληροφοριακών συστημάτων που προτείνεται από το μοντέλο του Καταρράκτη (Waterfall) και να περιγράψουμε σύντομα τις βασικές λειτουργίες που περιλαμβάνονται στην κάθε φάση
- ❖ Να διακρίνουμε σε ποια φάση επιτελείται κάποια λειτουργία
- ❖ Να εντοπίζουμε σε εργασιακά παραδείγματα σε ποια φάση παρουσιάστηκε πρόβλημα και ποιες φάσεις επηρεάζονται από το πρόβλημα αυτό.

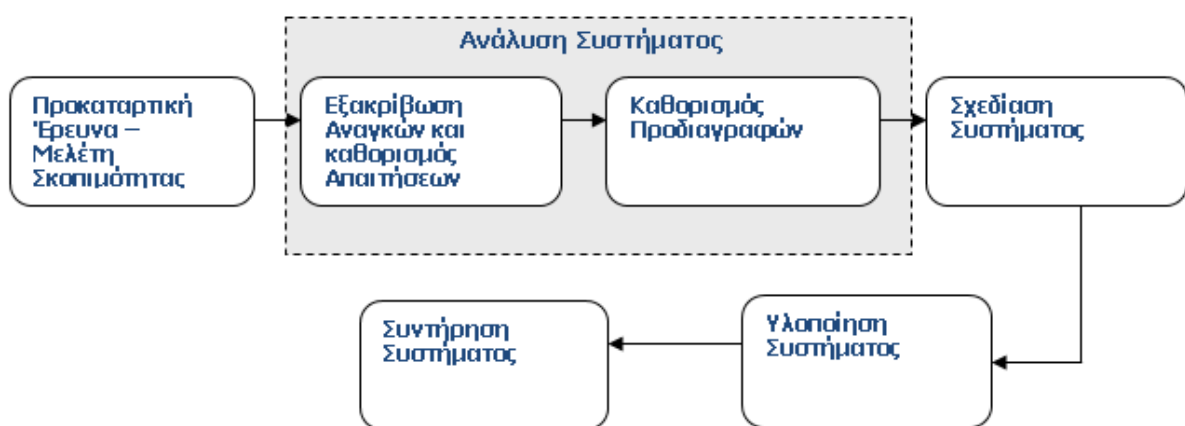
1. Κύκλος ζωής και Ανάπτυξης Πληροφοριακού Συστήματος - Μοντέλο Καταρράκτη

Η ανάπτυξη ενός πληροφοριακού συστήματος, χρησιμοποιώντας το μοντέλο του Καταρράκτη, όπως έχει αναφερθεί και πιο πάνω, διέρχεται από διάφορες φάσεις, οι οποίες συνδέονται μεταξύ τους και αλληλεξαρτώνται. Η όλη διαδικασία ανάπτυξης ονομάζεται Κύκλος Ζωής και Ανάπτυξης Πληροφοριακού Συστήματος.

2. Φάσεις Κύκλου Ζωής και Ανάπτυξης Πληροφοριακού Συστήματος:

- Προκαταρκτική έρευνα - Σύνταξη μελέτης σκοπιμότητας
 - Εξακρίβωση Αναγκών και Καθορισμός Απαιτήσεων
 - Καθορισμός Προδιαγραφών
 - Σχεδιασμός συστήματος
 - Υλοποίηση – Συνένωση Κώδικα και Έλεγχος συστήματος
 - Συντήρηση συστήματος
- } Ανάλυση Συστήματος

3. Κύκλος Ζωής και Ανάπτυξης Πληροφοριακού Συστήματος



Κατά την ανάπτυξη ενός πληροφοριακού συστήματος, είναι σημαντικό τα διάφορα προβλήματα ή λάθη να εντοπίζονται νωρίς, ώστε το κόστος διόρθωσής τους σε χρόνο και χρήμα να μην είναι πολύ μεγάλο. Όταν εντοπιστεί κάποιο πρόβλημα ή λάθος, πρέπει να επιστρέψουμε πίσω στη φάση εκείνη από όπου αναφύεται και να το διορθώσουμε, προσαρμόζοντας κατάλληλα και όλες τις φάσεις που ακολούθησαν μετά από τη φάση αυτή.

Για παράδειγμα, ένα λάθος στον καθορισμό των προδιαγραφών του συστήματος, το οποίο γίνεται αντιληπτό κατά τη φάση της υλοποίησης, θα προκαλέσει αλλαγές στις προδιαγραφές, στη σχεδίαση και στην υλοποίηση του συστήματος.

4. Προκαταρκτική Έρευνα - Σύνταξη μελέτης σκοπιμότητας

Η Προκαταρκτική Έρευνα σκοπό έχει να εξακριβώσει εάν υπάρχει ανάγκη για δημιουργία ενός νέου πληροφοριακού συστήματος. Είναι μία φάση μικρής διάρκειας, κατά την οποία γίνεται μία γρήγορη εξέταση του υπάρχοντος συστήματος και των προβλημάτων, ευκαιριών και εντολών.

- **Πρόβλημα** είναι μία κατάσταση που εμποδίζει έναν/μία οργανισμό/επιχείρηση να εκπληρώσει τους στόχους του/της. Για παράδειγμα, δημιουργούνται μεγάλες ουρές πελατών, επειδή υπάρχει καθυστέρηση στα ταμεία λόγω του ότι είναι αργό το υφιστάμενο πληροφοριακό σύστημα.
- **Ευκαιρία** είναι μία κατάσταση η οποία βελτιώνει την αποδοτικότητα ενός/μίας οργανισμού/επιχείρησης. Για παράδειγμα, ο αριθμός των παραγγελιών κάποιου προϊόντος μπορεί να αυξηθεί, φτιάχνοντας μία ιστοσελίδα για τοποθέτηση παραγγελιών από το διαδίκτυο.
- **Εντολή** είναι μία απαίτηση η οποία επιβάλλεται σε έναν/μία οργανισμό/επιχείρηση από το εξωτερικό περιβάλλον. Για παράδειγμα, αν ένα σύστημα φτιάχτηκε για να δέχεται 8-ψήφιους τηλεφωνικούς αριθμούς αλλά τώρα θα πρέπει να δέχεται τουλάχιστον 10-ψήφιους, θα πρέπει να τροποποιηθεί και να αλλαχτούν αποθηκευτικές μονάδες, Οθόνες εισαγωγής δεδομένων, αναφορές κ.λπ.

Εάν το αποτέλεσμα της προκαταρκτικής έρευνας είναι θετικό τότε αρχίζει η Μελέτη Σκοπιμότητας.

Η **Μελέτη Σκοπιμότητας** (Feasibility Study) έχει ως στόχο να ερευνήσει μέσα στον/στην οργανισμό/επιχείρηση τη δυνατότητα για τη δημιουργία του νέου συστήματος, πόσο ωφέλιμο θα είναι και αν πραγματικά συμφέρει.

Θα παρουσιάσει:

- Τα πλεονεκτήματα: Ποια τα οφέλη που θα υπάρξουν για τον/την οργανισμό/επιχείρηση
- Τα μειονεκτήματα: Ποιες είναι οι επιπτώσεις από την υλοποίηση
- Οικονομικό τομέα: Ποια είναι η οικονομική επιβάρυνση
- Τεχνικό τομέα: Ανάγκες για μηχανήματα και εξειδικευμένο προσωπικό
- Κοινωνικό τομέα: Επιδράσεις στο προσωπικό (ανακατατάξεις, απολύσεις, επιμορφώσεις κ.λπ.)
- Χρονοδιαγράμματα: Χρόνος αποπεράτωσης, πότε θα λειτουργήσει

5. Εξακρίβωση Αναγκών και Καθορισμός Απαιτήσεων

Η φάση Εξακρίβωσης Αναγκών και Καθορισμού Απαιτήσεων είναι η πρώτη από τις δύο φάσεις που περιλαμβάνει η Ανάλυση Συστήματος.

Κατά τη φάση αυτή, θα πρέπει να εξακριβωθούν οι πραγματικές ανάγκες του πελάτη, βάσει των οποίων θα καθορισθούν τα λειτουργικά χαρακτηριστικά (απαιτήσεις) που πρέπει να διαθέτει το πληροφοριακό σύστημα, ώστε να καλύπτει αυτές τις ανάγκες.

Ο εντοπισμός των πραγματικών αναγκών επιτυγχάνεται με τη λεπτομερή καταγραφή των λειτουργιών του/της οργανισμού/ επιχείρησης, χρησιμοποιώντας διάφορες τεχνικές, όπως:

- Παρατήρηση στον τόπο εργασίας (Observation)
- Συνεντεύξεις (Interviews)
- Ερωτηματολόγια (Questionnaires)
- Συλλογή και μελέτη έντυπου υλικού
- Γρήγορη πρωτοτυποποίηση

6. Καθορισμός Προδιαγραφών

Η φάση Καθορισμού Προδιαγραφών είναι η δεύτερη από τις δύο φάσεις που περιλαμβάνει η Ανάλυση Συστήματος. Κατά τη φάση αυτή οι ανάγκες του πελάτη αναλύονται, καταγράφονται και παρουσιάζονται στη μορφή ενός εγγράφου προδιαγραφών. Εδώ θα καθορισθούν επακριβώς οι λειτουργίες που πρέπει να διαθέτει το πληροφοριακό σύστημα, δηλαδή τι ακριβώς θα κάνει, χωρίς όμως ακόμη να προσδιορίζεται το πώς θα το κάνει. Επίσης, στο έγγραφο αυτό καταγράφονται οποιοιδήποτε περιορισμοί, όπως κόστος και χρονικές προθεσμίες. Το έγγραφο προδιαγραφών πρέπει να διακρίνεται από σαφήνεια και να είναι ολοκληρωμένο, γιατί με βάση αυτό θα γίνει η σχεδίαση και η υλοποίηση του συστήματος. Επίσης, αυτό το έγγραφο, χρησιμοποιείται συνήθως και ως συμβόλαιο μεταξύ του πελάτη και της εταιρείας που θα αναπτύξει το σύστημα, για αυτό θα πρέπει να γραφτεί με τρόπο που να γίνεται κατανοητό από τον πελάτη.

7. Σχεδιασμός συστήματος

Η φάση Σχεδίαση του Συστήματος θα καθορίσει το πώς το σύστημα θα κάνει αυτά που έχουν καταγραφεί στη φάση του καθορισμού προδιαγραφών. Στην αρχή καθορίζονται από την ομάδα ανάπτυξης τα διάφορα μέρη που θα αποτελούν το σύστημα και ο τρόπος επικοινωνίας τους. Στη συνέχεια γίνεται λεπτομερής σχεδιασμός των διαφόρων αυτών μερών με τη χρήση των κατάλληλων αλγορίθμων. Θα σχεδιαστούν οι βάσεις δεδομένων του συστήματος (πίνακες και σχέση μεταξύ τους), καθώς επίσης οι οθόνες εισαγωγής δεδομένων (Φόρμες) και οι αναφορές (Reports).

8. Υλοποίηση – Συνένωση Κώδικα και Έλεγχος συστήματος

Στη φάση Υλοποίηση – Συνένωση Κώδικα και Έλεγχος του Συστήματος γίνεται η ανάπτυξη των διαφόρων μερών του λογισμικού με κώδικα (κωδικοποίηση), η σύνδεση μεταξύ τους και ο απαιτούμενος έλεγχος. Κατά τη φάση αυτή γίνεται η εγκατάσταση του συστήματος στον πελάτη, εκπαίδευση των χρηστών στην χρήση του συστήματος και ο έλεγχος του συστήματος από τον πελάτη.

9. Συντήρηση συστήματος

Η φάση Συντήρηση Συστήματος αρχίζει μετά που ο πελάτης θα παραλάβει το πληροφοριακό σύστημα και θα το αποδεχτεί. Αυτή η φάση περιλαμβάνει οποιεσδήποτε αλλαγές θα πρέπει να γίνουν στο σύστημα συνεπεία νέων αναγκών και απαιτήσεων του πελάτη, καθώς επίσης και διορθώσεις οποιωνδήποτε λαθών εμφανιστούν στο σύστημα, κατά την διάρκεια της λειτουργίας του. Αυτή η φάση είναι και η μεγαλύτερη του κύκλου ανάπτυξης του Πληροφοριακού Συστήματος. Ένα βασικό πρόβλημα, που μπορεί να έχει αυτή η φάση, είναι η πιθανή έλλειψη καλής τεκμηρίωσης των διαφόρων προηγούμενων φάσεων.

**Ασκήσεις Κεφαλαίου****Άσκηση 2.1**

Να αναφέρετε με την σειρά τις έξι φάσεις που διέρχεται η ανάπτυξη ενός πληροφοριακού συστήματος με την μέθοδο του Καταρράκτη.

Άσκηση 2.2

Κατά τη φάση της Προκαταρκτικής έρευνας έχουν καταγραφεί μεταξύ άλλων και τα πιο κάτω:

- (1) Οι πελάτες της εταιρείας επιθυμούν όπως η αλληλογραφία τους με την εταιρεία διεξάγεται μέσω ηλεκτρονικού ταχυδρομείου, κάτι το οποίο αποδέχεται η διοίκηση της εταιρείας. Στο υπάρχον πληροφοριακό σύστημα δεν υπάρχει πρόνοια για την καταχώρηση του ηλεκτρονικού ταχυδρομείου του πελάτη. Πρέπει να αλλαχτεί το αρχείο πελατών, οι φόρμες εισαγωγής δεδομένων και οι αναφορές.
- (2) Το υπάρχον σύστημα είναι αργό και υπάρχει καθυστέρηση στην έκδοση τιμολογίων προς τους πελάτες.
- (3) Η Ευρωπαϊκή Ένωση έδωσε οδηγίες για αλλαγή του τρόπου φορολογίας των εισαγόμενων προϊόντων από χώρες μέλη της ΕΕ. Το σύστημα πρέπει να τροποποιηθεί και να αλλαχθούν οθόνες, αναφορές κ.λπ.
- (4) Η δημιουργία και λειτουργία ιστοσελίδας για εξυπηρέτηση των πελατών για παραγγελίες, εκτιμάται ότι θα αυξήσει κατά πολύ τους πελάτες και τις πωλήσεις της επιχείρησης.

Να ορίσετε αν το κάθε ένα από τα πιο πάνω είναι πρόβλημα, ευκαιρία ή εντολή και να δικαιολογήσετε την απάντησή σας.

Άσκηση 2.3

Να αναφέρετε ονομαστικά τη φάση του Κύκλου Ζωής και Ανάπτυξης Πληροφοριακού Συστήματος κατά την οποία ο αναλυτής θα χρησιμοποιήσει Διαγράμματα Ροής Δεδομένων (ΔΡΔ).

Άσκηση 2.4

Ποια φάση του Κύκλου Ζωής και Ανάπτυξης Πληροφοριακού Συστήματος προηγείται της φάσης «Υλοποίηση και Συνένωση Κώδικα» και ποια φάση ακολουθεί;

Άσκηση 2.5

Να αναφέρετε δύο τεχνικές για τον εντοπισμό των πραγματικών αναγκών κατά τη φάση «Εξακρίβωση Αναγκών και Καθορισμός Απαιτήσεων».

Άσκηση 2.6

Ποια είναι η τελευταία φάση του Κύκλου Ζωής και Ανάπτυξης Πληροφοριακού Συστήματος και πότε αρχίζει;

Άσκηση 2.7

Σε ποια φάση του Κύκλου Ζωής και Ανάπτυξης Πληροφοριακών Συστημάτων καθορίζεται το τι θα κάνει το πληροφοριακό σύστημα, το οποίο θα αναπτυχθεί και σε ποια φάση καθορίζεται το πώς θα το κάνει;

Άσκηση 2.8

Κατά τη διάρκεια της λειτουργίας ενός πληροφοριακού συστήματος ο πελάτης εντόπισε λάθη σε αυτό και ενημέρωσε σχετικά την εταιρεία που το ανέπτυξε για να τα διορθώσει. Να αναφέρετε ονομαστικά τη φάση του Κύκλου ζωής και Ανάπτυξης Πληροφοριακού Συστήματος όπου βρίσκεται αυτό το πληροφοριακό σύστημα;

Άσκηση 2.9

Τι είναι το έγγραφο προδιαγραφών και σε τι χρησιμεύει;

B6.3 Μέθοδοι Καταγραφής Προδιαγραφών

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ❖ Να αναφέρουμε τον σκοπό και τους λόγους που πιθανόν να οδηγήσουν στην εκπόνηση Προκαταρκτικής Έρευνας
- ❖ Να διακρίνουμε και να κατηγοριοποιούμε σε εργασιακά παραδείγματα τους λόγους που οδήγησαν στην εκπόνηση Προκαταρκτικής Έρευνας (Πρόβλημα, Ευκαιρία ή Εντολή)
- ❖ Να αναφέρουμε τον σκοπό της Μελέτης Σκοπιμότητας και τι περιλαμβάνει
- ❖ Να διακρίνουμε σε εργασιακά παραδείγματα κατά πόσο περιλαμβάνονται όλα τα απαραίτητα στοιχεία και να εντοπίζουμε ποια απουσιάζουν
- ❖ Να αναφέρουμε τεχνικές εξακρίβωσης αναγκών και να εισηγούμαστε κατάλληλη τεχνική σε εργασιακά παραδείγματα
- ❖ Να αναφέρουμε Μεθόδους καθορισμού/καταγραφής προδιαγραφών και να τις συγκρίνουμε ως προς το τι καταγράφουν, τα πλεονεκτήματα και τα μειονεκτήματά τους
- ❖ Να αναγνωρίζουμε ένα Διάγραμμα Ροής Δεδομένων και να αναφέρουμε ποιες πτυχές του πληροφοριακού συστήματος παρουσιάζει
- ❖ Να αναγνωρίζουμε τα σύμβολα ενός Διαγράμματος Ροής Δεδομένων και να αναφέρουμε τον ρόλο του καθενός
- ❖ Να αναγνωρίζουμε τα βασικά δομικά στοιχεία στην περιγραφή ενός πληροφοριακού συστήματος και να σχεδιάζουμε ένα Διάγραμμα Ροής Δεδομένων που να ταιριάζει στην περιγραφή
- ❖ Να διακρίνουμε μεταξύ φυσικών και λογικών μοντέλων και Διαγραμμάτων Ροής Δεδομένων
- ❖ Να αναγνωρίζουμε τον ρόλο του εσωτερικού και του εξωτερικού περιβάλλοντος στη διάκριση μεταξύ φυσικών και λογικών Διαγραμμάτων Ροής Δεδομένων
- ❖ Να εντοπίζουμε σφάλματα σε ένα Διάγραμμα ροής Δεδομένων και να προτείνουμε τρόπους διόρθωσής τους.

Η τρίτη φάση του Κύκλου Ζωής και Ανάπτυξης Πληροφοριακού Συστήματος, όπως έχουμε δει στο προηγούμενο κεφάλαιο, είναι η φάση Καταγραφής Προδιαγραφών κατά την οποία θα καθορισθεί τι ακριβώς θα κάνει το πληροφοριακό σύστημα. Υπάρχουν διάφορες μέθοδοι καταγραφής των προδιαγραφών, με την κάθε μία να έχει τα δικά της χαρακτηριστικά. Μερικές μέθοδοι έχουν ως επίκεντρο τις διαδικασίες, οι οποίες λαμβάνουν χώρα και ονομάζονται μέθοδοι μοντελοποίησης διαδικασιών, ενώ άλλες έχουν ως επίκεντρο τα δεδομένα και ονομάζονται μέθοδοι μοντελοποίησης δεδομένων. Τις περισσότερες φορές χρησιμοποιείται συνδυασμός των μεθόδων αυτών για την καταγραφή των προδιαγραφών.

1. Μέθοδοι καταγραφής προδιαγραφών:

- **Φυσική Γλώσσα.** Αποφεύγεται η χρήση της μεθόδου αυτής, διότι είναι επιρρεπής σε ασάφειες, λάθη, παραλήψεις και περιττολογία.
- **Διαγράμματα Ροής Δεδομένων (ΔΡΔ).** Η τεχνική αυτή επικεντρώνεται στις διαδικασίες ενός συστήματος. Είναι μέθοδος η οποία επιτρέπει αρκετά λεπτομερή προδιαγραφή και γίνεται σχετικά εύκολα κατανοητή από τους πελάτες.
- **Διαγράμματα Οντοτήτων Συσχετίσεων (ΔΟΣ).** Η τεχνική αυτή επικεντρώνεται στη μορφή των δεδομένων, τα οποία διακινούνται σε ένα σύστημα και χρησιμοποιείται σχεδόν πάντοτε για την προδιαγραφή βάσεων δεδομένων.

2. Λογικά και Φυσικά Μοντέλα

Στην ανάλυση και σχεδίαση πληροφοριακών συστημάτων χρησιμοποιούνται μοντέλα (απεικόνιση της πραγματικότητας με σχεδιαγράμματα), τα οποία βοηθούν στην καλύτερη κατανόηση του τρόπου με τον οποίο λειτουργεί ή θα πρέπει να λειτουργεί ένα σύστημα.

Υπάρχουν δύο κατηγορίες μοντέλων. Τα λογικά μοντέλα και τα φυσικά μοντέλα.

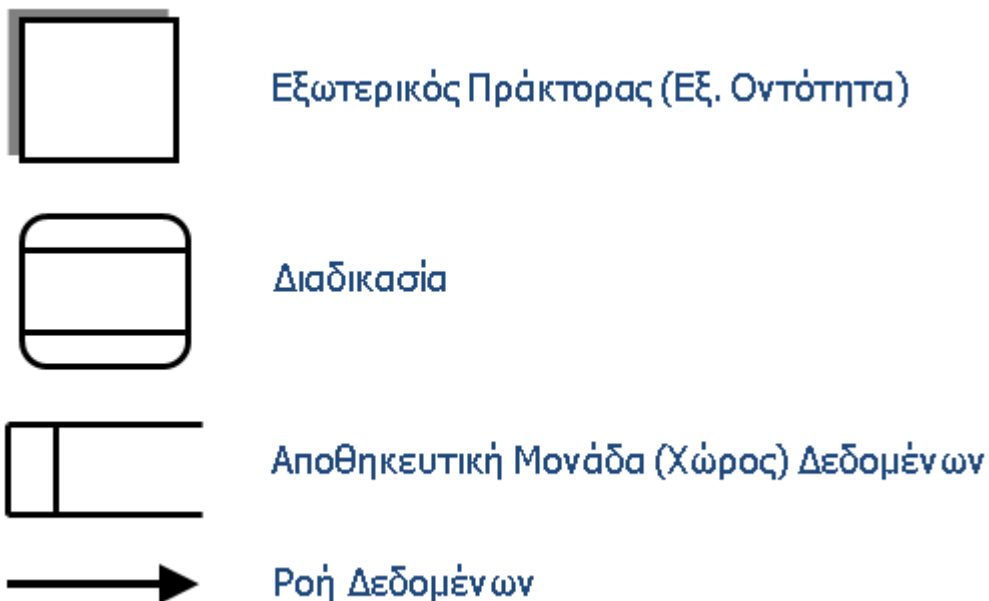
- **Λογικά Μοντέλα.** Φανερώνουν **τι** κάνει ένα σύστημα, όχι όμως **πώς**.
- **Φυσικά Μοντέλα.** Φανερώνουν όχι μόνο **τι** κάνει ένα σύστημα, αλλά, επίσης, και **πώς** το κάνει.

3. Διαγράμματα Ροής Δεδομένων (ΔΡΔ) (Data Flow Diagrams)

Μία από τις ιδιαίτερα προσφιλείς μεθόδους καταγραφής των προδιαγραφών του συστήματος, είναι η χρήση του Διαγράμματος Ροής Δεδομένων (ΔΡΔ).

Σε αυτό δίδεται σχηματικά η ροή των δεδομένων μέσα σε ένα πληροφοριακό σύστημα και οι επεξεργασίες που γίνονται πάνω σε αυτά. Είναι μέθοδος μοντελοποίησης διαδικασιών και χρησιμοποιεί λογικά μοντέλα, τα οποία μας φανερώνουν τι κάνει το σύστημα (όχι όμως πώς). Η χρήση του προσφέρει αρκετά λεπτομερή καταγραφή των προδιαγραφών, ώστε να γίνονται εύκολα κατανοητές και από τον πελάτη.

4. Σύμβολα Διαγραμμάτων Ροής Δεδομένων



4.1 Εξωτερικός Πράκτορας (Εξωτερική Οντότητα)

Πηγή δεδομένων ή προορισμός πληροφοριών, π.χ. Άτομα που παρέχουν δεδομένα, όπως μία παραγγελία κάποιου πελάτη ή δέχονται πληροφορίες, όπως ένα τιμολόγιο.

4.2 Διαδικασία

Μία επεξεργασία που επιτελείται στα δεδομένα, π.χ. Καταχώριση στοιχείων, δημιουργία κατάστασης.

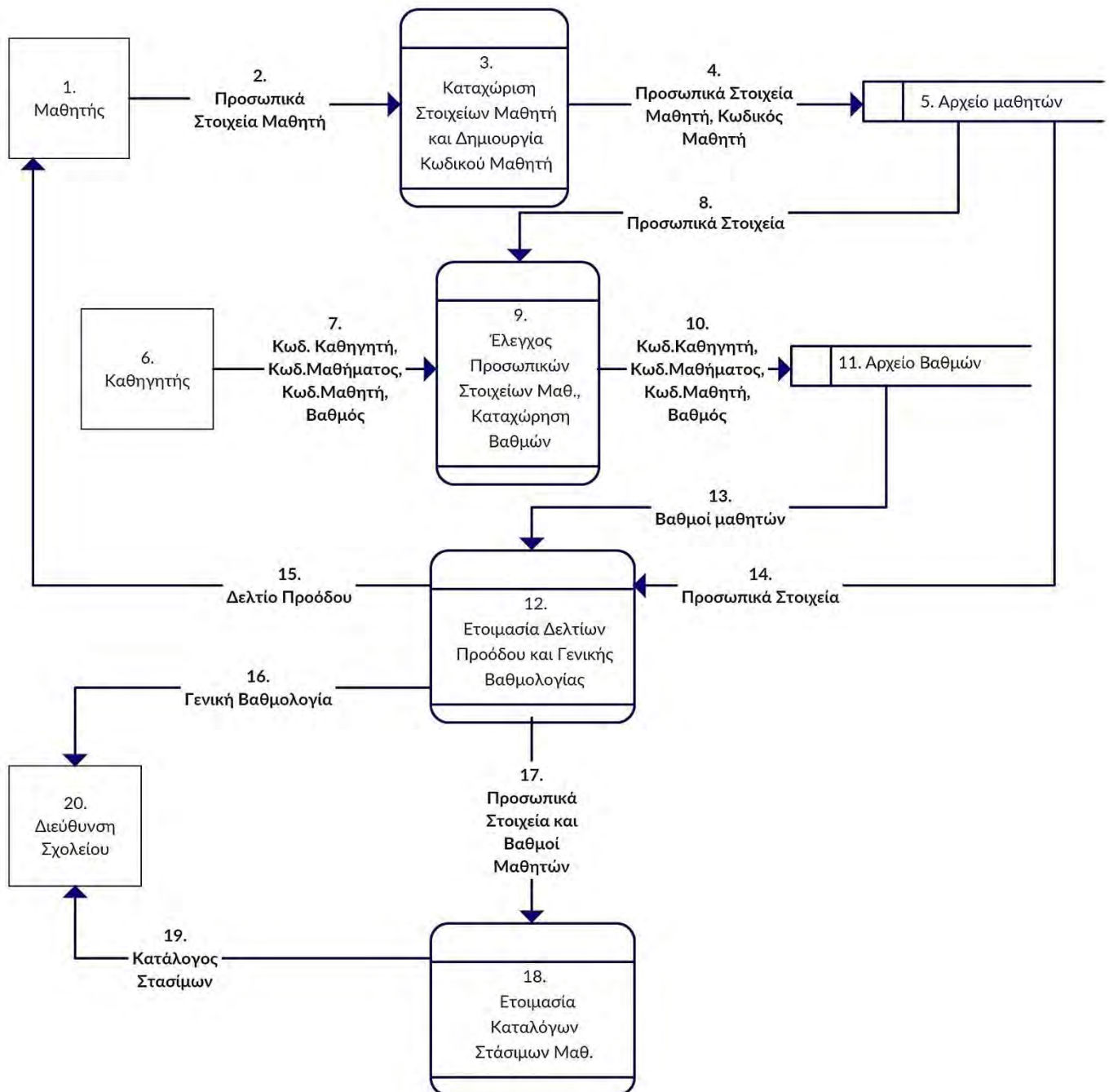
4.3 Αποθηκευτική Μονάδα (Χώρος) Δεδομένων

Είναι χώρος (αρχείο στον δίσκο), όπου φυλάγονται επί μόνιμης βάσεως δεδομένα, π.χ. Αρχείο Μαθητών

4.4 Ροή Δεδομένων

Αναπαριστούν εισόδους και εξόδους δεδομένων **από και προς μία διαδικασία**. Οι ροές δεδομένων προέρχονται από εξωτερικούς πράκτορες, αποθηκευτικές μονάδες ή άλλες διαδικασίες (όταν γίνονται την ίδια χρονική στιγμή ή/και όταν η εκτέλεση της δεύτερης διαδικασίας προϋποθέτει την εκτέλεση της πρώτης). Στο βέλος πρέπει να αναγράφονται τα δεδομένα που εμπλέκονται.

5. Παράδειγμα Διαγράμματος Ροής Δεδομένων



6. Βασικοί Κανόνες που διέπουν τον σχεδιασμό Διαγραμμάτων Ροής Δεδομένων (ΔΡΔ)

Κάθε σύμβολο (σχήμα) του ΔΡΔ πρέπει:

- να περιλαμβάνει (περιληπτικά) ποιος δίνει / ποιος παίρνει / τι κάνει / τι μεταφέρει,
- να αριθμείται (μοναδικοί αριθμοί, όχι κατά ανάγκη συνεχόμενοι),
- να γίνεται περιγραφή του,
- αν χρησιμοποιηθεί η ίδια οντότητα ή αρχείο ξανά στο ΔΡΔ, θα πρέπει να δοθεί ο ίδιος αριθμός. π.χ. Μαθητής

1

7. Λάθη σε ΔΡΔ

7.1 Μη Έγκυρες Ροές Δεδομένων

Οι ροές δεδομένων, οι οποίες δεν προέρχονται ή δεν καταλήγουν σε μία διαδικασία, είναι μη έγκυρες.

7.1.1 Από εξωτερικό πράκτορα απευθείας σε εξωτερικό πράκτορα

Μία ροή δεδομένων δεν μπορεί να φύγει από έναν εξωτερικό πράκτορα και να καταλήξει απ' ευθείας σε έναν άλλο εξωτερικό πράκτορα. Πρέπει να φτάσει μέσω κάποιας διαδικασίας.



7.1.2 Από εξωτερικό πράκτορα απευθείας σε αποθηκευτική μονάδα δεδομένων

Μία ροή δεδομένων δεν πρέπει να εξέρχεται από έναν εξωτερικό πράκτορα και να καταλήγει απ' ευθείας σε μία αποθηκευτική μονάδα δεδομένων (αρχείο). Πρέπει να καταλήγει εκεί μέσω κάποιας διαδικασίας.



7.1.3 Από αποθηκευτική μονάδα δεδομένων απευθείας σε εξωτερικό πράκτορα

Μία ροή δεδομένων δεν πρέπει να καταλήγει απ' ευθείας από μία αποθηκευτική μονάδα δεδομένων (αρχείο), σε έναν εξωτερικό πράκτορα. Θα πρέπει να πηγαίνει μέσω κάποιας διαδικασίας.



7.1.4 Από αποθηκευτική μονάδα δεδομένων απευθείας σε αποθηκευτική μονάδα δεδομένων

Μία ροή δεδομένων δεν πρέπει να φεύγει από μία αποθηκευτική μονάδα δεδομένων και να καταλήγει απ' ευθείας σε μία άλλη αποθηκευτική μονάδα δεδομένων. Θα πρέπει να καταλήγει εκεί μέσω κάποιας διαδικασίας.



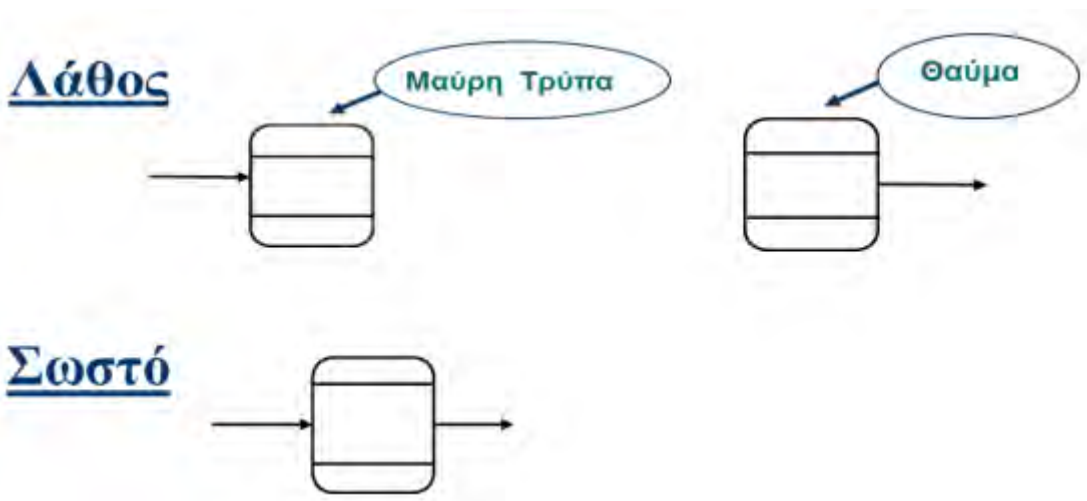
7.2 Λογικά Λάθη στα ΔΡΔ

7.2.1 Μαύρες Τρύπες

Μία διαδικασία θα πρέπει να έχει τουλάχιστον μία ροή δεδομένων, που να εξέρχεται από αυτήν.

7.2.2 Θαύματα

Μία διαδικασία θα πρέπει να έχει τουλάχιστον μία ροή δεδομένων, που να εισέρχεται σε αυτήν.



7.2.3 Γκρίζες Τρύπες

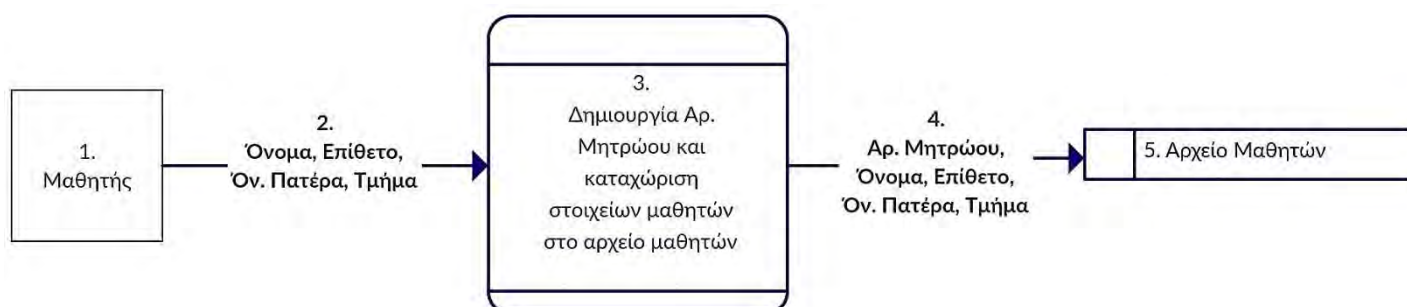
Τα δεδομένα που εισέρχονται σε μία διαδικασία θα πρέπει μετά την επεξεργασία που γίνεται σε αυτά, να μπορούν να παράγουν τις ζητούμενες πληροφορίες που πρέπει να εξαχθούν. Διαφορετικά έχουμε τις λεγόμενες **γκρίζες τρύπες**, οι οποίες είναι από τα πιο δύσκολα λάθη, ως προς την ανίχνευσή τους.

Επίσης, σε ένα ΔΡΔ μπορεί να υπάρχουν και άλλα λογικά λάθη, όπως μία κατάσταση να μην καταλήγει στον σωστό εξωτερικό πράκτορα.

Παράδειγμα 1: Σχεδιασμός Διαγράμματος Ροής Δεδομένων

Ένας καθηγητής χρειάζεται ένα πληροφοριακό σύστημα, το οποίο να δέχεται και να αποθηκεύει τα προσωπικά στοιχεία των μαθητών του και το τμήμα τους. Οι μαθητές θα δίνουν τα προσωπικά τους στοιχεία (όνομα, επίθετο, όνομα πατέρα και το τμήμα τους), τα οποία μαζί με έναν αριθμό μητρώου που θα δημιουργηθεί αυτόματα από το σύστημα, θα καταχωρίζονται στο αρχείο Μαθητών.

Να σχεδιάσετε το Διάγραμμα Ροής Δεδομένων (ΔΡΔ) για τη λειτουργία που περιγράφεται πιο πάνω και να το περιγράψετε.

Σχεδιασμός ΔΡΔ**Περιγραφή ΔΡΔ****Εξωτερική Οντότητα:**

Μαθητής (1): Οι μαθητές δίνουν τα προσωπικά τους στοιχεία (Όνομα, Επίθετο, Όνομα Πατέρα, Τμήμα), για να καταχωρισθούν στο σύστημα.

Διαδικασία:

Διαδικασία (3): Τα προσωπικά στοιχεία των μαθητών καταχωρίζονται στο σύστημα (Εγγραφή στο αρχείο μαθητών) με φόρμα οθόνης, αφού πρώτα δημιουργηθεί ο αριθμός μητρώου.

Αρχείο:

Αρχείο Μαθητών (5): Εκεί που φυλάσσονται όλα τα στοιχεία των μαθητών.

Ροές Δεδομένων:

Ροή (2): Τα προσωπικά στοιχεία των μαθητών εισάγονται στη φόρμα οθόνης.

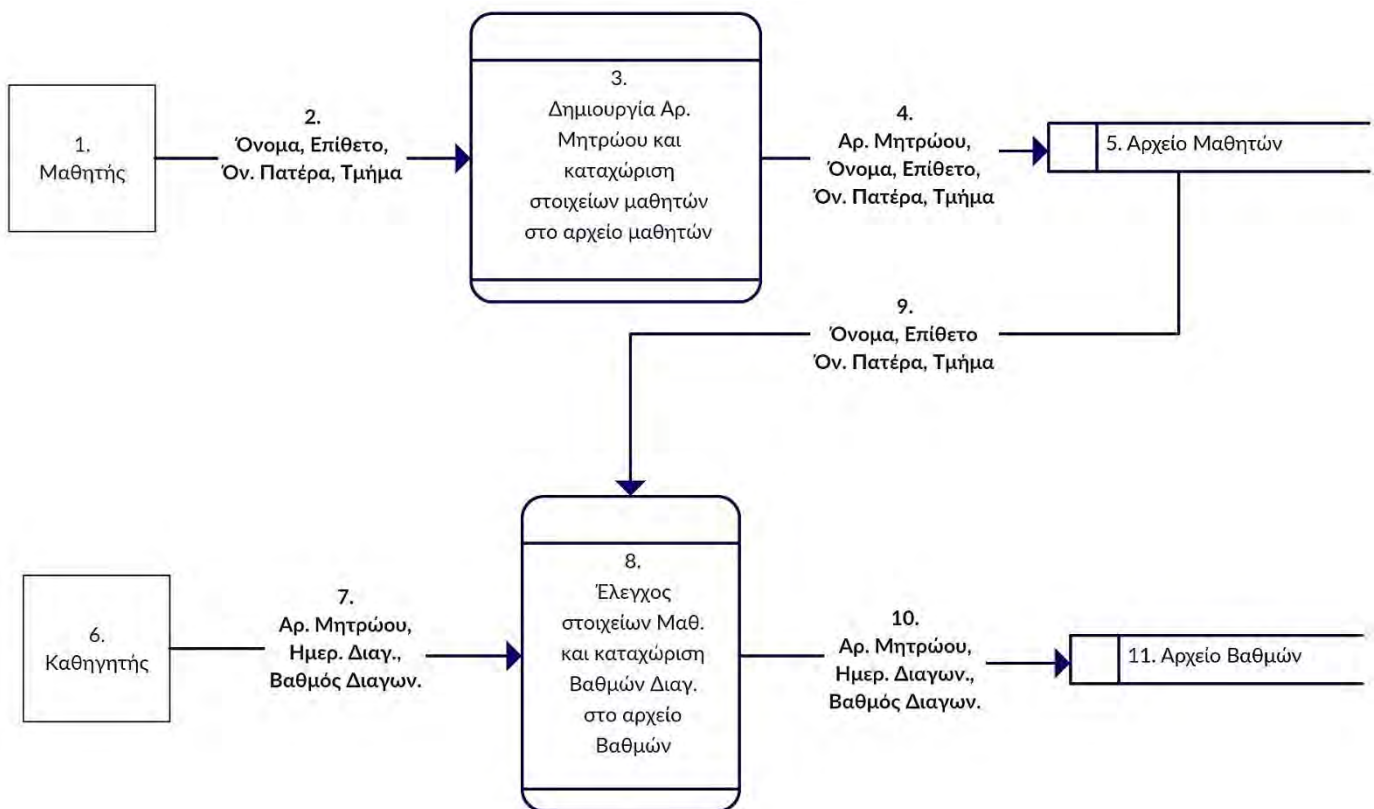
Ροή (4): Τα προσωπικά στοιχεία των μαθητών και ο αριθμός μητρώου εισάγονται από τη φόρμα οθόνης στο αρχείο μαθητών.

Παράδειγμα 2: Σχεδιασμός Διαγράμματος Ροής Δεδομένων

Το πιο πάνω πληροφοριακό σύστημα θα πρέπει επιπρόσθετα να δέχεται και να αποθηκεύει τον βαθμό διαγωνίσματος κάθε μαθητή. Ο καθηγητής θα δίνει τον αριθμό μητρώου και τον βαθμό διαγωνίσματος κάθε μαθητή του, καθώς επίσης την ημερομηνία του διαγωνίσματος. Αφού γίνει έλεγχος των προσωπικών στοιχείων του μαθητή (όνομα, επίθετο, όνομα πατέρα, τμήμα), τα στοιχεία θα καταχωρίζονται στο αρχείο Βαθμών.

Να συμπληρώσετε το πιο πάνω Διάγραμμα Ροής Δεδομένων (ΔΡΔ), ώστε να ανταποκρίνεται και στη νέα λειτουργία και να το περιγράψετε.

Σχεδιασμός ΔΡΔ



Περιγραφή ΔΡΔ

Εξωτερικές Οντότητες:

Μαθητής (1): Ο μαθητής δίνει τα προσωπικά του στοιχεία (όνομα, επίθετο, όνομα πατέρα, τμήμα), για να καταχωρισθούν στο σύστημα.

Καθηγητής (6): Ο καθηγητής δίνει τον αριθμό μητρώου και τον βαθμό διαγωνίσματος κάθε μαθητή του, καθώς επίσης την ημερομηνία του διαγωνίσματος, για να καταχωρισθούν στο σύστημα.

Διαδικασίες:

Διαδικασία (3): Τα προσωπικά στοιχεία των μαθητών καταχωρίζονται στο σύστημα (Εγγραφή στο αρχείο μαθητών) με φόρμα οθόνης, αφού πρώτα δημιουργηθεί ο αριθμός μητρώου.

Διαδικασία (8): Ο βαθμός διαγωνίσματος του μαθητή καταχωρίζεται στο σύστημα (Εγγραφή στο αρχείο βαθμών) με φόρμα οθόνης, αφού πρώτα γίνει έλεγχος των προσωπικών του στοιχείων.

Αρχεία:

Αρχείο Μαθητών (5): Εκεί που φυλάσσονται όλα τα στοιχεία των μαθητών.

Αρχείο Βαθμών (11): Εκεί φυλάσσονται οι βαθμοί διαγωνισμάτων των μαθητών.

Ροές Δεδομένων:

Ροή (2): Τα προσωπικά στοιχεία των μαθητών εισάγονται στη φόρμα οθόνης.

Ροή (4): Τα προσωπικά στοιχεία των μαθητών και ο αριθμός μητρώου εισάγονται από τη φόρμα οθόνης στο αρχείο μαθητών.

Ροή (7): Ο βαθμός διαγωνίσματος των μαθητών εισάγεται στη φόρμα οθόνης.

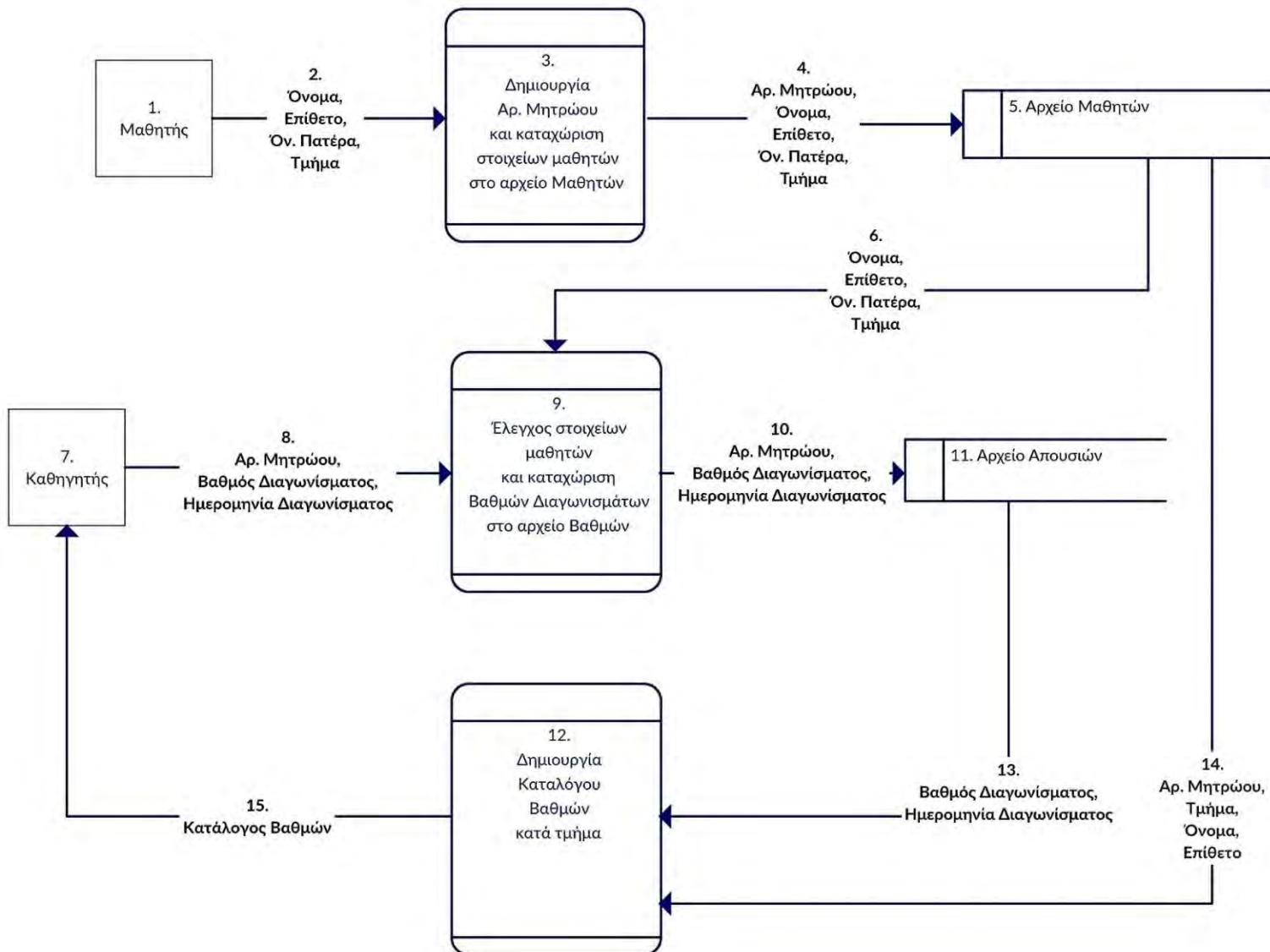
Ροή (9): Τα προσωπικά στοιχεία των μαθητών (όνομα, επίθετο, όνομα πατέρα, τμήμα) χρησιμοποιούνται για έλεγχο.

Ροή (10): Ο βαθμός διαγωνίσματος των μαθητών εισάγεται από τη φόρμα οθόνης στο αρχείο μαθητών.

Παράδειγμα 3: Σχεδιασμός Διαγράμματος Ροής Δεδομένων

Το πιο πάνω πληροφοριακό σύστημα θα πρέπει επιπρόσθετα να δημιουργεί κατάλογο μαθητών κατά τμήμα, με τους βαθμούς που πήραν στα διαγωνίσματα τους (αριθμό μητρώου, τμήμα, όνομα, επίθετο, ημερομηνία διαγωνίσματος, βαθμός). Ο κατάλογος θα δίνεται στον καθηγητή.

Να συμπληρώσετε το πιο πάνω Διάγραμμα Ροής Δεδομένων (ΔΡΔ), ώστε να ανταποκρίνεται και στη νέα λειτουργία και να το περιγράψετε.

Σχεδιασμός ΔΡΔ**Περιγραφή ΔΡΔ****Εξωτερικές Οντότητες:**

Μαθητής (1): Ο μαθητής δίνει τα προσωπικά του στοιχεία (όνομα, επίθετο, όνομα πατέρα, τμήμα), για να καταχωρισθούν στο σύστημα.

Καθηγητής (6): Ο καθηγητής δίνει τον αριθμό μητρώου και τον βαθμό διαγωνίσματος κάθε μαθητή του, καθώς επίσης την ημερομηνία του διαγωνίσματος, για να καταχωρισθούν στο σύστημα. Επίσης, παίρνει τον κατάλογο βαθμών.

Διαδικασίες:

Διαδικασία (3): Τα προσωπικά στοιχεία των μαθητών καταχωρίζονται στο σύστημα (Εγγραφή στο αρχείο μαθητών) με φόρμα οθόνης, αφού πρώτα δημιουργηθεί ο αριθμός μητρώου.

Διαδικασία (8): Ο βαθμός διαγωνίσματος του μαθητή καταχωρίζεται στο σύστημα (Εγγραφή στο αρχείο βαθμών) με φόρμα οθόνης, αφού πρώτα γίνει έλεγχος των προσωπικών του στοιχείων.

Διαδικασία (12): Δημιουργία καταλόγου μαθητών κατά τμήμα, με τους βαθμούς που πήραν στα διαγωνίσματά τους (τμήμα, όνομα, επίθετο, ημερομηνία διαγωνίσματος, βαθμός).

Αρχεία:

Αρχείο Μαθητών (5): Εκεί που φυλάσσονται όλα τα στοιχεία των μαθητών.

Αρχείο Βαθμών (11): Εκεί φυλάσσονται οι βαθμοί διαγωνισμάτων των μαθητών.

Ροές Δεδομένων:

Ροή (2): Τα προσωπικά στοιχεία των μαθητών εισάγονται στη φόρμα οθόνης.

Ροή (4): Τα προσωπικά στοιχεία των μαθητών και ο αριθμός μητρώου εισάγονται από τη φόρμα οθόνης στο αρχείο μαθητών.

Ροή (7): Ο βαθμός διαγωνίσματος των μαθητών εισάγεται στη φόρμα οθόνης.

Ροή (9): Τα προσωπικά στοιχεία των μαθητών (όνομα, επίθετο, όνομα πατέρα, τμήμα) χρησιμοποιούνται για έλεγχο.

Ροή (10): Ο βαθμός διαγωνίσματος των μαθητών εισάγεται από τη φόρμα οθόνης στο αρχείο μαθητών.

Ροή (13): Τα στοιχεία Βαθμός και Ημερομηνία Διαγωνίσματος χρησιμοποιούνται για τη δημιουργία του καταλόγου βαθμών ανά τμήμα.

Ροή (14): Τα στοιχεία Τμήμα, Όνομα, Επίθετο χρησιμοποιούνται για τη δημιουργία του καταλόγου βαθμών ανά τμήμα.

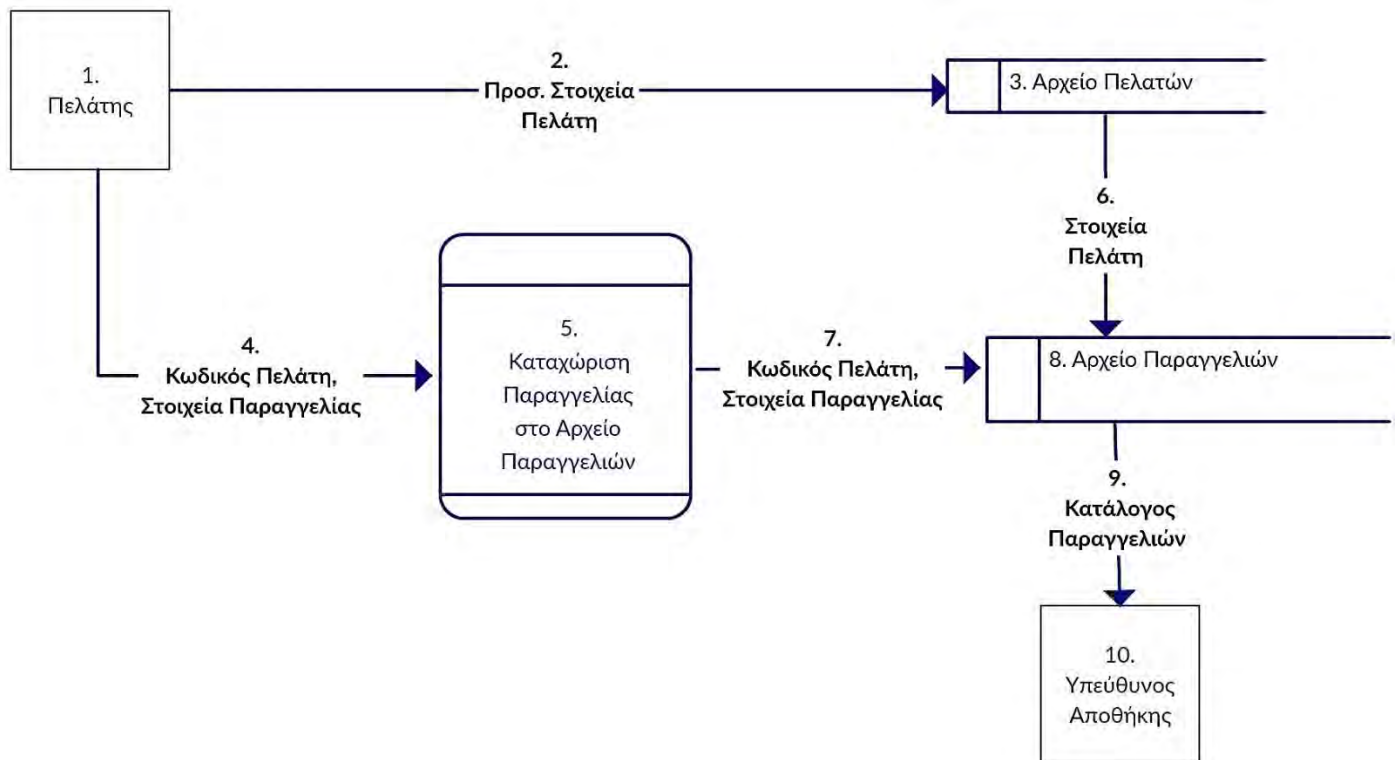
Ροή (15): Ο Κατάλογος Βαθμών δίδεται στον καθηγητή.



Ασκήσεις Κεφαλαίου

Άσκηση 3.1

Σας δίδεται πιο κάτω ένα μέρος Διαγράμματος Ροής Δεδομένων που έχει σχεδιαστεί για ένα πληροφοριακό σύστημα. Ποιες από τις ροές δεδομένων του διαγράμματος δεν είναι έγκυρες (να τις αναφέρετε με τον αριθμό τους) και γιατί; Να περιγράψετε τον σωστό τρόπο με τον οποίο πρέπει να διεξάγεται η λειτουργία τους.

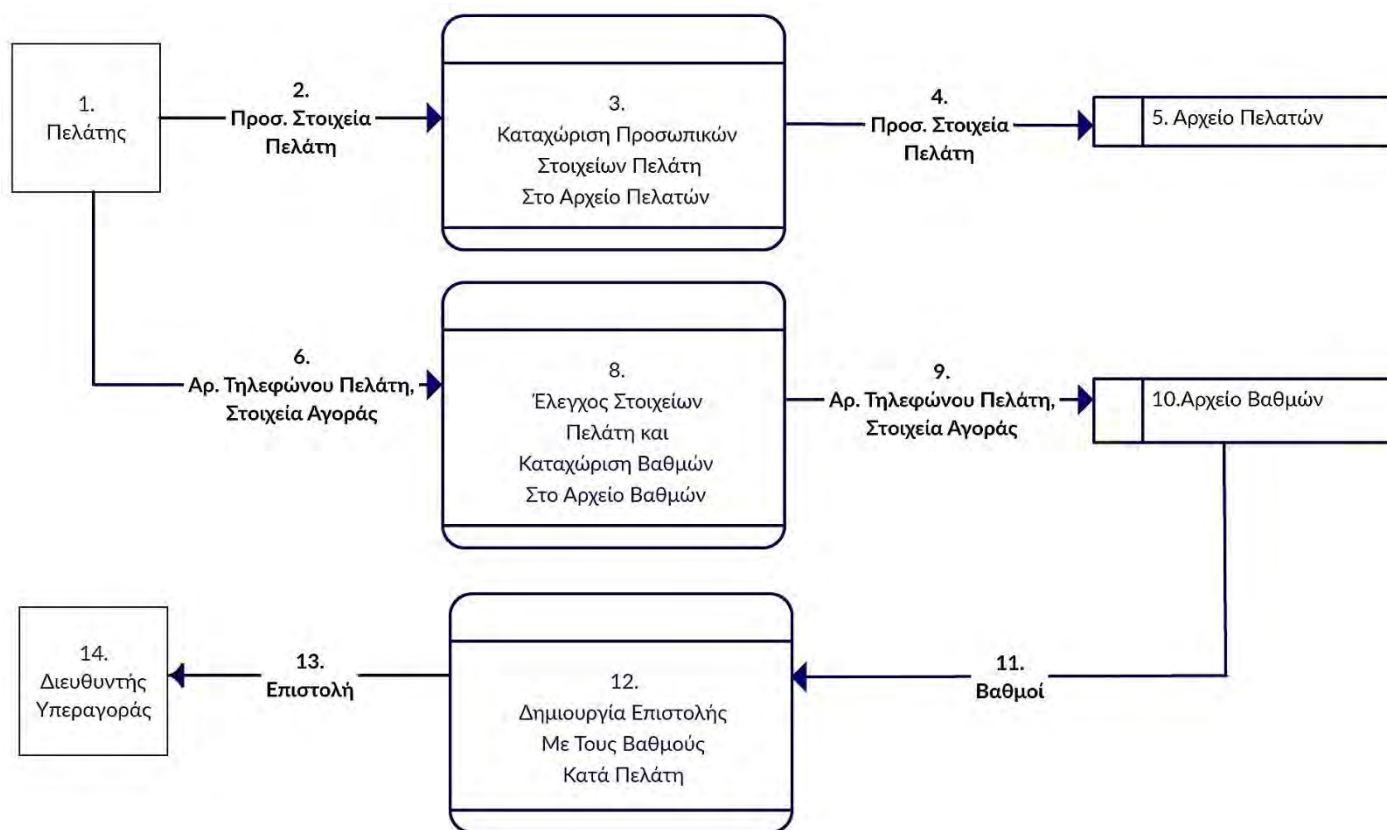


Άσκηση 3.2

Το πιο κάτω Διάγραμμα Ροής Δεδομένων (ΔΡΔ) περιγράφει την ακόλουθη λειτουργία:

Η υπεραγορά «HOFER» δίνει βαθμούς στον κάθε πελάτη της με βάση τις αγορές που κάνει, τους οποίους μετά μπορεί να εξαργυρώσει με άλλες αγορές. Όταν ένας πελάτης είναι καινούργιος, τότε θα καταχωρισθούν τα προσωπικά του στοιχεία στο σύστημα (όνομα, επίθετο, διεύθυνση, αριθμός τηλεφώνου). Στη συνέχεια για κάθε αγορά που θα κάνει, θα δίνει τον αριθμό τηλεφώνου του και τα στοιχεία της αγοράς. Ακολούθως θα γίνει έλεγχος των προσωπικών του στοιχείων από το σύστημα και θα υπολογίζονται οι βαθμοί που θα πάρει, με βάση τις αγορές του και οι οποίοι βαθμοί θα καταχωρισθούν μαζί με τον αριθμό τηλεφώνου του στο αρχείο βαθμών. Στο τέλος κάθε μήνα, θα δημιουργείται και θα αποστέλλεται σε κάθε πελάτη ονομαστική επιστολή με τους διαθέσιμους βαθμούς που έχει.

Να αναφέρετε **τέσσερα (4)** από τα λάθη που υπάρχουν στο διάγραμμα και να τα δικαιολογήσετε.



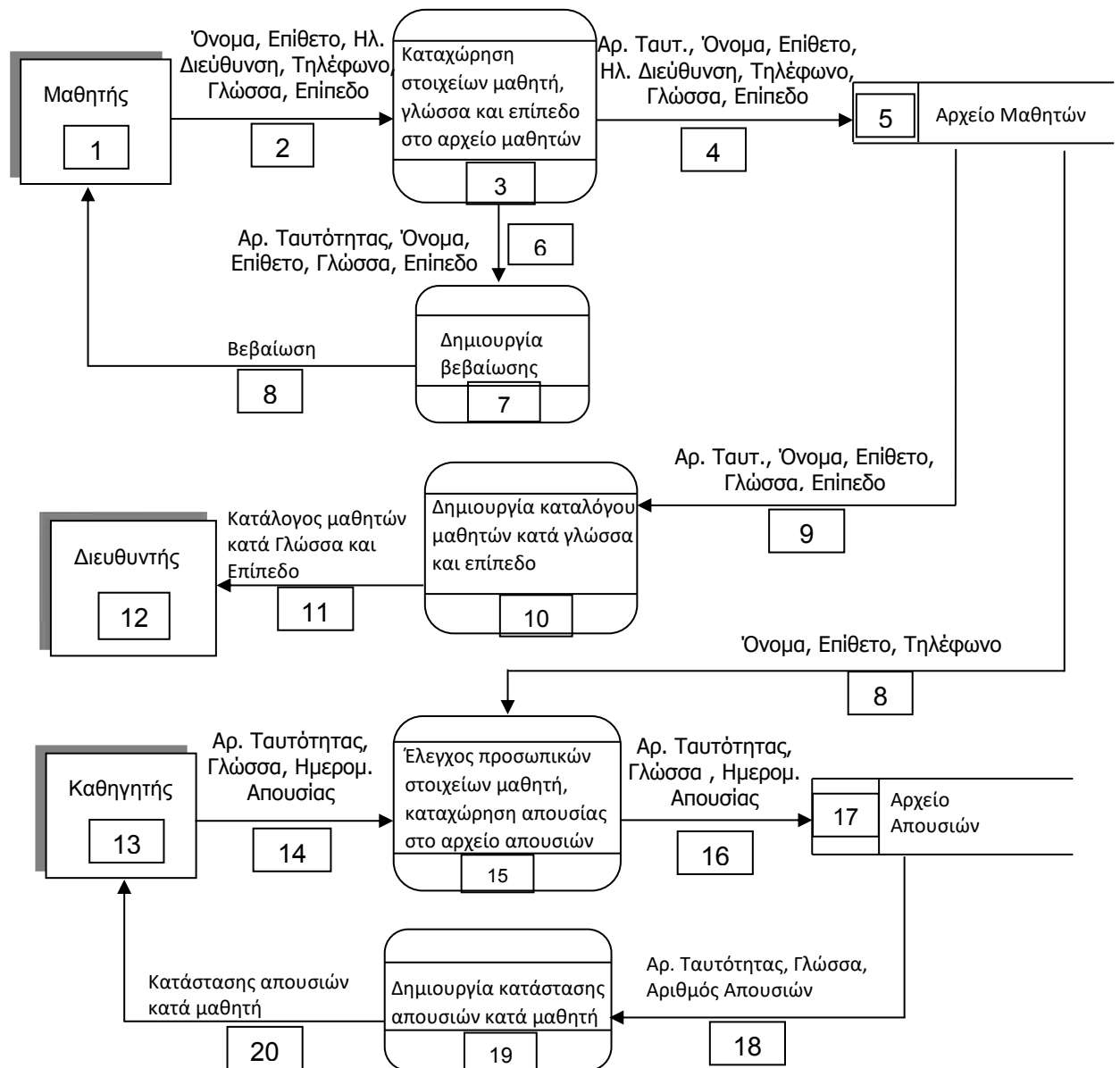
Άσκηση 3.3

Ένα ινστιτούτο ξένων γλωσσών διαθέτει σύστημα εγγραφής μαθητών. Κατά τις εγγραφές ο κάθε μαθητής δίνει τα προσωπικά του στοιχεία (αριθμό ταυτότητας, όνομα, επίθετο, ηλεκτρονική διεύθυνση, τηλέφωνο), τη γλώσσα και το επίπεδό της που θέλει να παρακολουθήσει, τα οποία καταχωρούνται στο αρχείο μαθητών. Ταυτόχρονα δημιουργείται μια βεβαίωση εγγραφής για τον κάθε μαθητή με τον αριθμό ταυτότητας, το όνομα, το επίθετο, τη γλώσσα και το επίπεδό της, η οποία δίνεται στο μαθητή. Όταν τελειώσουν οι εγγραφές, ετοιμάζεται κατάλογος όλων των μαθητών του ινστιτούτου κατά γλώσσα και επίπεδο, που περιλαμβάνει τον αριθμό ταυτότητας, το όνομα και το επίθετο του κάθε μαθητή και ο οποίος δίνεται στο διευθυντή.

Κατά τη διάρκεια των μαθημάτων οι καθηγητές παίρνουν παρουσίες και στο τέλος του μαθήματος δίνουν τα στοιχεία των μαθητών που απουσίαζαν (αριθμός ταυτότητας, γλώσσα, ημερομηνία απουσίας) για να καταχωρηθούν στο αρχείο απουσιών, αφού προηγηθεί αυτόματος έλεγχος των προσωπικών τους στοιχείων (όνομα, επίθετο, τηλέφωνο). Στο τέλος κάθε μήνα ετοιμάζεται κατάσταση απουσιών κατά μαθητή, που περιλαμβάνει τον αριθμό ταυτότητας, το όνομα, το επίθετο, το τηλέφωνο, τη γλώσσα και τις ημερομηνίες που απουσίαζε και η οποία δίνεται επίσης στο διευθυντή, για να ενημερώσει σχετικά τους γονείς.

Πιο κάτω έχει σχεδιαστεί το Διάγραμμα Ροής Δεδομένων (ΔΡΔ) το οποίο περιγράφει τις πιο πάνω διαδικασίες.

Να αναφέρετε **τέσσερα (4)** από τα λάθη που υπάρχουν στο ΔΡΔ και να τα δικαιολογήσετε.



(Παγκύπριες Εξετάσεις 2016)

Άσκηση 3.4

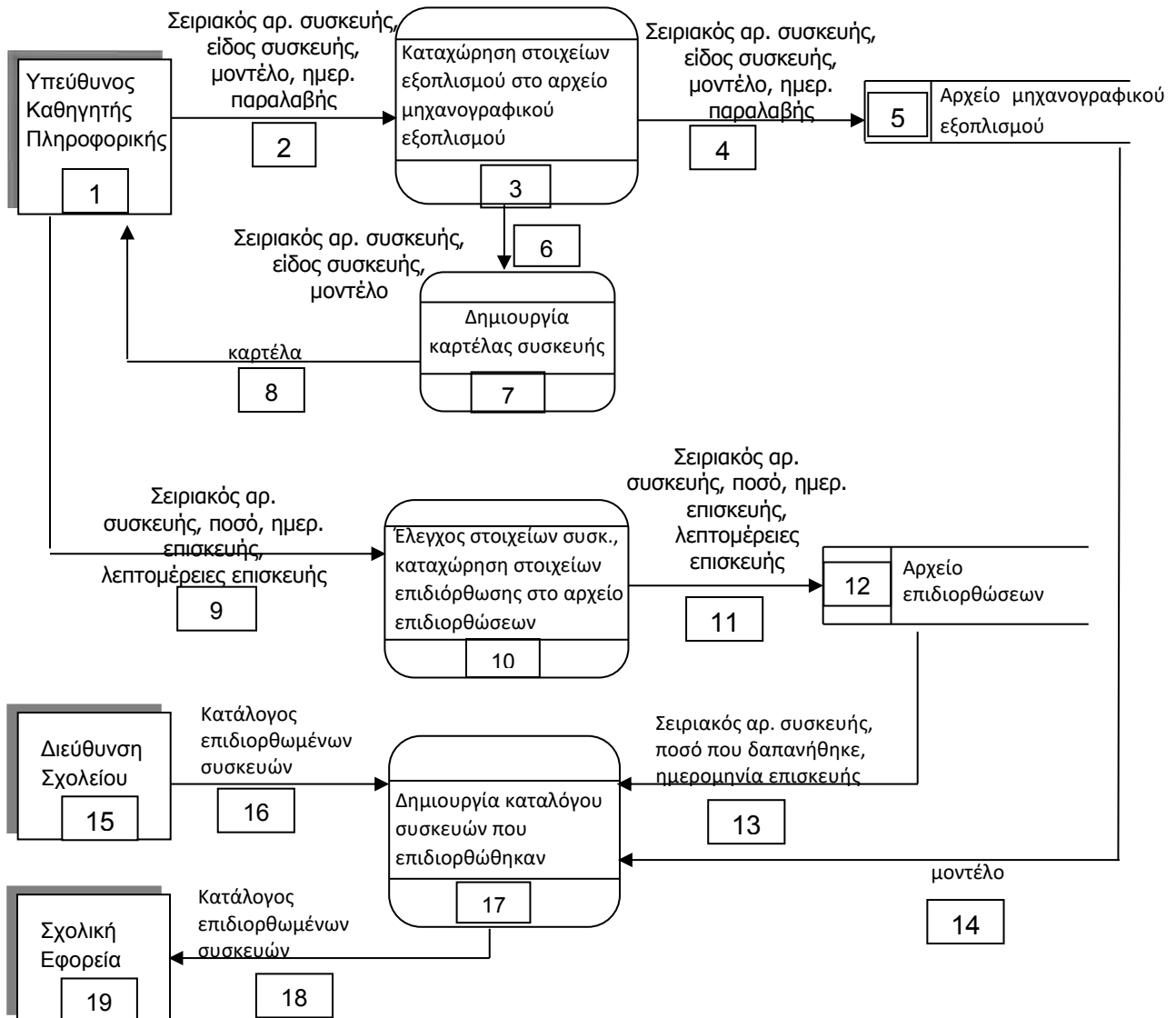
Ένα σχολείο διαθέτει αυτοματοποιημένο σύστημα καταγραφής του μηχανογραφικού του εξοπλισμού:

Τα στοιχεία του μηχανογραφικού εξοπλισμού του σχολείου (σειριακός αριθμός συσκευής, είδος συσκευής, μοντέλο, ημερομηνία παραλαβής) δίνονται από τον υπεύθυνο καθηγητή Πληροφορικής για να καταχωρηθούν στο αρχείο μηχανογραφικού εξοπλισμού. Κάθε φορά που καταχωρείται μια νέα συσκευή, το σύστημα εκτυπώνει μια καρτέλα με όλα τα στοιχεία της συσκευής, η οποία δίνεται στον υπεύθυνο καθηγητή Πληροφορικής για αρχειοθέτηση.

Όταν μια συσκευή επισκευάζεται λόγω βλάβης, ο υπεύθυνος καθηγητής Πληροφορικής δίνει το σειριακό αριθμό της συσκευής, το ποσό που δαπανήθηκε, την ημερομηνία επισκευής καθώς και λεπτομέρειες για την επισκευή που έγινε. Τα στοιχεία αυτά καταχωρούνται στο αρχείο επιδιορθώσεων, αφού προηγουμένως γίνει έλεγχος των στοιχείων της συσκευής (είδος συσκευής, μοντέλο) με βάση τον σειριακό αριθμό της.

Το σύστημα έχει τη δυνατότητα να δημιουργεί κατάλογο που περιλαμβάνει όλες τις συσκευές που επιδιορθώθηκαν (σειριακός αριθμός συσκευής, είδος συσκευής, ποσό που δαπανήθηκε, ημερομηνία επισκευής). Ο κατάλογος αυτός δίνεται τόσο στη διεύθυνση του σχολείου όσο και στη σχολική εφορεία. Πιο κάτω έχει σχεδιαστεί το Διάγραμμα Ροής Δεδομένων (ΔΡΔ) το οποίο περιγράφει τις πιο πάνω διαδικασίες.

Να αναφέρετε **τέσσερα (4)** από τα λάθη που υπάρχουν στο ΔΡΔ και να τα δικαιολογήσετε.



(Παγκύπριες Εξετάσεις 2017)

Άσκηση 3.5

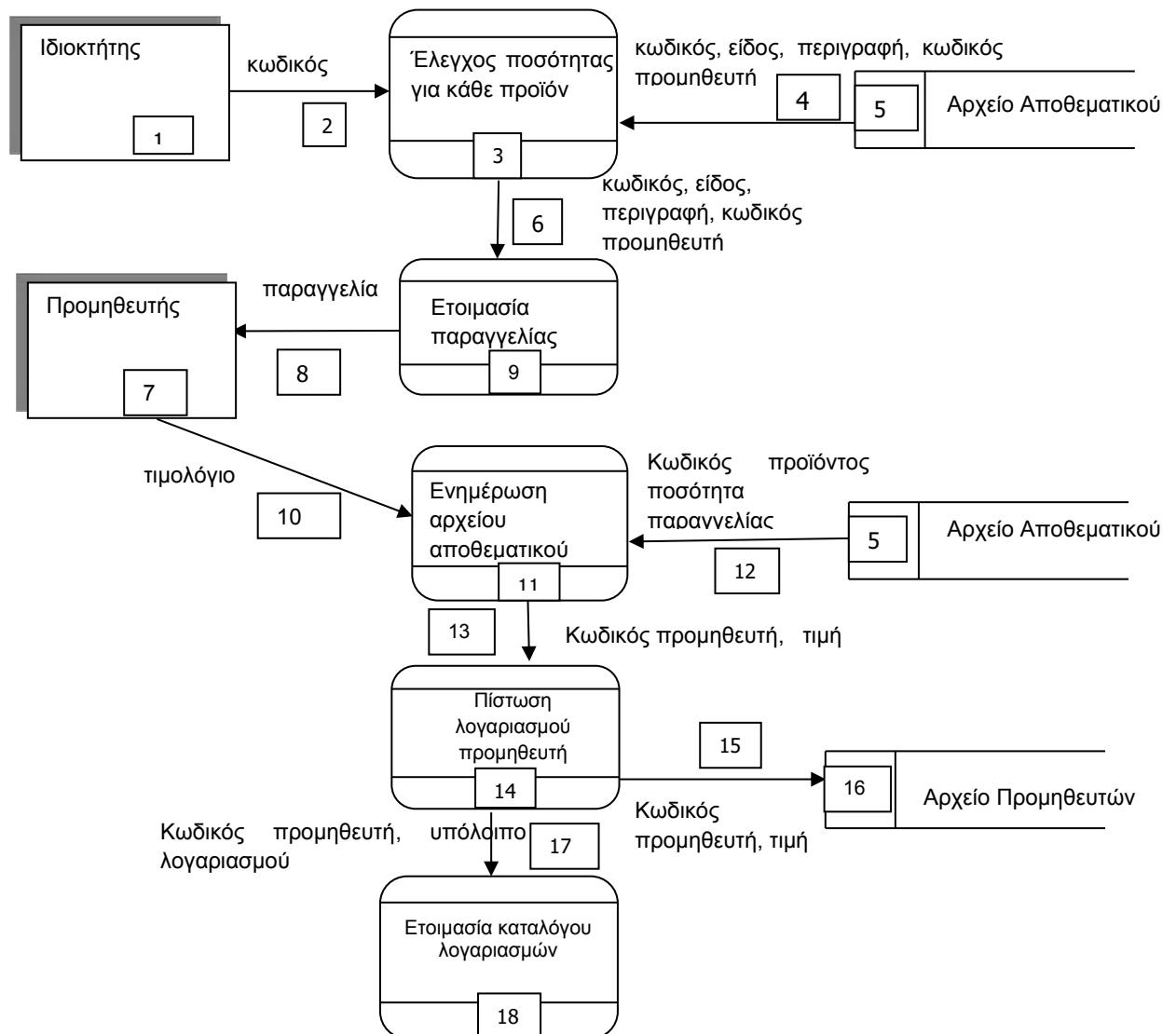
Το πιο κάτω Διάγραμμα Ροής Δεδομένων (ΔΡΔ) περιγράφει τη λειτουργία ενός συστήματος εισαγωγής ηλεκτρικών ειδών.

Ένας εισαγωγέας προμηθεύεται ηλεκτρικά είδη από διάφορους προμηθευτές. Διατηρεί για αυτό το σκοπό αρχείο αποθεματικού, το οποίο περιλαμβάνει τον κωδικό, το είδος, την περιγραφή και τη διαθέσιμη ποσότητα του προϊόντος, καθώς και τον κωδικό του προμηθευτή. Διατηρεί επίσης αρχείο προμηθευτών, το οποίο περιλαμβάνει τον κωδικό, το ονοματεπώνυμο, τη διεύθυνση, το τηλέφωνο, το φαξ, την ηλεκτρονική διεύθυνση και το υπόλοιπο λογαριασμού του προμηθευτή.

Στο τέλος κάθε εργάσιμης ημέρας, ο ιδιοκτήτης καταχωρεί τον κωδικό κάθε προϊόντος και γίνεται έλεγχος στο αρχείο αποθεματικού αν η διαθέσιμη ποσότητα για το κάθε προϊόν είναι μικρότερη από 5. Σε τέτοια περίπτωση τυπώνεται αυτόματα παραγγελία στην οποία αναγράφονται τα στοιχεία του προμηθευτή, ο κωδικός του προϊόντος και η ποσότητα παραγγελίας η οποία είναι πάντα 10 για κάθε προϊόν. Η παραγγελία αυτή αποστέλλεται στον προμηθευτή στην ηλεκτρονική του διεύθυνση ή στο φαξ.

Ο προμηθευτής στέλνει τα ηλεκτρικά είδη μαζί με το τιμολόγιο στο οποίο αναγράφεται ο κωδικός του προϊόντος, ο κωδικός του προμηθευτή, η ποσότητα και η τιμή. Αυτόματα με τον κωδικό κάθε προϊόντος ενημερώνεται το αρχείο αποθεματικού και αυξάνεται η ποσότητα. Επίσης, πιστώνεται η τιμή στο υπόλοιπο του λογαριασμού του προμηθευτή. Στο τέλος κάθε μήνα παραδίδεται στον ιδιοκτήτη της εταιρείας ένας κατάλογος με τα υπόλοιπα των λογαριασμών των προμηθευτών.

Να αναφέρετε **τρία (3)** από τα λάθη που υπάρχουν στο διάγραμμα.



(Παγκύπριες Εξετάσεις 2008)

Άσκηση 3.6

Μία σχολή χορού στην προσπάθεια της να εξυπηρετεί καλύτερα τους πελάτες της αποφάσισε να μηχανογραφηθεί. Κατά τη φάση «Εξακρίβωση Αναγκών και Καθορισμός Απαιτήσεων» έχει εξακριβωθεί ότι το πληροφοριακό σύστημα πρέπει να περιλαμβάνει και τις ακόλουθες λειτουργίες:

Κάθε νέος μαθητής θα δίνει τα στοιχεία του (αριθμός ταυτότητας, όνομα, επίθετο, διεύθυνση, τηλέφωνο και επίπεδο της τάξης του), τα οποία θα καταχωρίζονται στο αρχείο χορευτών. Το σύστημα θα πρέπει να περιλαμβάνει και τη δημιουργία καταλόγου (αναφοράς) με τα στοιχεία όλων των μαθητών κατά επίπεδο τάξης, που έχουν καταχωρισθεί στο αρχείο και η οποία παραδίδεται στη διεύθυνση της σχολής.

Να σχεδιάσετε το **Διάγραμμα Ροής Δεδομένων** (ΔΡΔ) για τη λειτουργία που περιγράφεται πιο πάνω.

Άσκηση 3.7

Το τμήμα διαχείρισης πληθυσμού του κράτους πρόκειται να μηχανογραφηθεί. Κατά τη φάση της Εξακρίβωσης Αναγκών και του Καθορισμού Απαιτήσεων, έχουν εξακριβωθεί τα πιο κάτω χαρακτηριστικά, τα οποία πρέπει να διαθέτει ένα μέρος του Πληροφοριακού Συστήματος:

Όταν γεννηθεί ένα παιδί, οι γονείς του θα πρέπει να προσέλθουν στο τμήμα και να δώσουν τα στοιχεία του παιδιού (όνομα, επίθετο, όνομα πατέρα, όνομα μητέρας, φύλο, ημερομηνία γέννησης και τόπος γέννησης), τα οποία μαζί με τον αριθμό ταυτότητας που θα δημιουργηθεί αυτόματα από το σύστημα, θα καταχωρισθούν στο αρχείο πληθυσμού. Σε μία εβδομάδα θα ετοιμάζεται το πιστοποιητικό γέννησης, το οποίο θα αποστέλλεται στους γονείς του παιδιού. Το πιστοποιητικό γέννησης θα περιλαμβάνει όλα τα πιο πάνω στοιχεία μαζί με τον αριθμό ταυτότητας.

Να σχεδιάσετε το **Διάγραμμα Ροής Δεδομένων** (ΔΡΔ) για τη λειτουργία που περιγράφηκε πιο πάνω και να περιγράψετε τους εξωτερικούς πράκτορες και τις διαδικασίες.

Άσκηση 3.8

Ένας ιατρός θέλει να δημιουργήσει ένα πληροφοριακό σύστημα, για να καταχωρίζει τους ασθενείς του και τις επισκέψεις τους. Κατά τη φάση της Εξακρίβωσης Αναγκών και του Καθορισμού Απαιτήσεων έχουν εξακριβωθεί τα πιο κάτω χαρακτηριστικά, τα οποία πρέπει να διαθέτει το σύστημα:

Για κάθε νέο ασθενή θα καταχωρίζονται τα προσωπικά του στοιχεία (αριθμός ταυτότητας, όνομα, επίθετο, διεύθυνση, τηλέφωνο) στο αρχείο ασθενών. Όταν ο ασθενής εξεταστεί από τον ιατρό, θα καταχωρισθούν στο αρχείο επισκέψεων τα στοιχεία της εξέτασης (αριθμός ταυτότητας, ημερομηνία επίσκεψης, λόγος επίσκεψης, διάγνωση, φάρμακα). Στο τέλος κάθε μέρας θα τυπώνεται κατάλογος με τους ασθενείς που εξετάστηκαν τη συγκεκριμένη μέρα από τον ιατρό και ο οποίος θα περιλαμβάνει την ημερομηνία επίσκεψης, το όνομα και επίθετο ασθενή, τον λόγο επίσκεψης, τη διάγνωση και τα φάρμακα. Ο κατάλογος αυτός θα δίδεται στον ιατρό.

Να σχεδιάσετε το **Διάγραμμα Ροής Δεδομένων** (ΔΡΔ) για τη λειτουργία που περιγράφηκε πιο πάνω.

Άσκηση 3.9

Το γυμναστήριο «Η ΔΥΝΑΜΗ» πρόκειται να μηχανογραφηθεί. Κατά τη φάση της Εξακρίβωσης Αναγκών και του Καθορισμού Απαιτήσεων έχουν εξακριβωθεί τα πιο κάτω χαρακτηριστικά, τα οποία πρέπει να διαθέτει το Πληροφοριακό Σύστημα:

Όταν ένας πελάτης προσέλθει για πρώτη φορά στο γυμναστήριο, θα πρέπει να δώσει τα προσωπικά του στοιχεία (ονοματεπώνυμο, φύλο, διεύθυνση, τηλέφωνο και ημερομηνία γεννήσεως), τα οποία μαζί με έναν κωδικό που θα δημιουργηθεί αυτόματα από το σύστημα, θα καταχωρισθούν στο αρχείο πελατών. Αφού καταχωρισθούν τα στοιχεία αυτά, θα εκδίδεται ατομική κάρτα μέλους, η οποία θα δίδεται στον πελάτη και η οποία θα του εξασφαλίζει την είσοδό του στο γυμναστήριο. Στη συνέχεια ο πελάτης θα εξετασθεί από τον αθλητίατρο του γυμναστηρίου, ο οποίος θα δώσει τα αποτελέσματα της εξέτασης μαζί με ένα πρόγραμμα ασκήσεων που πρέπει να ακολουθεί ο πελάτης, για να καταχωρισθούν στο αρχείο αθλουμένων του συστήματος (κωδικός πελάτη, αποτελέσματα εξέτασης και ασκήσεις). Την επόμενη φορά που ο πελάτης πάει στο γυμναστήριο, παρουσιάζει την κάρτα μέλους που περιλαμβάνει τον κωδικό του και ζητά να ετοιμαστεί το ατομικό του πρόγραμμα, με τα προσωπικά του στοιχεία και τις ασκήσεις του. Αυτό θα δημιουργηθεί από το σύστημα και θα δοθεί τόσο στον πελάτη όσο και στον γυμναστή του.

Να σχεδιάσετε το **Διάγραμμα Ροής Δεδομένων** (ΔΡΔ) για τη λειτουργία που περιγράφηκε πιο πάνω.

Άσκηση 3.10

Η εταιρεία «ΑΑ auto» ενοικιάζει αυτοκίνητα δύο τύπων (4Χ4 και σαλούν), σε ημερήσια βάση, σε τουρίστες που έρχονται στη Κύπρο με διαφήμιση στο Διαδίκτυο. Τα αυτοκίνητα παραλαμβάνονται μετά το μεσημέρι της ημέρας παραλαβής από τα γραφεία της εταιρείας στο αεροδρόμιο Λάρνακας και παραδίνονται πριν από το μεσημέρι της ημέρας παράδοσης στον ίδιο χώρο. Διατηρεί για το σκοπό αυτό, αρχείο αυτοκινήτων, το οποίο περιλαμβάνει στοιχεία των αυτοκινήτων (αρ. εγγραφής, τύπος οχήματος, μάρκα, μοντέλο, κυβικά, τιμή ενοικίασης), αρχείο διαθέσιμων αυτοκινήτων (αρ. εγγραφής, ημερομηνία) και αρχείο κρατήσεων (αρ. εγγραφής, ημερομηνία παραλαβής, ημερομηνία παράδοσης, αρ. διαβατηρίου του πελάτη).

Ο πελάτης καταχωρεί την ημερομηνία παραλαβής, την ημερομηνία παράδοσης και τον τύπο του αυτοκινήτου από τη φόρμα οθόνης στην ιστοσελίδα. Γίνεται αναζήτηση στο αρχείο αυτοκινήτων και στο αρχείο διαθέσιμων αυτοκινήτων και δημιουργείται κατάλογος των διαθέσιμων αυτοκινήτων που περιλαμβάνει τον αρ. εγγραφής, μάρκα, μοντέλο και τη τιμή ενοικίασης. Ο πελάτης μπορεί να κάνει την κράτησή του εισάγοντας τον αριθμό εγγραφής του αυτοκινήτου, την ημερομηνία παραλαβής, την ημερομηνία παράδοσης του αυτοκινήτου, τον αριθμό του διαβατηρίου και τον αριθμό της πιστωτικής του κάρτας. Τα στοιχεία αυτά καταχωρούνται στο αρχείο κρατήσεων και σχετική απόδειξη αποστέλλεται στον πελάτη. Παράλληλα διαγράφεται το αυτοκίνητο από το αρχείο διαθέσιμων αυτοκινήτων για αυτή την περίοδο. Στο τέλος της κάθε μέρας τυπώνεται για τον ιδιοκτήτη κατάλογος με τα αυτοκίνητα που πρέπει να παραδοθούν σε πελάτες αλλά και αυτά που θα επιστραφούν.

Να σχεδιάσετε το **Διάγραμμα Ροής Δεδομένων** (ΔΡΔ) για τη λειτουργία που περιγράφηκε πιο πάνω.

(Παγκύπριες Εξετάσεις 2008)

Άσκηση 3.11

Όταν ένας νέος πελάτης ενδιαφέρεται να ασφαλίσει το αυτοκίνητο του, δίνει στον ασφαλιστή το όνομα του, την ταυτότητα, την ηλικία, το τηλέφωνο, τη διεύθυνση, και τα στοιχεία από αφορούν το αυτοκίνητο (αρ. εγγραφής, μάρκα, μοντέλο, κυβικά) και αυτός με τη σειρά του τα καταχωρεί με φόρμα οθόνης στο αρχείο πελατών. Επίσης, ο ασφαλιστής ετοιμάζει ένα προσωρινό σημείωμα με βάση τα στοιχεία του πελάτη και το δίνει στον πελάτη.

Μετά από λίγες μέρες, ο ασφαλιστής ετοιμάζει το ασφαλιστικό συμβόλαιο με βάση τα στοιχεία που του έδωσε ο πελάτης και καταχωρεί τον αριθμό της ασφάλειας, τον αριθμό της ταυτότητας του πελάτη, την ημερομηνία έναρξης και λήξης της ασφάλειας και το ποσό που θα πληρώσει ο πελάτης, στο αρχείο ασφαλειών. Αφού ετοιμαστεί το συμβόλαιο, ειδοποιείται ο πελάτης και προσέρχεται στο γραφείο για να πληρώσει. Ο πελάτης δίνει στον ασφαλιστή το ποσό και παραλαμβάνει από αυτόν το ασφαλιστικό συμβόλαιο και την απόδειξη πληρωμής. Κάθε τέλος του μήνα ετοιμάζεται κατάλογος, που περιλαμβάνει το όνομα, τη διεύθυνση των πελατών την ημερομηνία λήξης και το ποσό των ασφαλειών, για όλες τις ασφάλειες που λήγουν τον επόμενο μήνα και παραδίδεται στον ασφαλιστή.

Να σχεδιάσετε το **Διάγραμμα Ροής Δεδομένων** (ΔΡΔ) για τη λειτουργία που περιγράφηκε πιο πάνω.

Άσκηση 3.12

Η Πάφος επιλέχθηκε ως η πολιτιστική πρωτεύουσα της Ευρώπης για το έτος 2017. Για την οργάνωση και διεκπεραίωση του σημαντικού αυτού γεγονότος, το Δημοτικό Συμβούλιο της πόλης αποφάσισε να δημιουργήσει αυτοματοποιημένο σύστημα για την καταχώρηση των εθελοντών και των εργασιών που θα αναλάβει ο κάθε εθελοντής. Κατά τη φάση της «Εξακρίβωσης Αναγκών και Καθορισμού Απαιτήσεων», έχει εξακριβωθεί ότι το Πληροφοριακό Σύστημα πρέπει να περιλαμβάνει τις ακόλουθες λειτουργίες:

Κάθε πολίτης που ενδιαφέρεται να γίνει εθελοντής θα υποβάλει αίτηση μέσω μιας φόρμας οθόνης που είναι αναρτημένη στο διαδίκτυο, καταχωρώντας τα προσωπικά του στοιχεία (αριθμό ταυτότητας, όνομα, επίθετο, φύλο, ηλικία, ηλεκτρονικό ταχυδρομείο, επίπεδο μόρφωσης, ξένες γλώσσες που γνωρίζει και ενδιαφέροντα). Τα στοιχεία αυτά μαζί με ένα κωδικό εθελοντή που θα δημιουργείται αυτόματα από το σύστημα, θα καταχωρούνται στο αρχείο εθελοντών. Ο κωδικός εθελοντή θα δίνεται και στον πολίτη.

Στο τέλος κάθε εβδομάδας, το σύστημα θα δημιουργεί κατάλογο με τους κωδικούς και όλα τα προσωπικά στοιχεία των πολιτών που δήλωσαν ενδιαφέρον τη συγκεκριμένη εβδομάδα για εθελοντές. Ο κατάλογος αυτός θα δίνεται στο Γραμματέα του Δημοτικού Συμβουλίου. Ακολούθως, αφού γίνει κατανομή των εργασιών σύμφωνα με τα στοιχεία του κάθε ενδιαφερόμενου, ο Γραμματέας θα δίνει τον κωδικό του εθελοντή και τον τομέα εργασίας που θα εργαστεί και αφού πρώτα γίνει έλεγχος των στοιχείων του εθελοντή από το σύστημα (αριθμός ταυτότητας, όνομα, επίθετο), θα καταχωρούνται τα στοιχεία αυτά (κωδικός εθελοντή και τομέας εργασίας) μαζί με την ημερομηνία καταχώρησης (δίνεται αυτόματα από το σύστημα) στο αρχείο εργασιών. Ακολούθως, θα δημιουργείται επιστολή εθελοντισμού, η οποία θα αποστέλλεται στον πολίτη μέσω του ηλεκτρονικού του ταχυδρομείου για να τον ενημερώνει για τον τομέα εργασίας που επιλέχθηκε να εργαστεί. Η επιστολή θα περιλαμβάνει

τον κωδικό εθελοντή, το όνομα, το επίθετο, τον τομέα εργασίας του και την ημερομηνία καταχώρησης.

Το σύστημα επίσης θα δημιουργεί κάθε τέλος του μήνα συγκεντρωτικό κατάλογο με τα στοιχεία όλων των εθελοντών (κωδικός εθελοντή, όνομα, επίθετο, ηλεκτρονικό ταχυδρομείο, τομέας εργασίας), ο οποίος θα δίνεται στο Δημοτικό Συμβούλιο.

Να σχεδιάσετε το **Διάγραμμα Ροής Δεδομένων** (ΔΡΔ) για τη λειτουργία που περιγράφεται πιο πάνω.

(Παγκύπριες Εξετάσεις 2016)

Άσκηση 3.13

Μια εταιρεία αποφάσισε να αυτοματοποιήσει τη διαδικασία ελέγχου του χρόνου προσέλευσης και αποχώρησης των υπαλλήλων της από τον χώρο εργασίας. Για τον σκοπό αυτό, η εταιρεία θα εγκαταστήσει ειδικές μηχανές χρονομέτρησης που θα εγκατασταθούν στις εισόδους της εταιρείας και οι οποίες θα είναι συνδεδεμένες με ένα Πληροφοριακό Σύστημα. Οι υπάλληλοι, τόσο κατά την προσέλευσή τους στην εργασία, όσο και κατά την αποχώρησή τους από αυτή, θα τοποθετούν την κάρτα υπαλλήλου σε μια από αυτές τις μηχανές. Κατά τη φάση της «Εξακρίβωσης Αναγκών και Καθορισμού Απαιτήσεων», έχει εξακριβωθεί ότι το Πληροφοριακό Σύστημα πρέπει να περιλαμβάνει τις ακόλουθες λειτουργίες:

Για κάθε υπάλληλο θα καταχωρούνται στο αρχείο υπαλλήλων τα εξής στοιχεία: αριθμός ταυτότητας, ονοματεπώνυμο, φύλο, αριθμός τηλεφώνου και ηλεκτρονικό ταχυδρομείο. Την ίδια στιγμή θα δημιουργείται και κάρτα υπαλλήλου, η οποία θα περιλαμβάνει τον αριθμό ταυτότητας και το ονοματεπώνυμο του υπαλλήλου και θα δίνεται στον υπάλληλο.

Κατά την προσέλευση ή την αποχώρησή του στην/από την εταιρεία, ο υπάλληλος θα τοποθετεί την κάρτα του σε μια από τις μηχανές και θα καταχωρείται στο αρχείο χρονομέτρησης, ο αριθμός ταυτότητας του υπαλλήλου (από την κάρτα), η ημερομηνία, η ώρα προσέλευσης ή αποχώρησης του υπαλλήλου.

Στο τέλος κάθε ημέρας θα δημιουργείται κατάλογος με τους υπαλλήλους που καθυστέρησαν να προσέλθουν στην εργασία τους, ο οποίος θα περιλαμβάνει τον αριθμό ταυτότητας, την ημερομηνία και την ώρα προσέλευσης. Ο κατάλογος αυτός θα δίνεται στο τμήμα προσωπικού.

Στο τέλος κάθε βδομάδας θα δημιουργείται εβδομαδιαία κατάσταση προσέλευσης / αποχώρησης για κάθε υπάλληλο, που θα περιλαμβάνει τον αριθμό ταυτότητας, το ονοματεπώνυμο, την ημερομηνία, την ώρα προσέλευσης και την ώρα αποχώρησης του για όλες τις ημέρες της βδομάδας. Η κατάσταση θα δίνεται τόσο στον υπάλληλο όσο και στον τμηματάρχη του.

Να σχεδιάσετε το **Διάγραμμα Ροής Δεδομένων** (ΔΡΔ) για τη λειτουργία που περιγράφεται πιο πάνω.

(Παγκύπριες Εξετάσεις 2017)

B6.4 Διαγράμματα Οντοτήτων-Συσχετίσεων (ΔΟΣ) – Entity Relationship Diagrams (ERDs)

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- Να αναγνωρίζουμε ένα Διάγραμμα Οντοτήτων-Συσχετίσεων και να αναφέρουμε ποιες πτυχές του πληροφοριακού συστήματος παρουσιάζει
- Να αναγνωρίζουμε τα σύμβολα ενός Διαγράμματος Οντοτήτων-Συσχετίσεων, καθώς και τα βασικά τους γνωρίσματα (ιδιότητες, πρωτεύον κλειδί) και να αναφέρουμε τον ρόλο του καθενός
- Να αναγνωρίζουμε τα βασικά δομικά στοιχεία στην περιγραφή ενός πληροφοριακού συστήματος και να σχεδιάζουμε ένα Διάγραμμα Οντοτήτων-Συσχετίσεων, που να ταιριάζει στην περιγραφή
- Να εντοπίζουμε σφάλματα σε ένα Διάγραμμα Οντοτήτων-Συσχετίσεων και να προτείνουμε τρόπους διόρθωσής τους.

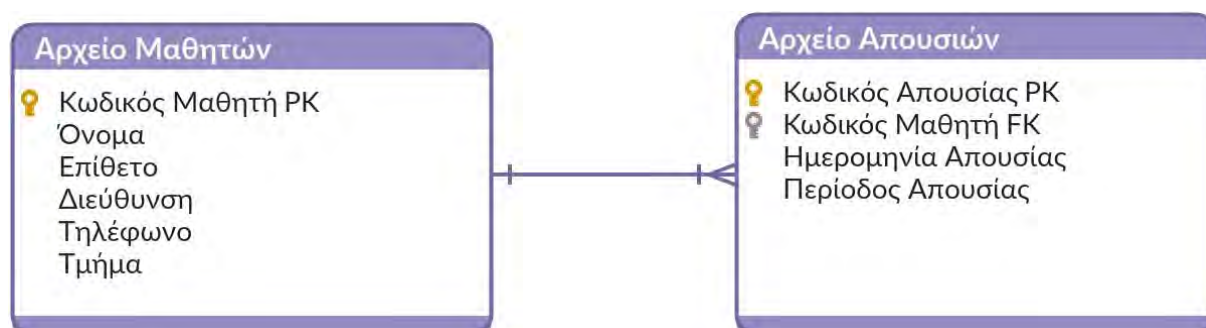
Μία επίσης ιδιαίτερα προσφιλή μέθοδος καταγραφής των προδιαγραφών του συστήματος, είναι η χρήση του Διαγράμματος Οντοτήτων-Συσχετίσεων (ΔΟΣ). Σε αυτό δίδεται σχηματικά η μορφή των δεδομένων, τα οποία διακινούνται μέσα σε ένα πληροφοριακό σύστημα. Είναι μέθοδος μοντελοποίησης δεδομένων. Είναι, κυρίως, λογικά μοντέλα (φανερώνουν τι κάνει το σύστημα), διότι δεν κάνουν καμιά αναφορά στον τρόπο υλοποίησης ή χρήσης των δεδομένων. Εντούτοις όμως, επειδή μπορούν εύκολα να μετατραπούν σε βάσεις δεδομένων, είναι σε κάποιον βαθμό και φυσικά μοντέλα (δηλ. φανερώνουν πώς το σύστημα κάνει κάτι).

1. Οντότητα (Αρχείο)

Μία οντότητα (αρχείο) περιγράφει σύνολα από παρόμοιες πληροφορίες. Για παράδειγμα ένα σύστημα σχολείου έχει μαθητές, καθηγητές, μαθήματα, τμήματα κ.λπ. Αυτά μπορούν να μοντελοποιηθούν ως οντότητες. Για κάθε οντότητα αποθηκεύονται κάποια δεδομένα. Για παράδειγμα, για κάθε μαθητή αποθηκεύονται τα εξής στοιχεία: κωδικός μαθητή (είναι μοναδικός), όνομα, επίθετο, φύλο, όνομα πατέρα, διεύθυνση, αριθμός τηλεφώνου, ημερομηνία γέννησης. Οι οντότητες σχετίζονται μεταξύ τους.

2. Σχέσεις (Relationships)

Οι σχέσεις είναι συσχετισμοί μεταξύ μίας ή περισσοτέρων οντοτήτων. Παριστάνουν κάποια σύνδεση, η οποία υπάρχει μεταξύ των οντοτήτων αυτών. Μία σχέση απεικονίζεται με μία γραμμή μεταξύ των συσχετιζόμενων οντοτήτων, όπως πιο κάτω:



Η σχέση του παραπάνω σχεδιαγράμματος καθορίζει ότι:

- Ένας μαθητής μπορεί να έχει μία ή περισσότερες απουσίες
- Μία απουσία γίνεται από έναν μόνο μαθητή

3. Πληθυκότητα (Cardinality)

Η πληθυκότητα ορίζει τον μέγιστο και ελάχιστο αριθμό περιπτώσεων, που μπορεί μία οντότητα να λάβει μέρος σε μία σχέση. Ορισμένοι συμβολισμοί που χρησιμοποιούμε για να απεικονίσουμε πληθυκότητες μεταξύ οντοτήτων είναι οι πιο κάτω:

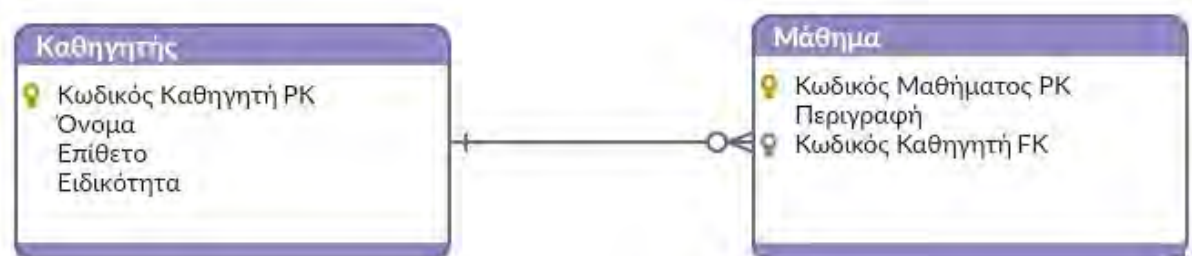
	Ακριβώς μία φορά
	Μία ή περισσότερες φορές
	Μηδέν, μία ή περισσότερες φορές

4. Παραδείγματα Διαγραμμάτων Οντοτήτων-Συσχετίσεων

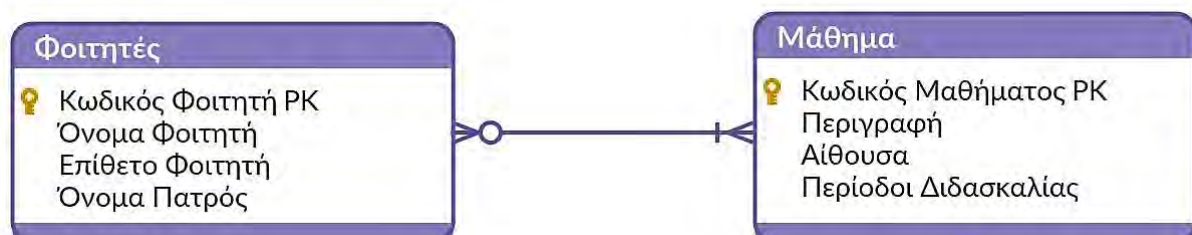
4.1. Ένας καθηγητής είναι υπεύθυνος σε ένα μόνο τμήμα, ένα τμήμα έχει υπεύθυνο μόνο έναν καθηγητή.



4.2. Ένας καθηγητής διδάσκει μηδέν ή περισσότερα μαθήματα, ένα μάθημα διδάσκεται από έναν μόνο καθηγητή.



4.3. Ένας φοιτητής επιλέγει ένα ή περισσότερα μαθήματα, ένα μάθημα επιλέγεται από μηδέν ή περισσότερους φοιτητές.



Η πιο πάνω σχέση (Πολλά προς Πολλά), δεν μπορεί να υλοποιηθεί απευθείας σε σχεσιακές βάσεις δεδομένων. Το πώς υλοποιείται θα το δούμε σε επόμενο κεφάλαιο.

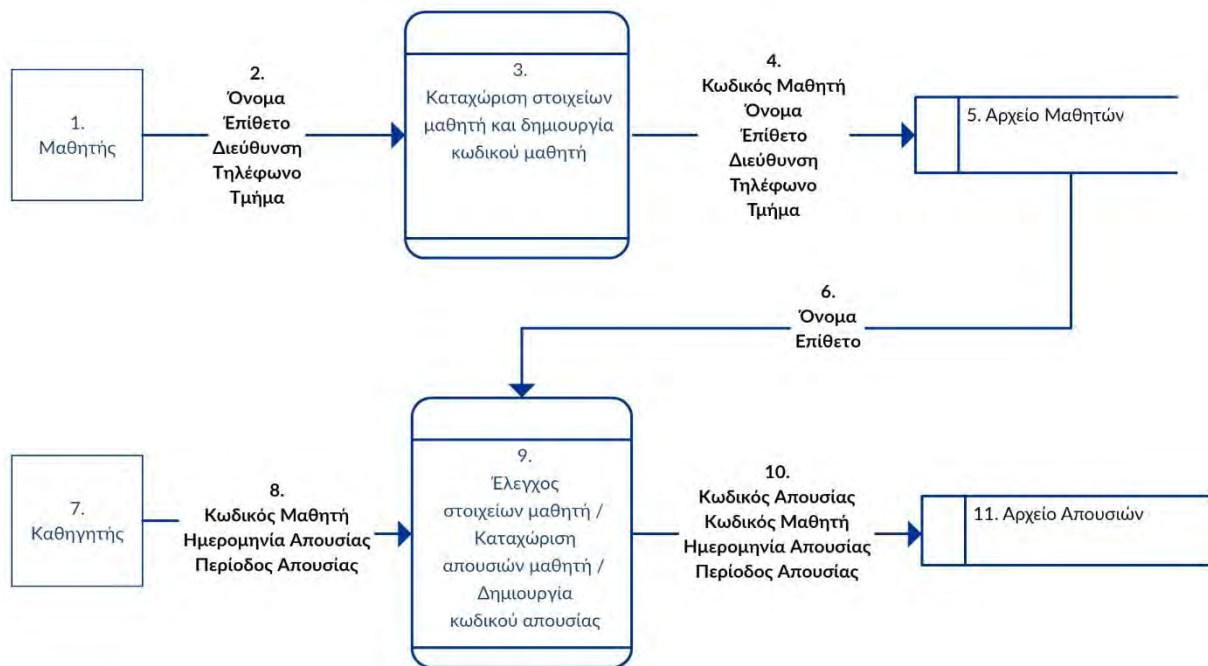
PK - Primary Key (Πρωτεύον Κλειδί): Ένα **πρωτεύον κλειδί** αποτελείται από ένα ή περισσότερα πεδία, που προσδιορίζουν τη μοναδικότητα κάθε εγγραφής μίας οντότητας (αρχείου). Κάθε οντότητα θα πρέπει να έχει ένα πρωτεύον κλειδί.

FK - Foreign Key (Ξένο κλειδί): **Δευτερεύον** κλειδί μίας οντότητας (αρχείου) είναι το πεδίο της οντότητας, το οποίο προσδιορίζει εγγραφές μίας άλλης οντότητας.

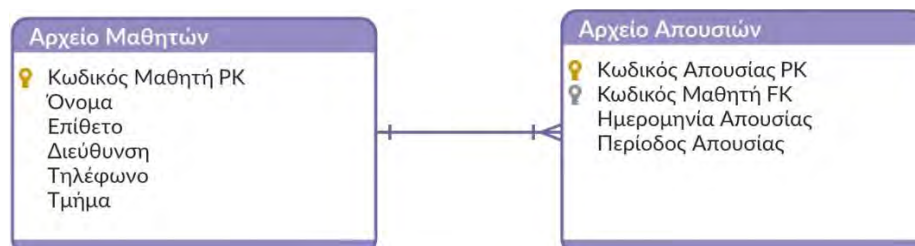
Παράδειγμα 4.1

Ένα σύστημα διαχείρισης σχολείου εκτελεί τις πιο κάτω διαδικασίες: Κατά τη διάρκεια των εγγραφών της Β' Λυκείου, ένας μαθητής δίνει στους υπεύθυνους για τις εγγραφές τα προσωπικά του στοιχεία που περιλαμβάνουν το όνομα, το επίθετο, τη διεύθυνση, το τηλέφωνο και το τμήμα. Τα στοιχεία αυτά, μαζί με έναν κωδικό μαθητή που δημιουργείται αυτόματα, καταχωρίζονται στο αρχείο των μαθητών. Κατά τη διάρκεια της χρονιάς, ένας καθηγητής ο οποίος θέλει να βάλει απουσία σε έναν μαθητή, καταχωρίζει τον κωδικό του μαθητή στο σύστημα και αφού ελέγξει τα προσωπικά στοιχεία του μαθητή (όνομα, επίθετο), καταχωρίζει την ημερομηνία και την περίοδο της απουσίας που μαζί με τον κωδικό απουσίας, που δημιουργείται αυτόματα από το σύστημα, καταχωρίζονται στο αρχείο των απουσιών. Να σχεδιάσετε το Διάγραμμα Ροής Δεδομένων και το Διάγραμμα Οντοτήτων-Συσχετίσεων για το πιο πάνω σύστημα.

Λύση



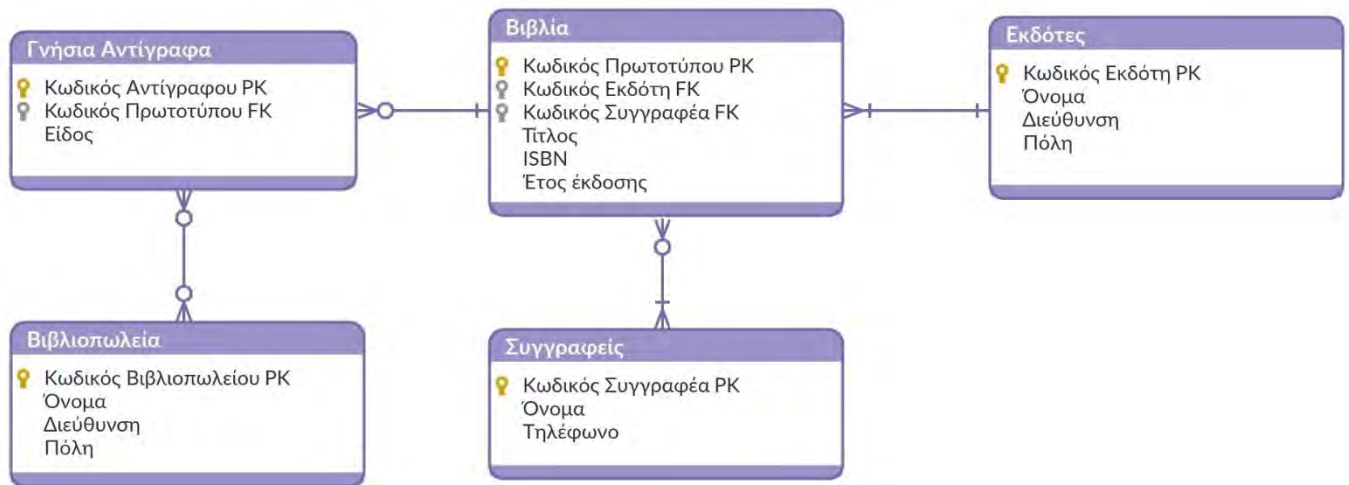
Εικόνα 1: ΔΡΔ με τη χρήση του εργαλείου Creately



Εικόνα 2: ΔΟΣ με τη χρήση του εργαλείου Creately

Παράδειγμα 4.2

Να περιγράψετε όλες τις σχέσεις του πιο κάτω Διαγράμματος Οντοτήτων Συσχετίσεων.

**Λύση**

- Κάθε βιβλίο έχει μηδέν ή περισσότερα γνήσια αντίγραφα.
- Κάθε γνήσιο αντίγραφο έχει ένα μόνο πρωτότυπο.
- Κάθε γνήσιο αντίγραφο μπορεί να βρίσκεται σε μηδέν ή περισσότερα βιβλιοπωλεία.
- Κάθε βιβλιοπωλείο μπορεί να έχει μηδέν ή περισσότερα αντίγραφα ενός βιβλίου.
- Κάθε εκδότης μπορεί να εκδώσει ένα ή περισσότερα βιβλία.
- Κάθε βιβλίο έχει έναν μόνο εκδότη.
- Κάθε συγγραφέας μπορεί να γράψει μηδέν ή περισσότερα βιβλία.
- Κάθε βιβλίο μπορεί να έχει έναν ή περισσότερους συγγραφείς.

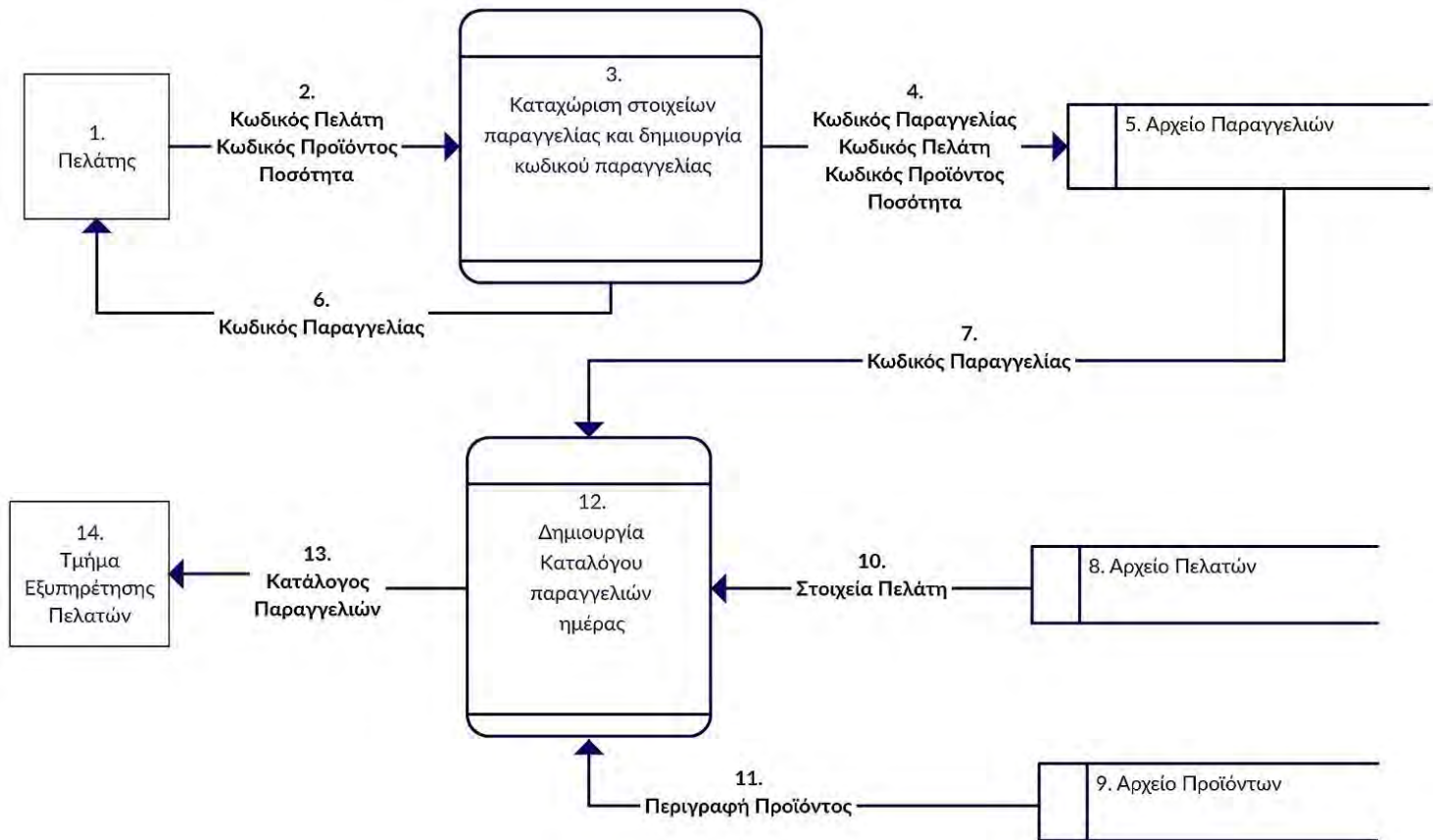
Παράδειγμα 4.3

Σε μία εμπορική εταιρεία διατηρούνται δεδομένα για τις παρακάτω οντότητες:

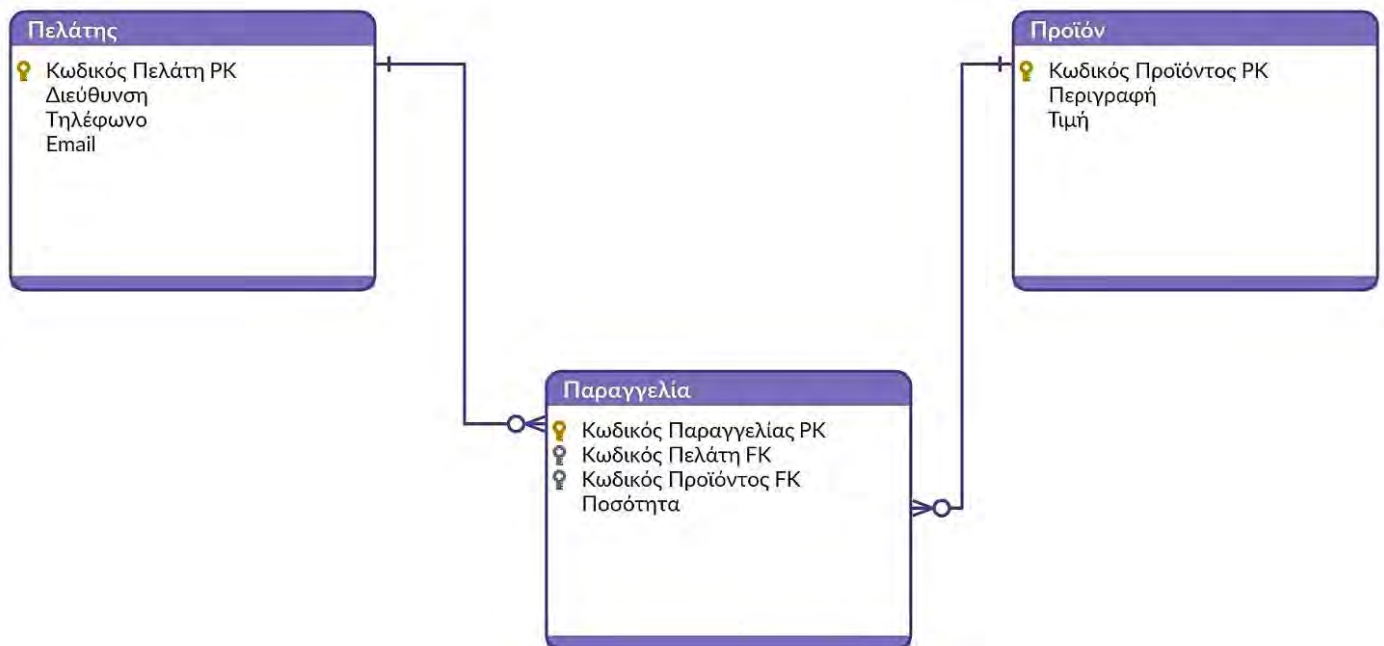
- (1) Πελάτης: κωδικός πελάτη, διεύθυνση, τηλέφωνο και ηλεκτρονικό ταχυδρομείο
- (2) Προϊόν: κωδικός προϊόντος, περιγραφή, τιμή

Όταν ένας πελάτης θέλει να παραγγείλει ένα προϊόν δίνει τον κωδικό του, τον κωδικό του προϊόντος και την ποσότητα. Τα στοιχεία αυτά μαζί με τον κωδικό παραγγελίας που δημιουργείται αυτόματα από το σύστημα καταχωρίζονται στο αρχείο παραγγελιών. Ο κωδικός παραγγελίας δίνεται στον πελάτη, ενώ στο τέλος της ημέρας δημιουργείται ένας κατάλογος που περιέχει τον κωδικό παραγγελίας, τα στοιχεία του πελάτη και την περιγραφή του προϊόντος για όλες τις παραγγελίες της ημέρας. Ο κατάλογος παραδίδεται στο τμήμα εξυπηρέτησεως πελατών.

Να σχεδιάσετε το Διάγραμμα Ροής Δεδομένων και το Διάγραμμα Οντοτήτων-Συσχετίσεων, που προκύπτει από τα πιο πάνω.

Λύση

Εικόνα 3: ΔΡΔ με τη χρήση του εργαλείου Creately



Εικόνα 4: ΔΟΣ με τη χρήση του εργαλείου Creately

**Ασκήσεις Κεφαλαίου****Άσκηση 4.1**

Να σχεδιάσετε το Διάγραμμα Οντοτήτων-Συσχετίσεων (ΔΟΣ), το οποίο απορρέει από το Διάγραμμα Ροής Δεδομένων (ΔΡΔ) που σχεδιάσατε στην **άσκηση 3.5**.

Άσκηση 4.2

Να σχεδιάσετε το Διάγραμμα Οντοτήτων-Συσχετίσεων (ΔΟΣ), το οποίο απορρέει από το Διάγραμμα Ροής Δεδομένων (ΔΡΔ) που σχεδιάσατε στην **άσκηση 3.6**.

Άσκηση 4.3

Στη γραμματεία του τμήματος ενός πανεπιστημίου διατηρούνται δεδομένα για τις παρακάτω οντότητες:

- (1) Μαθήματα: κωδικός μαθήματος, περιγραφή, έτος, εξάμηνο, ώρες και αίθουσες, μονάδες και προαπαιτούμενα
- (2) Φοιτητές: κωδικός φοιτητή, όνομα, διεύθυνση, email
- (3) Καθηγητές: κωδικός καθηγητή, όνομα και σχολή

Ένας φοιτητής μπορεί να επιλέξει ένα ή περισσότερα μαθήματα. Ένα μάθημα μπορεί να επιλεγεί από μηδέν ή περισσότερους φοιτητές. Ένας καθηγητής διδάσκει μηδέν ή περισσότερα μαθήματα, αλλά ένα μάθημα διδάσκεται από έναν μόνο καθηγητή.

Να σχεδιάσετε το **Διάγραμμα Οντοτήτων-Συσχετίσεων**, που προκύπτει από το πιο πάνω.

Άσκηση 4.4

Σε μία αεροπορική εταιρεία διατηρούνται δεδομένα για τις παρακάτω οντότητες:

- (1) Πελάτες: αριθμός ταυτότητας πελάτη, ονοματεπώνυμο, διεύθυνση, τηλέφωνο
- (2) Κρατήσεις: κωδικός κράτησης, ημερομηνία κράτησης
- (3) Πτήσεις: αριθμός πτήσης, κωδικός αεροσκάφους, ημερομηνία και ώρα αναχώρησης, ημερομηνία και ώρα άφιξης

Για κάθε πελάτη γίνεται μόνο μία κράτηση. Κάθε κράτηση μπορεί να γίνει για μία ή περισσότερες πτήσεις ενώ για μία πτήση γίνεται μόνο μία κράτηση.

Να σχεδιάσετε το **Διάγραμμα Οντοτήτων-Συσχετίσεων**, που προκύπτει από το πιο πάνω.

B6.5 Σχεδίαση Συστήματος

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ❖ Να μετατρέπουμε ένα διάγραμμα οντοτήτων-συσχετίσεων (ΔΟΣ) σε προδιαγραφές δημιουργίας σχετιζόμενων πινάκων, συγκεκριμένα:
 - Να διασπούμε κάθε σχέση Πολλά-Προς-Πολλά σε πίνακα και σχέσεις Ένα-Προς-Πολλά
 - Να καθορίζουμε τον τύπο δεδομένων για κάθε πεδίο και άλλες αναγκαίες απαιτήσεις (ιδιότητες)
 - Να ελέγχουμε/καθορίζουμε το πρωτεύον κλειδί σε κάθε πίνακα
 - Να ελέγχουμε/εισάγουμε για προσδιορισμό σχέσεων μεταξύ πινάκων
- ❖ Μετατρέπουμε το λογικό διάγραμμα ροής δεδομένων του πληροφοριακού συστήματος σε φυσικό, συγκεκριμένα:
 - Να ελέγχουμε/επαναπροσδιορίζουμε τους εξωτερικούς πράκτορες σύμφωνα με τον σχεδιασμό του συστήματος
 - Να ελέγχουμε/εισαγάγουμε ροές ή/και διαδικασίες από/προς εξωτερικούς πράκτορες, ώστε να υποδηλώνονται ξεκάθαρα οι φόρμες εισόδου και οι εκθέσεις/αναφορές εξόδου προς αυτούς
- ❖ Να εντοπίζουμε ποιες διαδικασίες αναφέρονται σε φόρμες (οθόνες) εισαγωγής δεδομένων και ποιες σε εκθέσεις (αναφορές)
- ❖ Να σχεδιάζουμε τις φόρμες (οθόνες) με έμφαση στα δεδομένα, που θα πρέπει να εμφανίζονται/εισάγονται και στη διάταξή τους
- ❖ Να σχεδιάζουμε τις εκθέσεις (αναφορές) με έμφαση στις πληροφορίες, που θα πρέπει να εμφανίζονται, στην ομαδοποίηση, ταξινόμηση και στη διάταξή τους
- ❖ Να καθορίζουμε τις προδιαγραφές ερωτημάτων, που χρειάζονται για τη δημιουργία των εκθέσεων (ποια πεδία, από ποιους πίνακες, με ποια ταξινόμηση και ποια κριτήρια).

1. Σχεδιασμός Οθονών (Forms) και Αναφορών (Reports)

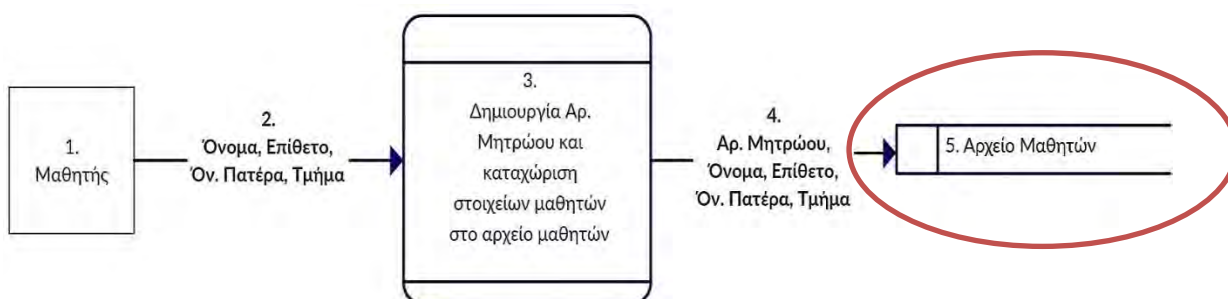
Όπως έχει ήδη αναφερθεί, η φάση Σχεδίαση του Συστήματος θα καθορίσει πώς το σύστημα θα κάνει αυτά που έχουν καταγραφεί στη φάση του καθορισμού προδιαγραφών. Κατά τη φάση αυτή, θα σχεδιαστούν οι βάσεις δεδομένων του συστήματος (πίνακες και σχέση μεταξύ τους), καθώς επίσης οι οθόνες εισαγωγής δεδομένων (Forms) και οι αναφορές (Reports).

Κάθε διαδικασία σε ένα ΔΡΔ είναι: είτε Οθόνη καταχώρησης και επεξεργασίας στοιχείων είτε δημιουργία Αναφοράς (κατάστασης) δεδομένων.


Κάθε αρχείο σε ένα ΔΡΔ είναι και μία αποθηκευτική μονάδα (πίνακας/αρχείο).

Παράδειγμα 1: Σχεδιασμός Οθόνης και δημιουργία Αποθηκευτικής Μονάδας

ΔΡΔ



Πίνακας Μαθητές

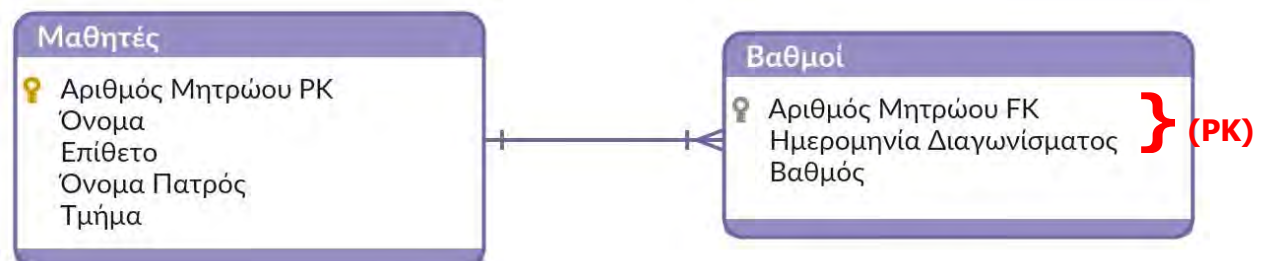
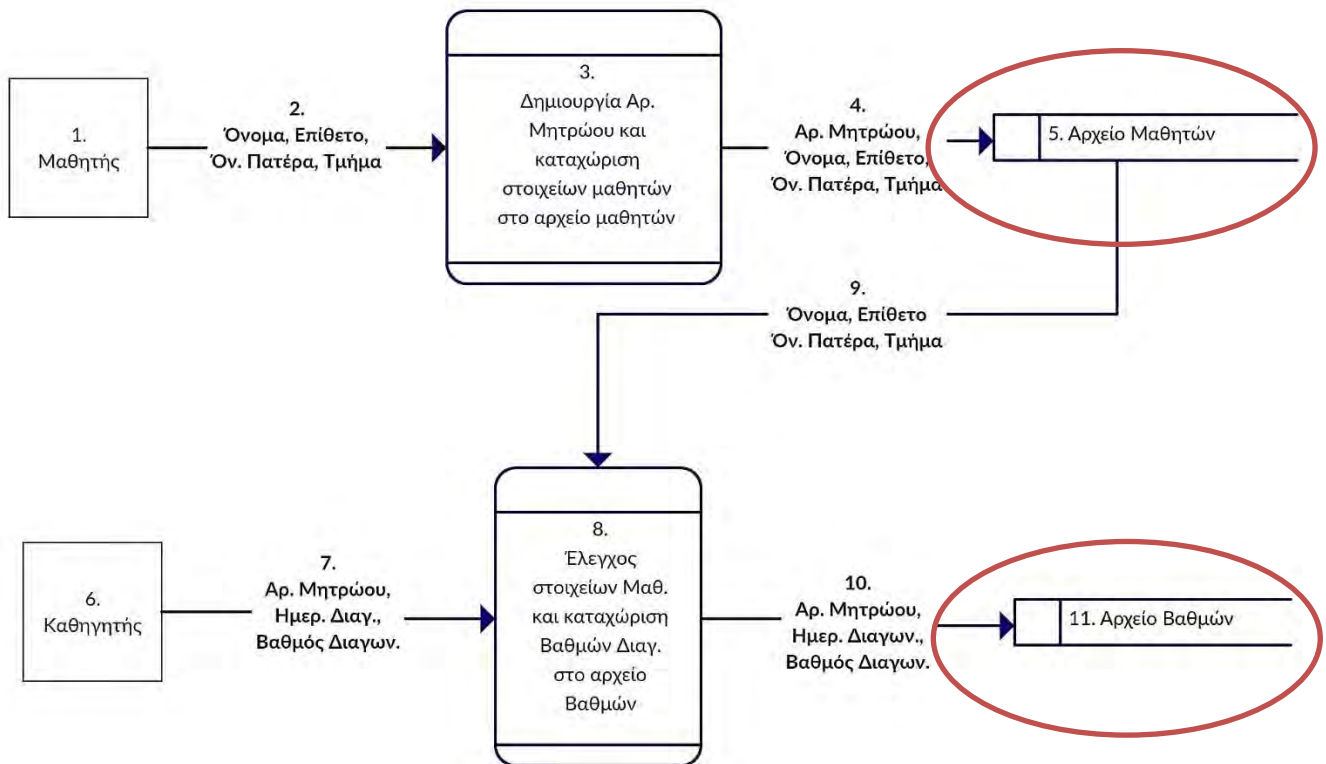
Μαθητές	
	Αριθμός Μητρώου PK
	Όνομα
	Επίθετο
	Όνομα Πατρός
	Τμήμα

PK: Primary Key – Πρωτεύον Κλειδί

Οθόνη (Φόρμα) Καταχώρισης Στοιχείων Μαθητή

Καταχώρηση Προσωπικών Στοιχείων Μαθητή	
Αριθμός Μητρώου:	<input type="text"/>
Όνομα:	<input type="text"/>
Επίθετο:	<input type="text"/>
Όνομα Πατέρα:	<input type="text"/>
Τμήμα:	<input type="text"/>
<input type="button" value="Καταχώρηση"/>	<input type="button" value="Έξοδος"/>

Παράδειγμα 2: Σχεδιασμός Οθόνης και δημιουργία Αποθηκευτικής Μονάδας

ΔΡΑΟθόνη ελέγχου στοιχείων Μαθητή και Καταχώρισης Βαθμού

Καταχώριση Βαθμού Μαθητή

Αρ. Μητρώου:

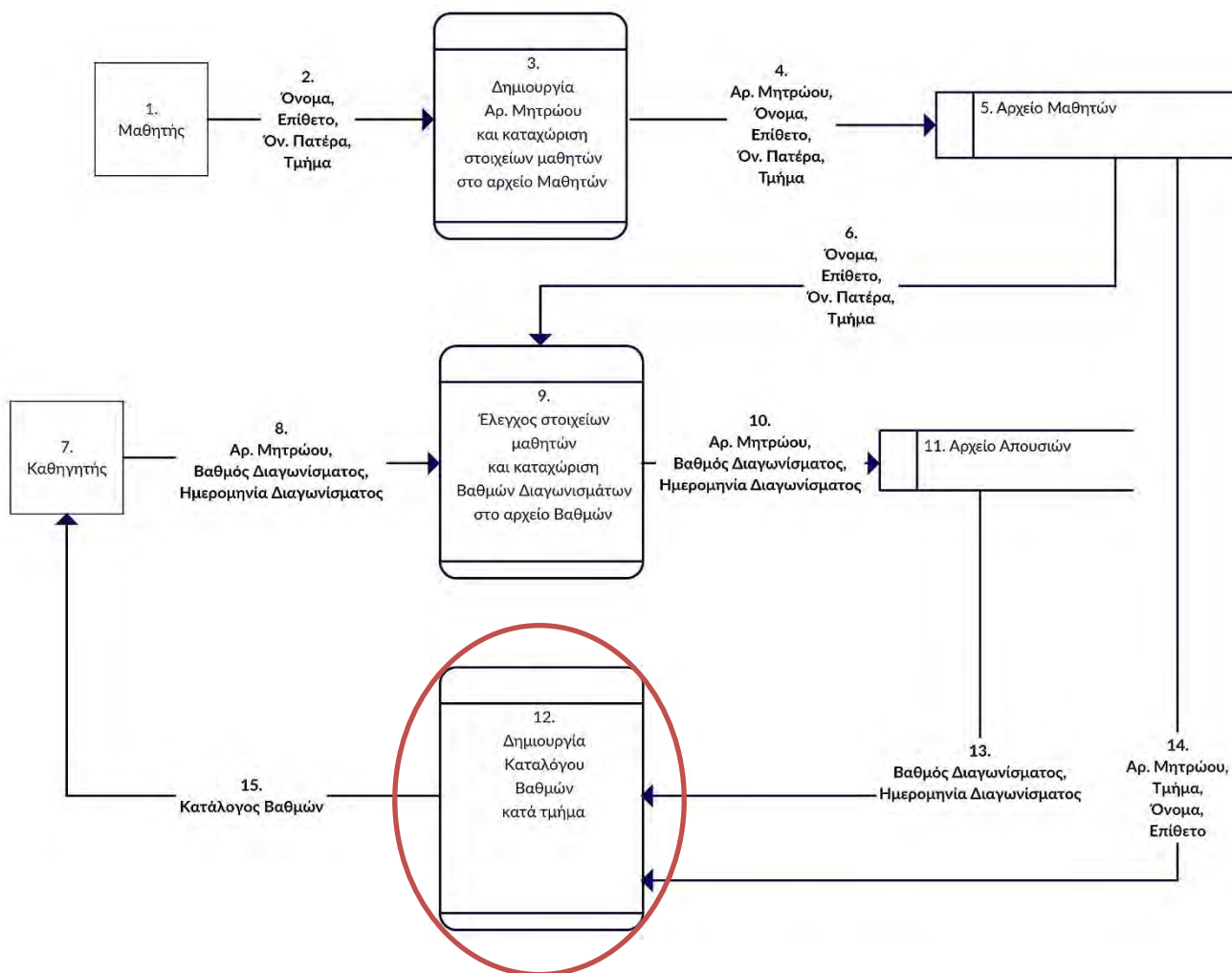
Όνομα:

Επίθετο:

Όνομα Πατέρα: **Τμήμα:**

Βαθμός: **Ημερ. Διαγωνίσματος:**

Παράδειγμα 3: Σχεδιασμός Αναφοράς (Καταλόγου)

ΔΡΑ**Αναφορά (Κατάλογος) Βαθμών μαθητών κατά Τμήμα**

Κατάλογος Βαθμών κατά Τμήμα			
Τμήμα:	Γ2		
Όνομα	Επίθετο	Βαθμός	Ημερ. Διαγωνίσμ
Ανδρέας	Ευγενίου	18,5	12/10/2016
Νικόλας	Νικολάου	19	12/10/2016
Τμήμα:	Γ5		
Όνομα	Επίθετο	Βαθμός	Ημερ. Διαγωνίσμ
Χρήστος	Ανδρέου	18	12/10/2016

**Ασκήσεις Κεφαλαίου****Άσκηση 5.1**

Να σχεδιάσετε την οθόνη, την αποθηκευτική μονάδα και την αναφορά (κατάλογο μαθητών), με βάση το Διάγραμμα Ροής Δεδομένων (ΔΡΔ) που σχεδιάσατε στην **άσκηση 3.3**.

Άσκηση 5.2

Να σχεδιάσετε την οθόνη, την αποθηκευτική μονάδα και την αναφορά (πιστοποιητικό γέννησης), με βάση το Διάγραμμα Ροής Δεδομένων (ΔΡΔ) που σχεδιάσατε στην **άσκηση 3.4**.

Άσκηση 5.3

Να σχεδιάσετε τις οθόνες, τις αποθηκευτικές μονάδες και τη σχέση μεταξύ τους, καθώς επίσης την αναφορά (κατάλογο ασθενών), με βάση το Διάγραμμα Ροής Δεδομένων (ΔΡΔ) που σχεδιάσατε στην **άσκηση 3.5**.

Άσκηση 5.4

Να σχεδιάσετε τις οθόνες, τις αποθηκευτικές μονάδες και τη σχέση μεταξύ τους, καθώς επίσης την κάρτα μέλους και το ατομικό πρόγραμμα, με βάση το Διάγραμμα Ροής Δεδομένων (ΔΡΔ) που σχεδιάσατε στην **άσκηση 3.6**.

B6.6 Υλοποίηση – Συνένωση Κώδικα και Έλεγχος συστήματος

Τι θα μάθουμε σε αυτό το κεφάλαιο:

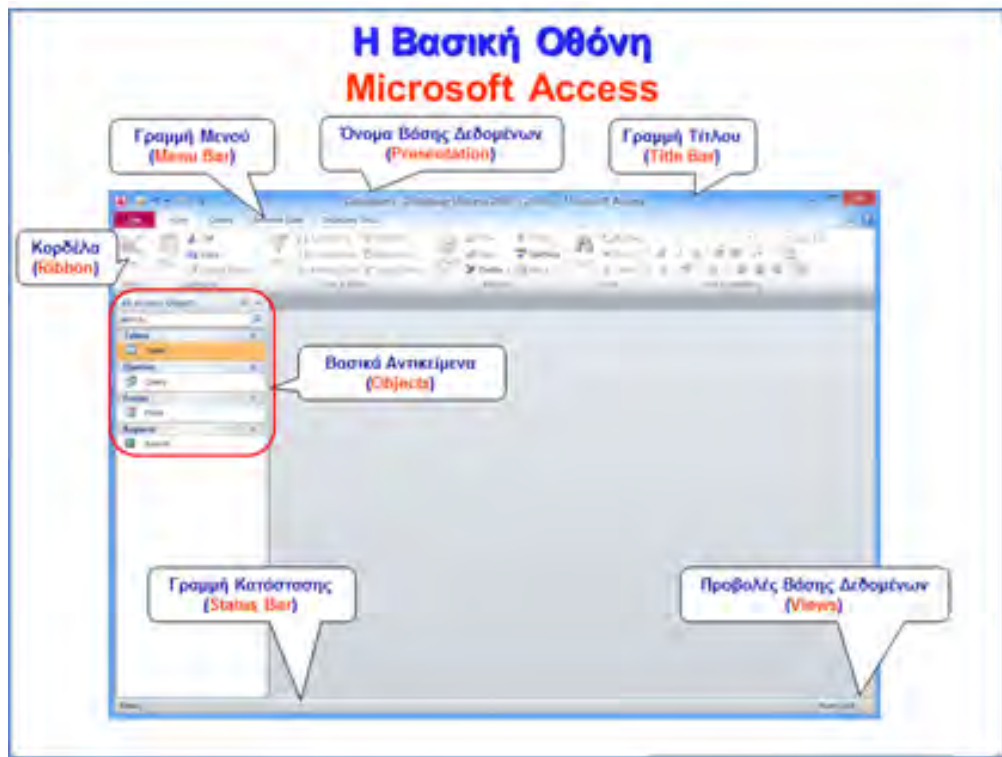
- ❖ Να χρησιμοποιούμε το περιβάλλον ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων και τα βασικά εργαλεία του
- ❖ Να δημιουργούμε μία νέα Βάση Δεδομένων (Blank database), σε μία μονάδα αποθήκευσης με συγκεκριμένο όνομα
- ❖ Να δημιουργούμε Πίνακα με τη χρήση της Προβολής Σχεδίασης (table design)
- ❖ Να καθορίζουμε τα πεδία του πίνακα και επιλέγουν τον κατάλληλο τύπο δεδομένων
- ❖ Να καθορίζουμε τις ιδιότητες των πεδίων (field properties)
- ❖ Να ορίζουμε το πρωτεύον κλειδί του πίνακα (Primary Key)
- ❖ Να εισαγάγουμε εγγραφές σε πίνακα μέσω του πληκτρολογίου ή μέσω αρχείου (import data)
- ❖ Να δημιουργούμε σχέσεις Ένα-Προς-Πολλά (One-To-Many) και Ένα-Προς-Ένα (One-To-One) μεταξύ δύο πινάκων σε μία Βάση Δεδομένων
- ❖ Να συνδέουμε τη βάση δεδομένων με το περιβάλλον ή τα περιβάλλοντα ανάπτυξης των διεπαφών (π.χ. φορμών και εκθέσεων) του συστήματος
- ❖ Να δημιουργούμε φόρμα με τη χρήση κατάλληλου, εύχρηστου εργαλείου (π.χ. του οδηγού φορμών-form wizard)
- ❖ Να δημιουργούμε ερωτήματα, επιλέγοντας πεδία από έναν ή περισσότερους σχετιζόμενους πίνακες και να εφαρμόζουμε κριτήρια και ταξινόμηση, σύμφωνα με τις απαιτήσεις της σχεδίασης του πληροφοριακού συστήματος
- ❖ Να δημιουργούμε έκθεση με τη χρήση κατάλληλου, εύχρηστου εργαλείου (π.χ. του οδηγού εκθέσεων-report wizard)
- ❖ Να εφαρμόζουμε ομαδοποίηση (grouping) και ταξινόμηση (sorting) σε μία έκθεση
- ❖ Να δημιουργούμε μενού επιλογών με τη χρήση κατάλληλου, εύχρηστου εργαλείου (π.χ. Switchboard Manager)
- ❖ Να ελέγχουμε κατά πόσο όλα τα μέρη του συστήματος λειτουργούν σύμφωνα με τις προδιαγραφές από τη σχεδίασή τους
- ❖ Να ελέγχουμε κατά πόσο ολόκληρο το πληροφοριακό σύστημα λειτουργεί σύμφωνα με τις προδιαγραφές του
- ❖ Να δημιουργούμε έναν απλό οδηγό/εγχειρίδιο χρήσης του συστήματος.

1. Υλοποίηση – Συνένωση Κώδικα και Έλεγχος συστήματος


Όπως έχει ήδη αναφερθεί, στη φάση της Υλοποίησης – Συνένωσης Κώδικα και Ελέγχου του Συστήματος γίνεται η ανάπτυξη των διαφόρων μερών του λογισμικού με κώδικα (κωδικοποίηση). Η ανάπτυξη του λογισμικού, το οποίο σχεδιάστηκε στην προηγούμενη φάση γίνεται χρησιμοποιώντας διάφορες γλώσσες προγραμματισμού και βάσεις δεδομένων, όπως τη Microsoft Access.

2. Εισαγωγή στη Microsoft Access

Το σχήμα στην επόμενη σελίδα παρουσιάζει το παράθυρο της Microsoft Access, από το οποίο και θα γίνει η ανάπτυξη της βάσης δεδομένων.

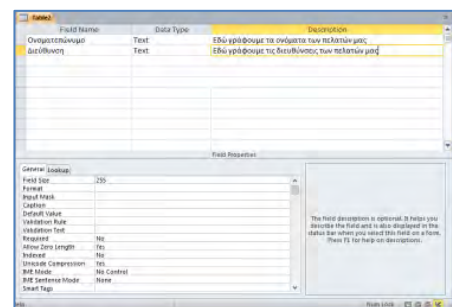
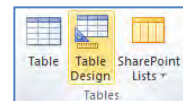


3. Δημιουργία και Αποθήκευση Βάσης Δεδομένων

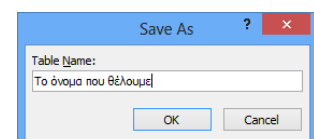
- Ενεργοποιούμε το πρόγραμμα Microsoft Access .
- Από το μενού File επιλέγουμε New, Blank Database και τέλος, πατούμε το κουμπί Create.

4. Δημιουργία και Αποθήκευση Πίνακα (Table)

- Από την καρτέλα **Create** και την ομάδα **Tables** επιλέγουμε το εικονίδιο **Table Design**.
- Στο παράθυρο που εμφανίζεται, πληκτρολογούμε το όνομα του πεδίου που θέλουμε να χρησιμοποιήσουμε στην **πρώτη γραμμή** του πλαισίου του **Field Name**.
- Στο πλαίσιο του **Data Type** επιλέγουμε από τη λίστα τον τύπο δεδομένων του πεδίου που θέλουμε να χρησιμοποιήσουμε.
- Στο πλαίσιο του **Description** πληκτρολογούμε την περιγραφή-σχόλια του πεδίου που θέλουμε να χρησιμοποιήσουμε.

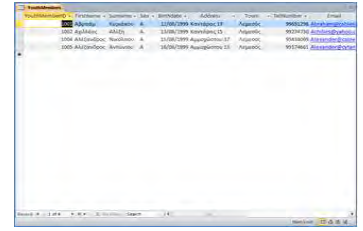


- Καθορίζουμε ένα από τα πεδία που έχουμε εισάγει σαν το **Πρωτεύων κλειδί** (Primary Key) του πίνακα.
- Στη συνέχεια, αποθηκεύουμε τον πίνακα, δίνοντας ένα όνομα που να ανταποκρίνεται στο περιεχόμενό του.





- Χρησιμοποιώντας την **προβολή φύλλου δεδομένων** (Datasheet View), αρχίζουμε την πληκτρολόγηση των εγγραφών που θέλουμε να καταχωρίσουμε στον πίνακά μας.



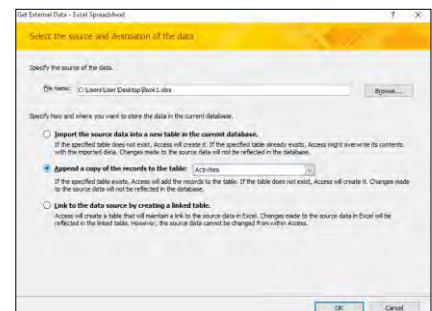
- Για τον **καθορισμό του τύπου δεδομένων** μπορούν να χρησιμοποιηθούν οι ακόλουθοι βασικοί τύποι:
 - **Short Text** Κείμενο μικρής έκτασης
 - **Number** Αριθμός
 - **Date/Time** Ημερομηνία/Ωρα
 - **Currency** Χρηματικό ποσό
 - **AutoNumber** Αυτόματη αρίθμηση
 - **Yes/No** Ναι/Όχι ή Σωστό/Λάθος ή Αληθές/Ψευδές
- Επιπρόσθετα, μπορούν να εισαχθούν ιδιότητες σε κάθε πεδίο, έτσι ώστε να καθοριστεί η εμφάνιση, η παρουσίαση και ο χειρισμός των δεδομένων ενός πεδίου. Οι βασικότερες **ιδιότητες των πεδίων** στη Microsoft Access είναι:

- **Field Size** Μέγεθος πεδίου
- **Format** Μορφή (τρόπος εμφάνισης)
- **Input Mask** Μάσκα εισαγωγής
- **Caption** Λεζάντα (Ταμπέλα)
- **Default Value** Προεπιλεγμένη (αρχική) τιμή
- **Validation Rule** Κανόνας επικύρωσης ή εγκυρότητας
- **Validation Text** Κείμενο επικύρωσης ή εγκυρότητας
- **Required** Απαιτείται (Εάν είναι υποχρεωτικό ή όχι)
- **Decimal Places** Δεκαδικές θέσεις ή Δεκαδικά ψηφία
- **Indexed** Έλεγχος επανεμφάνισης τιμής πεδίου
 - No – Κανένας Έλεγχος
 - Yes (Duplicates OK) – Επανάληψη αποδεκτή
 - Yes (No Duplicates) – Επανάληψη μη αποδεκτή

5. Εισαγωγή εγγραφών στον πίνακα από αρχείο

Υπάρχουν περιπτώσεις κατά τις οποίες ο χρήστης έχει ήδη αποθηκευμένα σε ένα αρχείο τα δεδομένα που επιθυμεί να εισαγάγει σε έναν πίνακα. Σε αυτή την περίπτωση δεν υπάρχει λόγος να πληκτρολογήσει ξανά τις εγγραφές, αφού η Access επιτρέπει την εισαγωγή δεδομένων από αρχείο, π.χ. βιβλίο εργασίας στο Microsoft Excel. Σε μία τέτοια περίπτωση ακολουθούμε τα πιο κάτω βήματα:

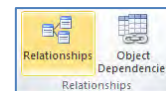
- Επιλέγουμε το μενού **External Data** και στη συνέχεια
- Εμφανίζεται η οθόνη (δεξιά), στην οποία επιλέγουμε που βρίσκεται αποθηκευμένο το αρχείο που περιέχει τα δεδομένα.
- Επιπρόσθετα, καθορίζουμε κατά πόσο τα δεδομένα αυτά θα εισαχθούν σε έναν ήδη δημιουργημένο πίνακα στη βάση δεδομένων ή σε έναν νέο πίνακα.
- Πατώντας το κουμπί **OK** εμφανίζονται τα δεδομένα και ολοκληρώνεται η διαδικασία.



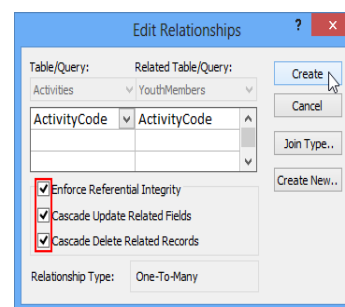
ΠΡΟΣΟΧΗ: στην περίπτωση που εισάγουμε στοιχεία σε έναν υφιστάμενο πίνακα, θα πρέπει να είμαστε ιδιαίτερα προσεκτικοί όσον αφορά την τιμή του πρωτεύοντος κλειδιού του πίνακα, διαφορετικά η διαδικασία δεν μπορεί να επιτευχθεί.

6. Δημιουργία Σχέσεων μεταξύ των Πινάκων (Relationships)

- Αρχικά, θα πρέπει να αποφασίσουμε ποιοι πίνακες θα συμπεριληφθούν στη σχέση που θέλουμε να χρησιμοποιήσουμε και βεβαιωνόμαστε ότι αυτοί οι πίνακες είναι **κλειστοί**.
- Από την καρτέλα Database Tools και την ομάδα Relationships επιλέγουμε το εικονίδιο Relationships, για να ανοίξουμε το παράθυρο των σχέσεων
- Στη συνέχεια, από το πλαίσιο διαλόγου εμφάνισης πινάκων επιλέγουμε τον/τους **πίνακα/ες** ή το/τα **ερώτημα/τα** που θέλουμε να χρησιμοποιήσουμε και κάνουμε κλικ στο κουμπί **Add**, για να το/τα **προσθέσουμε** στο παράθυρο των σχέσεων μας.
- Ακολούθως, κάνουμε κλικ στο κουμπί **Close**, για να **κλείσουμε** το πλαίσιο διαλόγου εμφάνισης πινάκων.
- Επιλέγουμε το πεδίο που είναι το πρωτεύον κλειδί από τον πρωτεύοντα πίνακα (Activities) της σχέσης, το **σύρουμε** πάνω στο συσχετιζόμενο πεδίο του δευτερεύοντος πίνακα και το αφήνουμε.
- Στο παράθυρο της επεξεργασίας των σχέσεων (Edit Relationships) που εμφανίζεται, μπορούμε να επιλέξουμε τα πιο κάτω:



- **Enforce Referential Integrity** (Ενεργοποίηση Ακεραιότητας Αναφορών) - Το επιλέγουμε για να βεβαιωθούμε ότι τα συνδεδεμένα πεδία δεν περιέχουν τιμές που δεν ταιριάζουν.

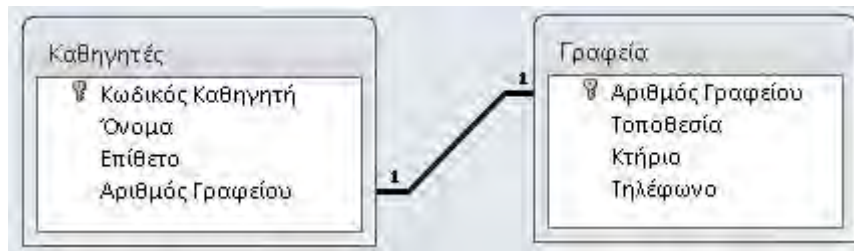


- **Cascade Update Related Fields** (Διαδοχική Ενημέρωση Συσχετιζόμενων Πεδίων) - Το επιλέγουμε για να ενημερωθούν όλες οι συσχετιζόμενες εγγραφές σε περίπτωση που αλλάξουμε την τιμή του πρωτεύοντος κλειδιού μίας εγγραφής.
- **Cascade Delete Related Records** (Διαδοχική Διαγραφή Συσχετιζόμενων Εγγραφών) - Το επιλέγουμε όταν στην περίπτωση που διαγράψουμε την εγγραφή στον πρώτο πίνακα, επιθυμούμε θα διαγράφονται και όλες οι συσχετιζόμενες εγγραφές στον άλλο πίνακα.

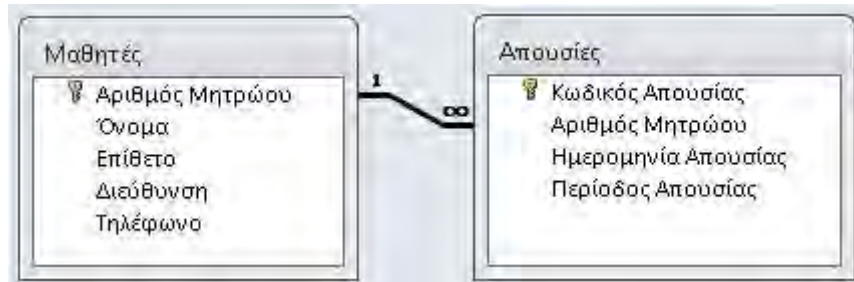
- Ακολούθως, κάνουμε κλικ στο κουμπί **Create** και η σχέση μας θα δημιουργηθεί.
- Τέλος, αποθηκεύουμε τις σχέσεις που έχουμε δημιουργήσει και κλείνουμε το παράθυρο των σχέσεων.

6.1 Δημιουργία Ειδικών Σχέσεων μεταξύ πινάκων

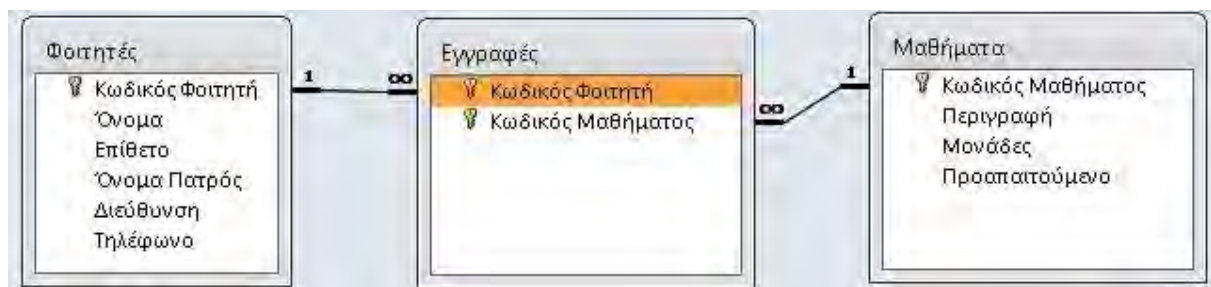
- Για δημιουργία σχέσης **Ένα προς Ένα** (One to One), πρέπει το πεδίο κλειδί του ενός πίνακα να συσχετιστεί είτε με το ίδιο πεδίο κλειδί σε άλλο πίνακα είτε με το δευτερεύον κλειδί, του οποίου η ιδιότητα **Indexed** να είναι **Yes (No Duplicates)**.



- Για δημιουργία σχέσης **Ένα προς Πολλά** (One to Many), πρέπει το πεδίο κλειδί του ενός πίνακα να συσχετιστεί με το δευτερεύον κλειδί του άλλου πίνακα.

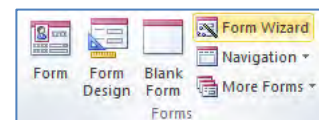


- Για δημιουργία σχέσης **Πολλά προς Πολλά** (Many to Many), πρέπει να δημιουργηθεί ένας επιπρόσθετος πίνακας, όπου θα συσχετιστούν τα δύο πεδία που αποτελούν το πεδίο κλειδί με τα πεδία κλειδιά των άλλων δύο πινάκων.



7. Δημιουργία Φόρμας Εισαγωγής Δεδομένων (Form)

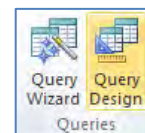
- Από την καρτέλα **Create** και την ομάδα **Forms** επιλέγουμε το εικονίδιο **Form Wizard**.
- Ακολουθώντας, επιλέγουμε τον πίνακα
- Κάνουμε κλικ στο κουμπί **>>**, για να μεταφέρουμε ΟΛΑ τα πεδία στο πλαίσιο **Selected Fields:** ή κάνουμε κλικ στο κουμπί **>**, για να μεταφέρουμε ΕΝΑ προς ΕΝΑ τα πεδία στο πλαίσιο **Selected Fields:** και μετά κάνουμε κλικ στο κουμπί **Next >**, για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.
- Στη συνέχεια, επιλέγουμε την **επιθυμητή διάταξη** για τη φόρμα μας και κάνουμε ξανά κλικ στο κουμπί **Next >**, για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.
- Τέλος, καθορίζουμε ένα **περιγραφικό όνομα** και κάνουμε κλικ στο κουμπί **Finish**, για να **αποθηκεύσουμε** τη φόρμα μας.



- Επιλέγουμε το εικονίδιο **Design View**, για να ανοίξουμε τη φόρμα μας σε προβολή σχεδίασης, έτσι ώστε να τη μορφοποιήσουμε.

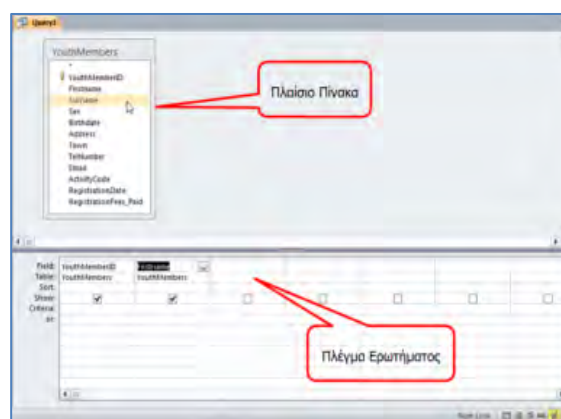
8. Δημιουργία Ερωτημάτων (Queries)

- Επιλέγουμε τον πίνακα
- Από την καρτέλα **Create** και την ομάδα **Queries** επιλέγουμε το εικονίδιο **Query Design**.



- Στη συνέχεια, από το πλαίσιο διαλόγου εμφάνισης πινάκων (**Show Table**) επιλέγουμε τον/τους **πίνακα/ες** που θέλουμε να χρησιμοποιήσουμε και κάνουμε κλικ στο κουμπί **Add**, για να τον/τους **προσθέσουμε**.
- Ακολούθως, κάνουμε κλικ στο κουμπί **Close**, για να **κλείσουμε** το πλαίσιο διαλόγου εμφάνισης πινάκων.

- Στο **πλαίσιο πίνακα**, κάνουμε **διπλό κλικ** στο πρώτο πεδίο, για να το **προσθέσουμε** στην πρώτη στήλη του πλέγματος ερωτημάτων μας.
- Συνεχίζουμε με τον ίδιο τρόπο όπως πιο πάνω, μέχρι να **προσθέσουμε** όλα τα πεδία που θέλουμε να χρησιμοποιήσουμε στο ερώτημά μας.



- Ακολούθως, από την καρτέλα **Query Tools→Design** και την ομάδα **Results** επιλέγουμε το εικονίδιο **Run**, για να εκτελέσει το ερώτημα η Microsoft Access και να εμφανίσει το αποτέλεσμα.



- Εκτελούμε το ερώτημα, για να δούμε τα αποτελέσματα επιλέγοντας **Run**.
- Τέλος, κλείνουμε το ερώτημα, του καθορίζουμε ένα **περιγραφικό όνομα** και αποθηκεύουμε το ερώτημά μας.

Επιπρόσθετα, κρίνεται αναγκαίο να γνωρίζουμε τα ακόλουθα για τη δημιουργία των ερωτημάτων:

Το πλέγμα ερωτήματος περιέχει:

Sort: Καθορίζουμε την ταξινόμηση που θα χρησιμοποιήσουμε όπως:

- **Ascending** (ΑΥΞΟΥΣΑ ταξινόμηση),
- **Descending** (ΦΘΙΝΟΥΣΑ ταξινόμηση),
- **(not sorted)** (ΔΙΧΩΣ ταξινόμηση).





Show: Καθορίζουμε εάν το πεδίο θα είναι ορατό ή όχι, όταν εκτελέσουμε το ερώτημα.

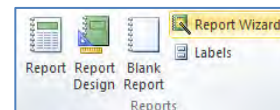
Criteria: Καθορίζουμε τα κριτήρια, δηλαδή τους περιορισμούς που ορίζουμε στα ερωτήματα, ώστε να μας βρίσκουν μόνο τις εγγραφές που μας ενδιαφέρουν.

Τα κριτήρια είναι οι **περιορισμοί** που ορίζουμε στα ερωτήματα, ώστε να βρίσκουν από τη Βάση Δεδομένων **μόνο τις εγγραφές** που μας ενδιαφέρουν, όπως:

Σταθερές Τιμές	Έκφραση	Περιγραφή
Λέξη ή Γράμμα	Λεμεσός	Ίση με ...
Αριθμός	2012	Ίση με ...
Ημερομηνία	31/1/1967	Ίση με ...
Συγκριτικοί Τελεστές	Έκφραση	Περιγραφή
Between ...and...	Between 1/1/2013 and 31/12/2013	Μεταξύ ... και ...
=	= 35	Ίση με ...
>	> 36	Μεγαλύτερη από ...
<	< 40	Μικρότερη από ...
<>	<> 0	Διάφορο του ... (άνισο)
>=	>= 50	Μεγαλύτερη ή ίση με ...
<=	<= 46	Μικρότερη ή ίση με ...
Λογικοί Τελεστές	Έκφραση	Περιγραφή
AND	< 50 AND > 40	και ο ένας περιορισμός και ο άλλος
OR	48 OR 52	ή ο ένας περιορισμός ή ο άλλος
NOT	NOT Ελένη	Εκτός από ...
Χαρακτήρες Μπαλαντέρ	Έκφραση	Περιγραφή
*	N*	Αντιστοιχεί σε κανέναν ή περισσότερους χαρακτήρες
?	N????	Αντιστοιχεί με οποιονδήποτε μεμονωμένο χαρακτήρα
#	2#5	Αντιστοιχεί με οποιονδήποτε μεμονωμένο αριθμητικό χαρακτήρα

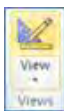
9. Δημιουργία Αναφορών (Reports)

- Επιλέγουμε το ερώτημα ή τον πίνακα για το/τον οποίο θέλουμε να δημιουργήσουμε μία έκθεση.
- Από την καρτέλα **Create** και την ομάδα **Reports** επιλέγουμε το εικονίδιο **Report Wizard**.
- Ακολουθώντας, κάνουμε κλικ στο κουμπί , για να μεταφέρουμε ΟΛΑ τα πεδία στο πλαίσιο **Selected Fields:** ή κάνουμε κλικ στο κουμπί , για να μεταφέρουμε ΕΝΑ προς ΕΝΑ τα πεδία στο πλαίσιο **Selected Fields:** και μετά κάνουμε κλικ στο κουμπί , για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.
- Στη συνέχεια, επιλέγουμε ένα από τα πεδία, βάσει του οποίου θέλουμε να **ομαδοποιήσουμε** την έκθεσή μας και κάνουμε κλικ στο κουμπί , για να το



μεταφέρουμε στο πλαίσιο στα δεξιά και κάνουμε ξανά κλικ στο κουμπί **Next >**, για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.

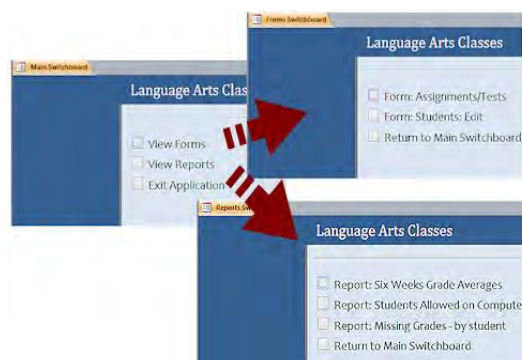
- Στη συνέχεια, μπορούμε επιλέγοντας μέχρι και τέσσερα πεδία, να **ταξινομήσουμε** την έκθεσή μας ανάλογα, σε αύξουσα ή φθίνουσα αλφαβητική/αριθμητική σειρά βάσει του πεδίου και κάνουμε ξανά κλικ στο κουμπί **Next >**, για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.
- Στη συνέχεια, επιλέγουμε την **επιθυμητή διάταξη** μαζί με τον **επιθυμητό προσανατολισμό** της έκθεσής μας και κάνουμε ξανά κλικ στο κουμπί **Next >**, για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.
- Τέλος, καθορίζουμε ένα **περιγραφικό όνομα** και κάνουμε κλικ στο κουμπί **Finish**, για να **αποθηκεύσουμε** την έκθεσή μας.



- Επιλέγουμε το εικονίδιο Design View, για να ανοίξουμε την αναφορά μας σε προβολή σχεδίασης, έτσι ώστε να τη μορφοποιήσουμε.

10. Δημιουργία Switchboard

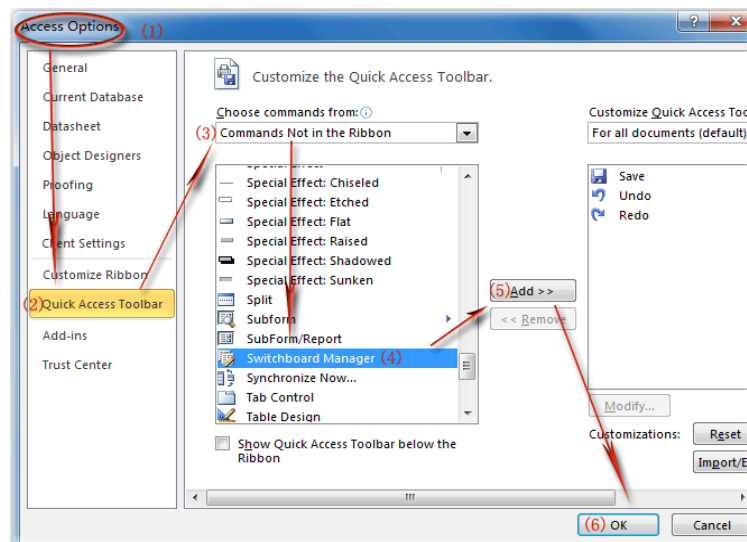
Το **Switchboard** είναι βασικά ένα μενού, το οποίο δίνει στον χρήστη τη δυνατότητα μέσω ενός φιλικού προς τον ίδιο περιβάλλοντος, να ενεργοποιήσει τα βασικά αντικείμενα που έχει δημιουργήσει στη βάση δεδομένων του. Επιπρόσθετα, είναι και ένας τρόπος προστασίας των αντικειμένων που έχει δημιουργήσει.



11. Εμφάνιση του Switchboard Manager


Στην περίπτωση της έκδοσης **Access 2010/2013/2016** δεν υπάρχει ενεργοποιημένη η επιλογή του Switchboard Manager. Σε αυτή την περίπτωση ακολουθούμε τα πιο κάτω βήματα για ενεργοποίηση του Switchboard Manager:

1. Από το μενού **File** επιλέγουμε **Options** και τότε εμφανίζεται το ακόλουθο παράθυρο.
2. Επιλέγουμε το **Quick Access Toolbar**, που βρίσκεται στο αριστερό μέρος της οθόνης μας.



3. Από τη λίστα «Choose Commands» επιλέγουμε **Commands Not in the Ribbon**.
4. Επιλέγουμε το Switchboard Manager και πατούμε το κουμπι **Add**.
5. Κλείνουμε το παράθυρο, πατώντας το κουμπι OK.

12. Δημιουργία Switchboard

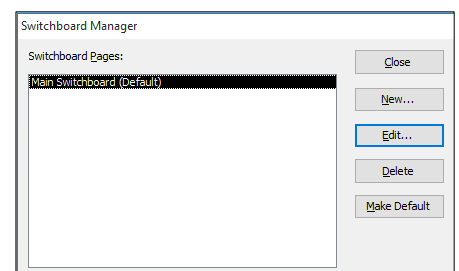
Από την κυρίως κορδέλα (ribbon) επιλέγουμε . Εμφανίζεται η διπλανή οθόνη, η οποία, μέσω των κουμπιών, μας δίνει τις ακόλουθες επιλογές:

New: Δημιουργία νέου Switchboard Page, για το οποίο μας ζητείται να δώσουμε όνομα

Edit: Τροποποίηση ενός υφιστάμενου Switchboard Page

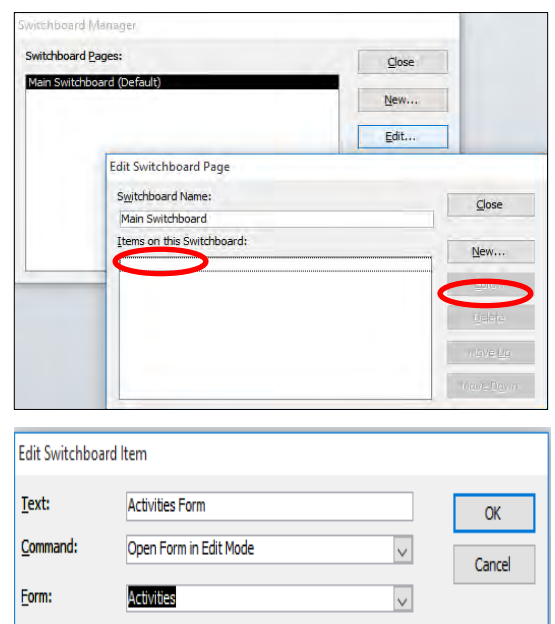
Delete: Διαγραφή Switchboard Page

Make Default: Το Switchboard Page εμφανίζεται πρώτο, όταν ανοίξουμε το Switchboard Form



Για δημιουργία **ενός απλού Switchboard** επιλέγουμε το κουμπι **Edit**. Ακολούθως, εμφανίζεται το διπλανό παράθυρο στο οποίο αρχικά δίνουμε ένα **όνομα** που να περιγράφει το μενού. Πατώντας το κουμπι **New**, μας δίνεται η δυνατότητα να καθορίσουμε τις επιλογές που θα εμφανίζονται πάνω στο Switchboard.

Στο παράθυρο που εμφανίζεται καλούμαστε να δώσουμε **όνομα για την επιλογή**, το είδος της εντολής που θα εκτελείται και σε ποιο αντικείμενο θα εφαρμοστεί η εντολή που έχουμε δηλώσει. Π.χ. αν θέλουμε να εμφανίζεται η επιλογή ενεργοποίησης της φόρμας Activities και να επιτρέπεται η τροποποίηση



δεδομένων μέσω της φόρμας, τότε καθορίζουμε ως **Command** την **επιλογή Open Form in Edit Mode** και στο Form επιλέγουμε τη φόρμα Activities. Υπάρχουν και άλλες ενέργειες που μπορούν να γίνουν μέσω ενός Switchboard. Πιο κάτω δίνονται οι πιο βασικές ενέργειες:

- | | |
|--------------------------|---|
| • Go to Switchboard | Μετάβαση στην αρχική οθόνη |
| • Open Form in Add Mode | Προσθήκη Εγγραφών μέσω μίας φόρμας |
| • Open Form in Edit Mode | Τροποποίηση των Εγγραφών μέσω μίας φόρμας |
| • Open Report | Παρουσίαση αναφοράς |
| • Exit Application | Έξοδος από τη βάση δεδομένων |

Είναι σημαντικό να αναφερθεί ότι μας δίνεται η δυνατότητα να έχουμε υπό-μενού (sub-menus) σε ένα Switchboard, μέσω των Switchboard pages.

Επίσης, κατά τη δημιουργία του Switchboard, δημιουργείται αυτόματα από την Access ένας πίνακας, ο οποίος περιέχει εγγραφές σχετικές με κάθε επιλογή του. Δεν πρέπει να τροποποιούμε ή να διαγράψουμε τις εγγραφές αυτές.

Item	Command	Argument
0 Main Switchboard	Default	
1 View Reports	2.1	
2 View Reports	3.2	
3 View Reports	0	
4 Reports on Weeks Grade Averages	4	Six Weeks Grade Averages
5 Report: Students Allowed on Computer	4	Allowed on Computer
6 Report: Missing Grades - by student	4	Missing Grades-by student
7 Form Switchboard	3.1	
8 Form Switchboard	0	
9 Form Switchboard	2	Assignments/Tests Form
10 Form Switchboard	3	Students Form
11 New Switchboard Page	3.1	

13. Πρόσβαση στη Βάση Δεδομένων από πολλούς χρήστες

Μία βάση δεδομένων της Access, όπως έχουμε δει, αποτελείται από αντικείμενα, όπως πίνακες, ερωτήματα, φόρμες και εκθέσεις. Στους πίνακες αποθηκεύονται δεδομένα. Κάθε άλλο είδος αντικειμένου βάσης δεδομένων μάς βοηθά να χρησιμοποιήσουμε τα δεδομένα που είναι αποθηκευμένα στους πίνακες. Όταν αναφερόμαστε στην κοινή χρήση μίας βάσης δεδομένων, συνήθως, εννοούμε την κοινή χρήση των πινάκων, διότι αυτοί περιέχουν τα δεδομένα. Υπάρχουν αρκετές μέθοδοι κοινής χρήσης δεδομένων. Οι πιο συχνά χρησιμοποιούμενες είναι:

- **Διαίρεση Βάσης Δεδομένων (Split Database)**
- **Δημιουργία Φάκελου Δικτύου (Network folder)**
- **Δημιουργία τοποθεσίας SharePoint (SharePoint Site)**
- **Χρήση Διακομιστή Βάσης Δεδομένων (Database Server)**

13.1 Διαίρεση Βάσης Δεδομένων (Split Database)

Θα χρησιμοποιήσουμε τη μέθοδο της **Διαίρεσης της Βάσης Δεδομένων** (Split Database). Όταν διαιρούμε μία βάση δεδομένων, την αναδιοργανώνουμε σε δύο αρχεία — **μία παρασκηνιακή βάση δεδομένων (back-end database)**, η οποία περιέχει τους πίνακες δεδομένων και μία **βάση δεδομένων προσκηνίου (front-end database)**, η οποία περιέχει όλα τα άλλα αντικείμενα της βάσης δεδομένων, όπως ερωτήματα, φόρμες και εκθέσεις. Κάθε χρήστης αλληλεπιδρά με τα δεδομένα, χρησιμοποιώντας ένα τοπικό αντίγραφο της βάσης δεδομένων προσκηνίου.

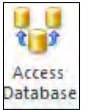
Προτού εκτελέσουμε τη διαδικασία διαίρεσης της βάσης δεδομένων, θα πρέπει να βεβαιωθούμε για τα ακόλουθα θέματα:

- Να έχουμε δημιουργήσει ένα εφεδρικό αρχείο αποθήκευσης της βάσης δεδομένων (backup)

- Όλοι οι χρήστες θα πρέπει να έχουν εγκαταστημένη στον υπολογιστή τους μία έκδοση της Access, που να είναι συμβατή με την έκδοση της παρασκευαστικής βάσης δεδομένων.

Στη συνέχεια ακολουθούμε τα πιο κάτω βήματα:

1. Αντιγράφουμε τη βάση δεδομένων μας, για να έχουμε ένα αντίγραφο της.
2. Ανοίγουμε τη βάση δεδομένων και από το Database Tools επιλέγουμε Access Database
3. Στην οθόνη που εμφανίζεται επιλέγουμε Split Database.
4. Στο **Create Back-end Database** παράθυρο καθορίζουμε το όνομα, τον τύπο του αρχείου και την τοποθεσία αποθήκευσης του back-end database file (Λάβετε υπόψη ότι όλοι οι χρήστες θα πρέπει να έχουν πρόσβαση στην τοποθεσία αποθήκευσης και ότι το όνομα το οποίο δίνεται από την Access έχει **-be**, που σημαίνει Back end).
5. Πατώντας στο κουμπί **Split**, δίνεται ένδειξη ότι η βάση δεδομένων έχει διαιρεθεί και εμφανίζεται σήμανση μπροστά από τον κάθε πίνακα.
6. Το **back-end database** θα χρησιμοποιείται από τον κάθε χρήστη. Για να συνδέσουμε τους πίνακες του front-end database με το back-end database, επιλέγουμε το **External Data** και στη συνέχεια **Linked Table Manager**.
7. Στην οθόνη που εμφανίζεται επιλέγουμε τους πίνακες και τέλος, πατούμε το κουμπί OK.



13.2 Δημιουργία διασύνδεσης με συνδεδεμένο πίνακα (Linked to data source by creating a linked table)

Εναλλακτικά με τη διαίρεση της Βάσης Δεδομένων, μπορούμε να χρησιμοποιήσουμε τη μέθοδο της **Δημιουργίας διασύνδεσης με συνδεδεμένο πίνακα** (Linked to data source by creating a linked table).

Η διαδικασία επιτυγχάνεται ως εξής:

1. Δημιουργούμε αρχικά μία άδεια βάση δεδομένων.
2. Επιλέγουμε από το **External Data** το εικονίδιο **Access**.
3. Καθορίζουμε τη βάση δεδομένων από την οποία θα πάρουμε τα δεδομένα και επιλέγουμε το **Linked to data source by creating a linked table**.



4. Στην οθόνη που εμφανίζεται επιλέγουμε τους πίνακες και τέλος, πατούμε το κουμπί OK.



Ασκήσεις Κεφαλαίου

Άσκηση 6.1

Χρησιμοποιώντας μία βάση δεδομένων, να αναπτύξετε το σύστημα που σχεδιάσατε στην **άσκηση 5.1**.

Άσκηση 6.2

Χρησιμοποιώντας μία βάση δεδομένων, να αναπτύξετε το σύστημα που σχεδιάσατε στην **άσκηση 5.2**.

Άσκηση 6.3

Χρησιμοποιώντας μία βάση δεδομένων, να αναπτύξετε το σύστημα που σχεδιάσατε στην **άσκηση 5.3**.

Άσκηση 6.4

Χρησιμοποιώντας μία βάση δεδομένων, να αναπτύξετε το σύστημα που σχεδιάσατε στην **άσκηση 5.4**.

Βιβλιογραφία

1. Κάκας, Α., Ανδρέου, Α., Ρετάλης, Σ. (2003). *Εισαγωγή στην Ανάλυση και Σχεδίαση Πληροφοριακών Συστημάτων*. Λευκωσία: Πανεπιστήμιο Κύπρου.
2. Βεσκούκης, Β. (2000). *Τεχνολογία Λογισμικού Ι*. Πάτρα: Ελληνικό Ανοικτό Πανεπιστήμιο.

ΕΝΟΤΗΤΑ Β7 Αλγοριθμική Σκέψη, Προγραμματισμός και Σύγχρονες Εφαρμογές Πληροφορικής

B7.1 Προγραμματισμός στη C++

Οι ασκήσεις της ενότητας που σημειώνονται με το **λογότυπο του HackerRank**, έχουν αναρτηθεί στον πιο κάτω σύνδεσμο:



<https://www.hackerrank.com/b-lykeiou>

Οι περισσότερες από τις υπόλοιπες ασκήσεις της ενότητας βρίσκονται στους συνδέσμους που εμφανίζονται στην επικεφαλίδα των ασκήσεων κάθε κεφαλαίου, οι οποίοι παρουσιάζονται συνοπτικά πιο κάτω:

B7.1.2 Τα πρώτα προγράμματα σε C++	https://www.hackerrank.com/7-1-2
B7.1.3 Δομές Διακλάδωσης	https://www.hackerrank.com/7-1-3
B7.1.4 Δομές Επανάληψης	https://www.hackerrank.com/7-1-4
B7.1.5 Πίνακες	https://www.hackerrank.com/7-1-5

B7.1.1 Ξεκινώντας...

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ❖ Να εγκαθιστούμε ένα κατάλληλο προγραμματιστικό περιβάλλον
- ❖ Να ενεργοποιούμε το προγραμματιστικό περιβάλλον
- ❖ Να δημιουργούμε ένα νέο αρχείο πηγαίου κώδικα
- ❖ Να ανοίγουμε ένα υπάρχον αρχείο πηγαίου κώδικα
- ❖ Να αποθηκεύουμε ένα αρχείο πηγαίου κώδικα με κατάλληλη επέκταση
- ❖ Να ενεργοποιούμε τη διαδικασία μεταγλώττισης του προγράμματος πηγαίου κώδικα και να εντοπίζουμε και να διορθώνουμε συντακτικά λάθη (αποσφαλμάτωση)
- ❖ Να ενεργοποιούμε τη διαδικασία εκτέλεσης ενός προγράμματος μέσα από το προγραμματιστικό περιβάλλον
- ❖ Να εντοπίζουμε το πρόγραμμα αντικείμενου κώδικα που προέκυψε από τη μεταγλώττιση και να το εκτελούμε έξω από το προγραμματιστικό περιβάλλον.

1. Εισαγωγή

Σε αυτό το εισαγωγικό κεφάλαιο παρουσιάζουμε τον τρόπο χρήσης των σημειώσεων, μία ιστορική αναδρομή στις γλώσσες προγραμματισμού και ειδικότερα στη C++ και το περιβάλλον του Code::Blocks (του IDE που θα χρησιμοποιήσουμε στο βιβλίο). Τέλος, γίνεται αναφορά σε άλλες επιλογές IDE (Integrated Development Environment), που θα μπορούσαν να χρησιμοποιηθούν.

Σκοπός των σημειώσεων είναι να βοηθήσουν τους μαθητές και τους καθηγητές της Β' και Γ' Λυκείου να κατακτήσουν τους Δείκτες Επιτυχίας του μαθήματος Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών. Στο πλαίσιο της αναδόμησης και του εκσυγχρονισμού του μαθήματος, αποφασίστηκε από την ομάδα των Αναλυτικών Προγραμμάτων η εισαγωγή της γλώσσας προγραμματισμού C++ στη διδασκαλία της αλγοριθμικής σκέψης.

Οι σημειώσεις ΔΕΝ αποτελούν εγχειρίδιο της C++. Δεν παρουσιάζουν τη χρήση δεικτών (pointers) ή κλάσεων (classes). Περιληπτικά, οι σημειώσεις ασχολούνται με τα ακόλουθα:

- Βασικές δομές προγράμματος (ακολουθιακή, διακλάδωση, επανάληψη)
- Δημιουργία συναρτήσεων από τον χρήστη
- Πίνακες (μονοδιάστατοι – δισδιάστατοι)
- Συμβολοσειρές (strings)
- Αρχεία (Files)
- Αφηρημένες δομές δεδομένων (structs)
- Στοίβες και ουρές (stacks – queues)

Η χρήση των strings, stacks και queues γίνεται μέσα από τις αντίστοιχες βιβλιοθήκες του STL (Standard Template Library).

Οι σημειώσεις είναι χωρισμένες σε πέντε κεφάλαια. Σε κάθε κεφάλαιο υπάρχουν αρκετά παραδείγματα ώστε να είναι ευκολότερη η επίτευξη των Δεικτών Επιτυχίας. Για έναν αριθμό αυτών των παραδειγμάτων έχουν ετοιμαστεί επεξηγηματικά βίντεο. Επίσης, ένας αριθμός παραδειγμάτων/ασκήσεων μπορεί να δοκιμαστεί διαδικτυακά (online), χρησιμοποιώντας την πλατφόρμα του HackerRank. Μέσω αυτής της πλατφόρμας, ο μαθητής μπορεί να υποβάλει την λύση του, η οποία θα εξεταστεί για αριθμό δοκιμαστικών περιπτώσεων (test cases).

Στην προσπάθειά μας να δείξουμε ότι τα βήματα που πρέπει να ακολουθήσει κάποιος για να αναπτύξει ένα πρόγραμμα είναι ανεξάρτητα από τη γλώσσα προγραμματισμού, παραθέτουμε τις λύσεις ορισμένων παραδειγμάτων σε Pascal, Python και Java. Μέσα από αυτά ο μαθητής

μπορεί να έχει μία πρώτη επαφή και με γνωστές γλώσσες προγραμματισμού, που δεν θα διδαχτεί στο σχολείο, για σκοπούς σύγκρισης.

Επεξήγηση εικονιδίων που θα συναντήσετε μέσα στις σημειώσεις:



Κώδικας γραμμένος σε Pascal



Κώδικας γραμμένος σε Python



Κώδικας γραμμένος σε Java



Επεξήγηση του παραδείγματος σε video



Η άσκηση είναι διαθέσιμη στο HackerRank

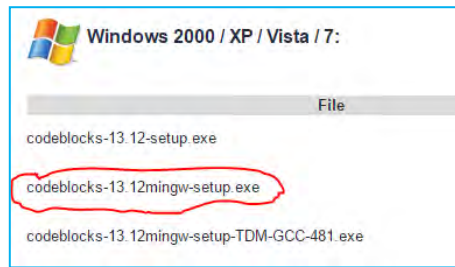
Στη σελίδα του Υπουργείου Παιδείας (Εκπαιδευτικό Υλικό) μπορείτε να βρείτε το υποστηρικτικό υλικό για τις σημειώσεις.

2. Λίγα λόγια για τη C++

- Δημιουργήθηκε το 1979 από τον Bjarne Stroustrup
- Στην αρχή ονομαζόταν C with classes
- Από τις πιο διαδεδομένες και ισχυρές γλώσσες προγραμματισμού
- Περιλαμβάνει όλες τις εντολές της C
- Σημαντικές εφαρμογές που γράφτηκαν σε C++
 - Microsoft Windows
 - Microsoft Office
 - Photoshop
 - Mozilla Firefox
 - MySQL
 - μTorrent

3. Χρησιμοποιώντας το Code::Blocks

Το Code::Blocks είναι ένα δωρεάν IDE (Integrated Development Environment) σχεδιασμένο για τη C/C++ (και όχι μόνο). Περιέχει έναν πολύ καλό editor που σας βοηθά να γράψετε τον κώδικα σας δομημένα, ενώ ως μεταγλωττιστή (compiler) χρησιμοποιεί τον g++. Μπορείτε να το κατεβάσετε δωρεάν από τη σελίδα <http://www.codeblocks.org/downloads/26>. Προσέξτε να κατεβάσετε την έκδοση που περιλαμβάνει και τον compiler (εικόνα 1), διαφορετικά θα πρέπει να τον εγκαταστήσετε ξεχωριστά.




Εικόνα 1

Πέραν από το Code::Blocks υπάρχουν αρκετά IDE που μπορείτε να χρησιμοποιήσετε, όπως το Dev C++ και το CodeLite. Επίσης, μπορείτε να δοκιμάσετε τα προγράμματα σας διαδικτυακά σε ιστοσελίδες, όπως <https://ideone.com> ή <https://www.codechef.com/ide>.

4. Δημιουργία νέου προγράμματος:


- Να ανοίξετε το codeblocks.
- Να επιλέξετε **File → New → Empty File**.
- Να αποθηκεύσετε το πρόγραμμά σας (Save As) με κατάληξη **.cpp**

5. Μεταγλώττιση και Εκτέλεση προγράμματος:

- Να μεταγλωττίσετε (compile) το πρόγραμμά σας πατώντας το κουμπί . Αν το πρόγραμμά σας είναι συντακτικά σωστό, τότε θα εμφανιστεί το μήνυμα που φαίνεται στην εικόνα 2.

```
Process terminated with status 0 (0 minute(s), 1 second(s))
0 error(s), 0 warning(s) (0 minute(s), 1 second(s))
```

Εικόνα 2

- Αν το πρόγραμμά σας δεν περιέχει συντακτικά λάθη, τότε να πατήσετε το κουμπί  για να το εκτελέσετε (run).

6. Επαναφορά προγράμματος:

- Να επιλέξετε File → Open.
- Να βρείτε το αρχείο που θέλετε να ανοίξετε και να πατήσετε το κουμπί Open.

7. Παραδείγματα

Παράδειγμα 1.1



- (1) Να ανοίξετε το codeblocks.
- (2) Να δημιουργήσετε ένα νέο αρχείο.
- (3) Να αποθηκεύσετε το αρχείο με όνομα Hello.cpp
- (4) Να αντιγράψετε τις πιο κάτω εντολές:

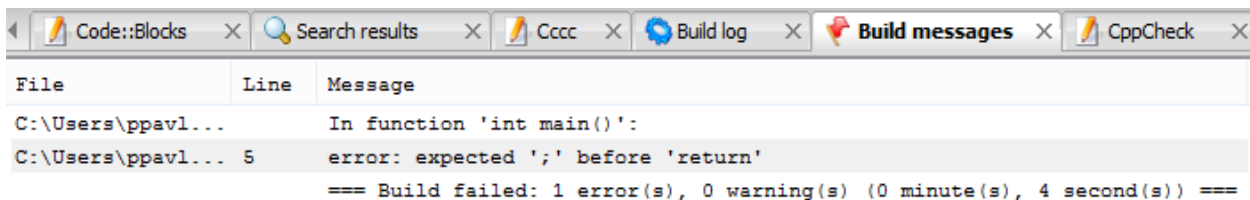
```
#include<iostream>
using namespace std;
int main() {
    cout<<"Hello World";
return 0;
}
```

- (5) Να μεταγλωττίσετε το πρόγραμμά σας
- (6) Αν δεν προέκυψαν λάθη από το βήμα 5, τότε να εκτελέσετε το πρόγραμμα.

Παράδειγμα 1.2



- (1) Να ανοίξετε το codeblocks.
- (2) Να ανοίξετε το Hello.cpp που έχετε δημιουργήσει στο παράδειγμα 1.1.
- (3) Να αποθηκεύσετε το αρχείο με νέο όνομα Hello1.cpp (Save as).
- (4) Να αφαιρέσετε το ερωτηματικό στην εντολή `cout<<"Hello World";`
- (5) Να μεταγλωττίσετε το πρόγραμμά σας.
- (6) Να παρατηρήσετε ότι στο παράθυρο Build Messages εμφανίζεται το μήνυμα της εικόνας 3
- (7) Να διορθώσετε την εντολή `cout<<"Hello World";` και να την αλλάξετε σε `cout<<"Hello You";`
- (8) Να μεταγλωττίσετε το πρόγραμμά σας.
- (9) Να εκτελέσετε το πρόγραμμά σας.



Εικόνα 3

B7.1.2 Τα Πρώτα Προγράμματα σε C++

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ❖ Να δημιουργούμε τη βασική δομή ενός προγράμματος (π.χ. `#include <>`, `namespace`, συνάρτηση `main() { ... }`)
- ❖ Να αναγνωρίζουμε την ανάγκη χρήσης μεταβλητών σε ένα πρόγραμμα
- ❖ Να δηλώνουμε μεταβλητές και σταθερές
- ❖ Να εντοπίζουμε ποιες μεταβλητές (Variables) χρειάζονται, με βάση τις ανάγκες του αλγορίθμου/προγράμματος
- ❖ Να αποφασίζουμε εάν κάποια τιμή στο πρόγραμμα χρειάζεται να οριστεί ως σταθερά
- ❖ Να δίνουμε κατάλληλα ονόματα στις μεταβλητές και στις σταθερές
- ❖ Να επιλέγουμε κατάλληλο τύπο δεδομένων (ακέραιος/πραγματικός αριθμός, Boolean, χαρακτήρας/κείμενο) για μία μεταβλητή ή σταθερά
- ❖ Να δηλώνουμε μεταβλητές (Variables) και σταθερές (Constants), με βάση τις ανάγκες του αλγορίθμου/προγράμματος, χρησιμοποιώντας κατάλληλα ονόματα
- ❖ Να χρησιμοποιούμε μεταβλητές (Variables) και σταθερές (Constants), με βάση τις ανάγκες του αλγορίθμου/προγράμματος
- ❖ Να χρησιμοποιούμε την εντολή εισόδου (π.χ. `cin`)
- ❖ Να χρησιμοποιούμε την εντολή εξόδου (π.χ. `cout`)
- ❖ Να εφαρμόζουμε εντολές/παραμέτρους για τη μορφοποίηση της εξόδου (αλλαγή γραμμής, καθορισμός ακρίβειας δεκαδικών ψηφίων πραγματικών αριθμών, καθορισμός διαστημάτων)
- ❖ Να χρησιμοποιούμε την εντολή εκχώρησης τιμής (Assignment)
- ❖ Να εντοπίζουμε και να διορθώνουμε συντακτικά σφάλματα που σχετίζονται με τη δήλωση και τη χρήση μεταβλητών και σταθερών (π.χ. απουσία δήλωσης, ακατάλληλα αναγνωριστικά, σφάλματα στη χρήση της εντολής εκχώρησης τιμής και στις εντολές εισόδου και εξόδου)
- ❖ Να χρησιμοποιούμε βασικούς αριθμητικούς τελεστές (π.χ. `+`, `-`, `*`, `/`, `%`) και παρενθέσεις (`()`)
- ❖ Να ακολουθούμε τους κανόνες προτεραιότητας των αριθμητικών τελεστών κατά την προκαταρκτική εκτέλεση ενός προγράμματος με εκφράσεις/παραστάσεις
- ❖ Να εφαρμόζουμε τους κανόνες προτεραιότητας, όταν δημιουργούμε μία έκφραση/παρασταση σε δικό μας πρόγραμμα
- ❖ Να αναγνωρίζουμε τους περιορισμούς των τύπων δεδομένων και τη διαδικασία μετατροπής τους από έναν τύπο σε άλλο, όταν χρησιμοποιούνται ταυτόχρονα σε μία έκφραση
- ❖ Να διατυπώνουμε εκφράσεις/παραστάσεις, οι οποίες περιλαμβάνουν δεδομένα με διαφορετικούς τύπους και να προβλέπουμε το αποτέλεσμα
- ❖ Να προετοιμάζουμε ένα πρόγραμμα για χρήση συγκεκριμένης βιβλιοθήκης έτοιμων συναρτήσεων.
- ❖ Να χρησιμοποιούμε έτοιμες συναρτήσεις από βιβλιοθήκες (εισόδου/εξόδου, μαθηματικών, μετατροπών, κ.λπ.) σε ένα πρόγραμμα
- ❖ Να εντοπίζουμε και να διορθώνουμε σφάλματα που σχετίζονται με τους τελεστές, εκφράσεις, παραστάσεις και έτοιμες συναρτήσεις (π.χ. ισοζυγισμού παρενθέσεων, χρήση τελεστή σε ακατάλληλη τιμή, λογικά σφάλματα σε σχέση με την προτεραιότητα, σφάλματα δεξιού/αριστερού μέρους εκχώρησης τιμής, σφάλματα δήλωσης βιβλιοθήκης συναρτήσεων)
- ❖ Να καθορίζουμε το πρόβλημα με ακρίβεια, συγκεκριμένα: να διακρίνουμε τα Δεδομένα, τις Πληροφορίες και την Επεξεργασία
- ❖ Να δημιουργούμε το Λογικό Διάγραμμα για την επίλυση του προβλήματος
- ❖ Να μεταφράζουμε ένα λογικό διάγραμμα σε πρόγραμμα με τη χρήση του προγραμματιστικού περιβάλλοντος, ώστε να επιλυθεί το πρόβλημα
- ❖ Να επιλέγουμε κατάλληλα δεδομένα και στρατηγική για έλεγχο του προγράμματος
- ❖ Να ελέγχουμε την ορθότητα της λύσης του προβλήματος, χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης και της χρήσης του debugger για επαλήθευση
- ❖ Να μελετούμε έτοιμο πρόγραμμα και να εντοπίζουμε βασικά μέρη του, τα οποία συνδέονται με πτυχές του προβλήματος που επιλύει
- ❖ Να προσθέτουμε επεξηγηματικά σχόλια σε ένα πρόγραμμα
- ❖ Να συμπληρώνουμε ένα έτοιμο πρόγραμμα, ώστε να αποτελεί λύση ενός διαφοροποιημένου προβλήματος
- ❖ Να εντοπίζουμε και να αναγνωρίζουμε σε ένα πρόγραμμα πρότυπα σχεδίασης και στρατηγικές (design patterns, τμήματα κώδικα)
- ❖ Να χρησιμοποιούμε πρότυπα σχεδίασης και στρατηγικές που εντοπίσαμε, ως εργαλεία επίλυσης προβλημάτων σε νέα προγράμμάτα μας.

Στο κεφάλαιο αυτό παρουσιάζονται τα πρώτα προγράμματα σε C++. Πρόκειται για προγράμματα ακολουθιακής δομής. Μέσα από αυτά τα προγράμματα γίνεται παρουσίαση της δομής ενός προγράμματος στη C++, των εντολών εισόδου-εξόδου, της μορφοποίησης των αποτελεσμάτων της εξόδου, των τύπων δεδομένων, της δήλωσης σταθερών, των αριθμητικών τελεστών και της προτεραιότητάς τους, των τελεστών προσαρμογής τύπου και της ανάθεσης τιμής σε μία μεταβλητή.

1. Δομή προγράμματος

Όπως ήδη παρουσιάσαμε στο προηγούμενο κεφάλαιο, το παρακάτω παράδειγμα (2.1) τυπώνει στην οθόνη το μήνυμα «**C++ is Easy!**». Έχοντας υπόψη αυτό το παράδειγμα, θα εξηγήσουμε τη βασική δομή ενός προγράμματος στη C++.

Παράδειγμα 2.1

```
//my first program
#include <iostream>
using namespace std;
int main() {                               // κύρια συνάρτηση προγράμματος
    cout << "C++ is easy!";               // εντολή εξόδου
return 0;
}                                           //τέλος προγράμματος
```

2. Επεξήγηση της δομής του προγράμματος

```
//my first program
```

Πολλές φορές είναι αναγκαίο για τον προγραμματιστή (ειδικότερα όταν έχει να γράψει μεγάλα και πολύπλοκα προγράμματα), να γράφει επεξηγηματικά σχόλια. Κείμενο το οποίο ακολουθεί δύο κάθετες γραμμές // ή συμπεριλαμβάνεται μεταξύ των /* και */ θεωρείται σχόλιο και δεν λαμβάνεται υπόψη από τον μεταγλωττιστή κατά τη διάρκεια της εκτέλεσης του προγράμματος.

```
#include <iostream>
```

Η δήλωση `#include <iostream>`, με την οποία ξεκινά το πρόγραμμα, είναι μία οδηγία για να συμπεριληφθεί η βιβλιοθήκη `iostream`, η οποία μας δίνει τη δυνατότητα να χρησιμοποιήσουμε το πληκτρολόγιο και την οθόνη για είσοδο δεδομένων και έξοδο πληροφοριών. Στο πιο πάνω παράδειγμα, η χρήση της εντολής `cout` δεν θα ήταν δυνατή χωρίς τη συμπερίληψη της βιβλιοθήκης `iostream`.

Γενικά, η δήλωση `#include <filename>` επιτρέπει στον προγραμματιστή να χρησιμοποιήσει τις εντολές που βρίσκονται στη βιβλιοθήκη, το όνομα της οποίας δίνεται μετά την λέξη `include`. Ένα πρόγραμμα μπορεί να χρειάζεται να χρησιμοποιήσει πολλές βιβλιοθήκες. Επίσης, ένας προγραμματιστής μπορεί να δημιουργήσει τη δική του βιβλιοθήκη και να συμπεριλάβει σε αυτή άλλες εντολές.

Άλλα παραδείγματα βιβλιοθηκών:

Βιβλιοθήκη	Χρήση
cmath	Περιέχει μαθηματικές εντολές, όπως η sqrt που υπολογίζει την τετραγωνική ρίζα ενός αριθμού
string	Περιέχει εντολές που αφορούν σε συμβολοσειρές, π.χ. η εντολή size που υπολογίζει το μέγεθος μίας συμβολοσειράς

```
using namespace std;
```

Σκεφτείτε τώρα το εξής παράδειγμα: Έχετε χρησιμοποιήσει (έχετε κάνει include) μία βιβλιοθήκη που χρησιμοποιεί μία εντολή με το όνομα test και θέλετε να δημιουργήσετε και εσείς μία βιβλιοθήκη που θα περιέχει μία εντολή με το όνομα test. Αυτό θα δημιουργούσε σύγχυση στον μεταγλωττιστή, γιατί δε θα ήξερε ποια εντολή test θα χρησιμοποιούσε. Το πρόβλημα αυτό λύνεται με τη χρήση του namespace. Μπορείτε να φανταστείτε ότι το namespace είναι ένα τεράστιο κτήριο που στεγάζει μέσα έναν μεγάλο αριθμό από βιβλιοθήκες και τα βιβλία σε κάθε βιβλιοθήκη είναι μοναδικά. Η συμπερίληψη αυτής της εντολής στην αρχή του προγράμματος μάς επιτρέπει να χρησιμοποιούμε το standard (std) namespace. Θα μπορούσαμε να παραλείψουμε αυτή την εντολή και να γράψουμε το πρόγραμμά μας όπως φαίνεται στο παράδειγμα 2.2.

Παράδειγμα 2.2

```
//my first program
#include <iostream>
int main() {
    std::cout << "C++ is easy!";
    return 0;
}
```

Όπως φαίνεται από το πιο πάνω παράδειγμα, είναι προτιμότερο για λόγους ευκολίας να δηλώνουμε το namespace στην αρχή του προγράμματος. Υπάρχουν πλεονεκτήματα από τη μη δήλωση του namespace στην αρχή, όμως η επεξήγηση είναι πέρα από τους στόχους των σημειώσεων.

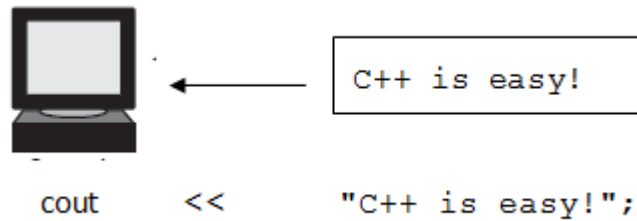
```
int main()
```

Η δήλωση int main() ορίζει τη βασική συνάρτηση του προγράμματος. Όλες οι εντολές της συνάρτησης πρέπει να συμπεριληφθούν ανάμεσα σε άγκιστρα { }. Κάθε πρόγραμμα στη C++ αποτελείται από μία βασική συνάρτηση.

```
cout << "C++ is easy!";
```

Η εντολή cout (standard output στην κονσόλα) χρησιμοποιείται για εμφάνιση πληροφοριών στην οθόνη. Ο τελεστής εξαγωγής (<<) στέλνει οτιδήποτε βρίσκεται στα δεξιά του στο αριστερό του όρισμα (cout), δηλαδή στην οθόνη. Μεταξύ των διπλών εισαγωγικών μπορούμε να συμπεριλάβουμε οποιαδήποτε ακολουθία λατινικών χαρακτήρων, συμβόλων κ.λπ. Το ελληνικό ερωτηματικό σηματοδοτεί το τέλος της εντολής.

Μπορείτε να φανταστείτε ότι η εντολή `cout` δουλεύει όπως φαίνεται στο πιο κάτω σχήμα.



Σχήμα 1. Αναπαράσταση της εντολής `cout`

```
return 0;
```

Η εκτέλεση του προγράμματος ξεκινά με την κλήση της κύριας συνάρτησης (`main`) και καταλήγει με την επιστροφή μίας τιμής σε αυτό. Η επιστροφή της τιμής 0 υποδηλώνει την επιτυχή εκτέλεσή της.

Για παράδειγμα, όταν εκτελέσουμε το πρόγραμμα του παραδείγματος 1, το αποτέλεσμα που εμφανίζεται στην οθόνη θα είναι το εξής:

```
C++ is easy!
Process returned 0 (0x0)    execution time : 0.061 s
Press any key to continue.
```

3. Εντολή εξόδου - `cout`

Όπως είδαμε πιο πάνω, με την εντολή `cout` μπορούμε να τυπώσουμε ένα μήνυμα στην οθόνη. Ας δούμε ένα παράδειγμα με περισσότερες εντολές:

Παράδειγμα 2.3

```
//multiple cout commands
#include <iostream>
using namespace std;
int main() {
    cout << "C++ is easy!";
    cout << "C++ is flexible!";
    cout << "C++ is my choice!";
    return 0;
}
```

Εκτελώντας το πιο πάνω πρόγραμμα, παρατηρούμε ότι οι προτάσεις τυπώνονται όλες σε μία γραμμή, η μία μετά την άλλη:

```
C++ is easy!C++ is flexible!C++ is my choice!
```

Για να μην ενώνονται οι προτάσεις μας, μπορούμε να αφήσουμε ένα κενό μετά από τις δύο πρώτες προτάσεις:

```
cout << "C++ is easy! ";
cout << "C++ is flexible! ";
cout << "C++ is my choice!";
```

Οι πιο κάτω εντολές θα έχουν το ίδιο ακριβώς αποτέλεσμα με το πιο πάνω:

```
cout << "C++ is easy! C++ is flexible! C++ is my choice!";
cout << "C++ is easy! " << "C++ is flexible! " << "C++ is my choice!";
```

4. Μορφοποίηση εξόδου

Μπορούμε να μορφοποιήσουμε την πληροφορία που θα βλέπουμε στην οθόνη μας με διάφορους τρόπους. Παρακάτω περιγράφονται μερικοί από αυτούς.

4.1 Αλλαγή γραμμής (endl)

Για να προχωρήσει το πρόγραμμα στην επόμενη γραμμή μετά από μία εντολή cout, πρέπει να του δώσουμε την εντολή endl. Η λέξη endl είναι ένας χειριστής, ο οποίος εισάγει μία αλλαγή γραμμής (end line).

Ας ξαναγράψουμε το πιο πάνω παράδειγμα ως εξής:

Παράδειγμα 2.4

```
//multiple cout commands
#include <iostream>
using namespace std;
int main() {
    cout << "C++ is easy!" << endl;
    cout << "C++ is flexible!" << endl;
    cout << "C++ is my choice!" << endl;
    return 0;
}
```

Εκτελώντας το πιο πάνω πρόγραμμα, παρατηρούμε ότι οι προτάσεις εμφανίζονται τώρα η μία μετά την άλλη σε ξεχωριστές γραμμές:

```
C++ is easy!
C++ is flexible!
C++ is my choice!
```

Μπορούμε, επίσης, να χρησιμοποιήσουμε το endl μεταξύ προτάσεων της ίδιας εντολής:

```
cout <<"C++ is easy!"<< endl <<"C++ is flexible!"<< endl <<"C++ is my choice!";
```

4.2 Καθορισμός μεγέθους διαστήματος εκτύπωσης (setw)

Μπορούμε να καθορίσουμε το μέγεθος διαστήματος με τη χρήση της εντολής **setw**. Η εντολή αυτή βρίσκεται στη βιβλιοθήκη **iomanip**. Η εντολή setw δέχεται ως παράμετρο έναν ακέραιο αριθμό, που καθορίζει το διάστημα μέσα στο οποίο θα γίνει η εκτύπωση.

Παράδειγμα 2.5

```
#include<iostream>
#include<iomanip> // περιλαμβάνει την εντολή setw
using namespace std;
int main() {
    cout<<"Keep"<<setw(5)<<"calm"<<endl;
```

```
cout<<setw(6)<<"and"<<endl;
cout<<"learn"<<setw(3)<<"to"<<setw(5)<<"code";
return 0;
}
```

Το αποτέλεσμα του πιο πάνω παραδείγματος θα είναι το ακόλουθο (το κενό διάστημα παρουσιάζεται με _).

```
Keep_calm
_ _ _and
learn_to_code
```

Να σημειώσουμε ότι με την εντολή `setw` καθορίζεται το διάστημα εκτύπωσης του κειμένου ή της μεταβλητής που ακολουθεί. Για παράδειγμα στην εντολή

```
cout<<"Keep"<<setw(5)<<"calm"<<endl; //το setw(5) αφορά τη λέξη calm.
```

Παράδειγμα 2.5 (Υλοποίηση στη γλώσσα προγραμματισμού Pascal)

```
Program par2_5;
Begin
    writeln('Keep', 'calm':5);
    writeln('and':6);
    writeln('learn', 'to':3, 'code':5);
End.
```



4.3 Ακρίβεια δεκαδικών ψηφίων (*setprecision*)

Πολλές φορές υπάρχει η ανάγκη να καθορίσουμε τον αριθμό των δεκαδικών ψηφίων που θα εμφανίζονται μετά το αποτέλεσμα μίας πράξης. Η εντολή `setprecision` δέχεται ως παράμετρο έναν ακέραιο αριθμό, που καθορίζει τον αριθμό των δεκαδικών ψηφίων που θα τυπωθούν. Να σημειώσουμε ότι γίνεται στρογγυλοποίηση στον αριθμό των ψηφίων που θα καθορίσει η `setprecision`. Τέλος, είναι απαραίτητο να γράψουμε την εντολή `fixed` πριν από την εντολή `setprecision`, αν θέλουμε να έχουμε ακριβώς τον αριθμό των ψηφίων που καθορίζει η εντολή.

Τόσο η `setprecision` όσο και η `fixed` συμπεριλαμβάνονται στη βιβλιοθήκη `<iomanip>`.

Στο παράδειγμα 2.6 τυπώνεται η τετραγωνική ρίζα του 2 με ακρίβεια 3 δεκαδικών ψηφίων.

Παράδειγμα 2.6 (Υλοποίηση στη γλώσσα προγραμματισμού C++)

```
#include<iostream>
#include<iomanip>
#include<cmath> //για την εντολή sqrt
using namespace std;
int main() {
    cout<<fixed<<setprecision(3)<<sqrt(2)<<endl; // 1.414
    return 0;
}
```

Παράδειγμα 2.6 (Υλοποίηση στη γλώσσα προγραμματισμού Pascal)

```

Program par2_6;
Begin
    write(sqrt(2):0:3);
End.

```

**5. Μεταβλητές και Σταθερές**

Στη C++ όλα τα δεδομένα αποθηκεύονται με δύο τρόπους, είτε ως μεταβλητές είτε ως σταθερές. Μία μεταβλητή περιέχει μία τιμή, η οποία μπορεί να αλλάξει κατά την εκτέλεση του προγράμματος. Αντίθετα, οι σταθερές έχουν σε όλη την διάρκεια της εκτέλεσης μία προκαθορισμένη τιμή που δεν αλλάζει. Αν ένα δεδομένο είναι μεταβλητή ή σταθερά καθορίζεται από τον τρόπο δήλωσής του. Στη C++ τα ονόματα των μεταβλητών και των σταθερών πρέπει να ακολουθούν τους παρακάτω κανόνες:

- Μπορούν να χρησιμοποιηθούν γράμματα, αριθμοί και ο χαρακτήρας (_).
- Ο πρώτος χαρακτήρας είναι υποχρεωτικά λατινικό γράμμα.
- Οι μεταβλητές είναι ευαίσθητες στη χρήση μικρών ή κεφαλαίων χαρακτήρων (case sensitive), σε αντίθεση με την Pascal. Δηλαδή, οι μεταβλητές num και NUM είναι δύο διαφορετικές μεταβλητές.
- Δεσμευμένες λέξεις της C++ δεν μπορούν να χρησιμοποιηθούν ως ονόματα μεταβλητών (π.χ. main, namespace, return, cout).

Οι μεταβλητές και οι σταθερές διακρίνονται και ως προς τον τύπο των δεδομένων που περιέχουν. Αυτό γίνεται για λόγους οικονομίας στη χρήση της μνήμης, επειδή ένας ακέραιος καταλαμβάνει λιγότερο χώρο για να αποθηκευτεί από ό,τι ένας δεκαδικός. Στη C++ υπάρχουν οι ακόλουθοι τύποι δεδομένων:

Τύπος	Διάκριση	Bytes	Εμβέλεια
char	Χαρακτήρας	1	-128 έως 127
short	ακέραιος αριθμός	2	-32,768 έως 32,767
int	ακέραιος αριθμός	4	-2,147,483,648 έως 2,147,483,647
long long	ακέραιος αριθμός	8	$(-2^{63}+1)$ έως $(+2^{63}+1)$
float	πραγματικός αριθμός	4	1.2E-38 έως 3.4E+38
double	πραγματικός αριθμός	8	2.2E-308 έως 1.8E+308
long double	πραγματικός αριθμός	16	3.4E-4932 έως 1.2E+4932
bool	Λογική τιμή	1	true ή false

Στην περίπτωση που επιθυμούμε να ορίσουμε μια μεταβλητή η οποία να είναι συμβολοσειρά, όπως για παράδειγμα το όνομα ενός μαθητή, τότε χρησιμοποιούμε τον τύπο **string**. Θα μάθουμε σε μεταγενέστερο στάδιο περισσότερα για τον τύπο δεδομένων string.

Για να χρησιμοποιηθεί, όμως, μία μεταβλητή πρέπει πρώτα να δηλωθεί, να της δοθεί δηλαδή ένα όνομα και να επιλεγεί ο τύπος δεδομένων που θα αποθηκεύει. Η δήλωση γίνεται γράφοντας έναν τύπο δεδομένων, ακολουθούμενο από ένα όνομα.

Για παράδειγμα:

```
int num;           // δηλώνουμε μία μεταβλητή ακέραιου τύπου με όνομα num
double ratio;      // δηλώνουμε μία μεταβλητή δεκαδικού τύπου με όνομα ratio
bool check;        // δηλώνουμε μία μεταβλητή λογικού τύπου με όνομα check
char alpha, beta;  // δηλώνουμε δύο μεταβλητές τύπου char, ξεχωρίζουν με το κόμμα
```

Ταυτόχρονα με τη δήλωση μπορεί να γίνει και αρχικοποίηση των μεταβλητών. Η εκχώρηση τιμών γίνεται με το σύμβολο ανάθεσης τιμής (=), το οποίο εκχωρεί την τιμή ή το αποτέλεσμα των πράξεων που βρίσκονται στα δεξιά του, στη μεταβλητή που βρίσκεται στα αριστερά του. Σε μεταβλητές char ο χαρακτήρας που θέλουμε να αποθηκεύσουμε θα πρέπει να περικλείεται σε μονά εισαγωγικά.

```
int num = 10;           // δηλώνεται η num και αρχικοποιείται με τιμή 10
int add2 = 10+20;       // δηλώνεται η add2 και παίρνει την τιμή 10+20, δηλαδή 30
double ratio = 25.56;   // δηλώνεται η ratio και αρχικοποιείται με τιμή 25.56
bool check = false;     // δηλώνεται η check και αρχικοποιείται με τιμή false
char alpha='A', beta='B'; // δηλώνονται οι alpha και beta και αρχικοποιούνται με
                          // τιμές 'A' και 'B'
```

Οι σταθερές δηλώνονται είτε όπως οι μεταβλητές, με προσθήκη της δεσμευμένης λέξης const στην αρχή, είτε με την οδηγία define. Κανένας από τους δύο τρόπους δεν επιτρέπει να αλλάξει η τιμή μετά την αρχικοποίηση. Η συντακτική ορθότητα μίας δήλωσης const ελέγχεται αμέσως, ενώ για το #define ελέγχεται μόνο αφού γίνει η αντικατάσταση. Αν υπάρχει λάθος, θα εμφανίζεται στη γραμμή του προγράμματος και όχι στη δήλωση του #define, σε αντίθεση με το const.

Παράδειγμα 2.7

```
#include<iostream>
using namespace std;
#define FPA_PERCENT 0.23           // Δήλωση σταθερής τιμής με define
int main() {
    const float fpa_percent = 0.23; // Δήλωση σταθερής τιμής με const
    return 0;
}
```

6. Εντολή εισόδου - cin

Η εντολή για εισαγωγή δεδομένων από το πληκτρολόγιο είναι η cin (standard input).

Έτσι γράφοντας:

```
int num1;
cin >> num1;           // εντολή εισόδου
```

το πρόγραμμα περιμένει από τον χρήστη να εισάγει έναν ακέραιο αριθμό από το πληκτρολόγιο. Όταν η εντολή `cin` δεχθεί έναν αριθμό, τον αποθηκεύει στη μεταβλητή `num1` και έπειτα τερματίζει. Ο τελεστής (`>>`) διαβάζει από το πληκτρολόγιο (το αριστερό του όρισμα που είναι το `cin`) μία τιμή που την καταχωρίζει στη μεταβλητή που βρίσκεται στα δεξιά του. Με μία εντολή `cin` μπορούμε να διαβάζουμε πολλαπλές μεταβλητές διαφορετικού τύπου:

```
int n;
double pi;
char let;
cin >> n >> pi >> let;
```

Η εισαγωγή των δεδομένων για τον πιο πάνω κώδικα μπορεί να γίνει σε μία γραμμή, διαχωρίζοντας τα δεδομένα με κενό:

```
100 5.67 A
```

ή εισάγοντας τα δεδομένα σε ξεχωριστές γραμμές:

```
100
5.67
A
```

Παράδειγμα 2.8

```
#include<iostream>
using namespace std;
int main() {
    int ak1, ak2, ak3;
    double de1, de2;
    bool test;
    cin >> ak1 >> de1;           //input: 56 78.7
    cin >> ak2 >> de2;           //input: 62 67.3
    cin >> ak3 >> test;          //input: 64 false
    cout << ak1 << " " << ak2 << " " << ak3 << endl; //output: 56 62 64
    cout << de1 << " " << de2 << endl;   //output: 78.7 67.3
    cout << test << endl;             //output: 0 (0=false, 1=true)
    return 0;
}
```

7. Τελεστές

7.1 Τελεστής Ανάθεσης Τιμής (=)

Στην αρχικοποίηση μεταβλητών είδαμε τη χρήση του τελεστή ανάθεσης τιμής (`=`). Χρησιμοποιώντας το σύμβολο αυτό, αποδίδουμε μία τιμή σε μία μεταβλητή.


```
int a;      // δηλώνουμε μία μεταβλητή ακεραίου τύπου με όνομα a
a = 5;      // αποδίδουμε την τιμή 5 στη μεταβλητή a
```

Ο τελεστής μπορεί να ερμηνευτεί απλά και ως: πάρε αυτό που βρίσκεται στο δεξί μέλος της εντολής (μετά το =) κάνε τις πράξεις και εκχώρησε το αποτέλεσμα στη μεταβλητή που βρίσκεται στο αριστερό μέλος.

Παράδειγμα 2.9

```
#include<iostream>
using namespace std;
int main() {
    int a, b = 10;
    a = 5;
    b = a;
    a = 10;
    cout << a << endl;      // a = 10;
    cout << b << endl;      // b = 5;
    return 0;
}
```

7.2 Αριθμητικοί τελεστές (+ , - , * , / , %)

Οι πέντε αριθμητικοί τελεστές που υποστηρίζονται από τη C++ είναι:

+	πρόσθεση
-	αφαίρεση
*	πολλαπλασιασμός
/	διαίρεση
%	υπόλοιπο ακέραιας διαίρεσης

Παράδειγμα 2.10

```
#include<iostream>
using namespace std;
int main() {
    int a, b, c;
    a = 25;                      // a = 25
    b = a - 15;                  // b = 10
    c = a / 10;                  // c = 2 (ακέραια διαίρεση)
    a = a * 3;                   // a = 75
    b = c * 12 + 6;              // b = 30
    c = a % b; // c = 15 (το υπόλοιπο της διαίρεσης 75 / 30 είναι 15)
    cout << a << endl;          // 75
    cout << b << endl;          // 30
}
```

```
cout << c << endl;           // 15
return 0;
}
```

7.3 Τελεστές προσαρμογής τύπου

Οι τελεστές προσαρμογής τύπου (type casting) μάς επιτρέπουν να μετατρέψουμε ένα δεδομένο ενός τύπου σε δεδομένο διαφορετικού τύπου.

```
int n;
float pi = 3.14;
n = (int) pi;           // n = 3
```

Στο παραπάνω παράδειγμα μετατρέπουμε τα δεδομένα (3.14) της μεταβλητής *pi* σε ακέραιο και το αποτέλεσμα (3) το αποθηκεύουμε στη μεταβλητή *n*. Για την προσαρμογή τύπου μπορούμε να χρησιμοποιήσουμε είτε την παραπάνω σημειογραφία, με τις παρενθέσεις να περιβάλλουν το όνομα του τύπου, που είναι συμβατή με τη C++, είτε τη σημειογραφία με τις παρενθέσεις να περιβάλλουν τη μεταβλητή, της οποίας θα αλλάξουμε τον τύπο δεδομένων:

```
n = int (pi);
```

Στο παράδειγμα 2.11 παρουσιάζεται ο υπολογισμός του μέσου όρου 3 ακέραιων αριθμών. Η παρουσίαση του μέσου όρου γίνεται με ακρίβεια 2 δεκαδικών ψηφίων.

Σημείωση: Προσέξτε τον τρόπο με τον οποίο υπολογίζεται ο μέσο όρος.

Παράδειγμα 2.11

```
#include<iostream>
#include<iomanip>
using namespace std;
int main() {
    int a,b,c;
    float mo;
    cout<<"Δώσε 3 ακέραιους αριθμούς: "<<endl;
    cin>>a>>b>>c;
    mo=(float) (a+b+c)/3;    //Μετατρέπουμε το άθροισμα σε δεκαδικό
                           // ή mo=(a+b+c)/3.0;
    cout<<"Ο μέσος όρος είναι "<<fixed<<setprecision(2)<<mo;
return 0;
}
```

7.4 Προτεραιότητα τελεστών

Η σειρά με την οποία γίνονται οι πράξεις αλλάζει με τη χρήση παρενθέσεων. Να χρησιμοποιείτε συχνά παρενθέσεις, γιατί κάνουν πιο σωστή τη σύνταξη του κώδικα και σας προφυλάσσουν από λάθη. Η σειρά με την οποία ενεργούν οι τελεστές είναι:

()	παρενθέσεις
*, /, %	πολλαπλασιασμός, διαίρεση, υπόλοιπο ακέραιας διαίρεσης
+, -	πρόσθεση, αφαίρεση

Όταν οι τελεστές έχουν την ίδια προτεραιότητα, τότε οι πράξεις εκτελούνται με τη σειρά, από αριστερά προς δεξιά.

8. Συναρτήσεις της C++

Στη C++, όπως και στις πλείστες γλώσσες προγραμματισμού, υπάρχουν υλοποιημένες συναρτήσεις που μπορεί να χρησιμοποιήσει ο προγραμματιστής. Ο αριθμός των συναρτήσεων είναι πολύ μεγάλος και δεν είναι απαραίτητο να τις γνωρίζουμε όλες. Μπορείτε να βρείτε σχετικό κατάλογο στη σελίδα <http://www.cplusplus.com/>. Η χρήση της κάθε συνάρτησης απαιτεί και τη δήλωση της βιβλιοθήκης στην οποία ανήκει. Παρακάτω παρουσιάζουμε χρήσιμες συναρτήσεις που υπάρχουν στη βιβλιοθήκη <cmath>

Συνάρτηση	Χρήση	Παράμετροι	Παράδειγμα
<code>sqrt(x)</code>	Επιστρέφει την τετραγωνική ρίζα του αριθμού x. Η επιστρεφόμενη τιμή είναι δεκαδικός αριθμός.	Ένας θετικός αριθμός (ακέραιος ή δεκαδικός).	<code>sqrt(9)=3.0</code>
<code>abs(x)</code>	Επιστρέφει την απόλυτη τιμή του αριθμού x. Η επιστρεφόμενη τιμή εξαρτάται από τον τύπο του αριθμού x.	Ένας αριθμός (ακέραιος ή δεκαδικός).	<code>abs(-2)=2</code> <code>abs(-3.1)=3.1</code>
<code>pow(x, y)</code>	Επιστρέφει το αποτέλεσμα της δύναμης x^y . Η επιστρεφόμενη τιμή είναι δεκαδικός αριθμός.	Δύο αριθμοί (δεκαδικοί).	<code>pow(2.0, 3.0)=8.0</code> <code>pow(2.0, -1.0)=0.5</code>
<code>trunc(x)</code>	Επιστρέφει το ακέραιο μέρος του αριθμού x σε δεκαδική μορφή, αγνοώντας το δεκαδικό μέρος του.	Ένας δεκαδικός αριθμός.	<code>trunc(1.7)=1.0</code> <code>trunc(-2.1)=-2.0</code>
<code>round(x)</code>	Επιστρέφει το ακέραιο μέρος του αριθμού x σε δεκαδική μορφή, στρογγυλοποιημένο στην πλησιέστερη τιμή.	Ένας δεκαδικός αριθμός.	<code>round(1.5)=2.0</code> <code>round(1.4)=1.0</code>

9. Παραδείγματα απλών προγραμμάτων

Παρακάτω παραθέτουμε ορισμένα παραδείγματα για εξάσκηση και κατανόηση των εννοιών που παρουσιάστηκαν σε αυτό το κεφάλαιο. Σε δύο παραδείγματα (2.12 και 2.13) γίνεται παρουσίαση και των λογικών διαγραμμάτων τους. Επίσης, σε επιλεγμένα παραδείγματα δίδονται λύσεις και σε άλλες γλώσσες προγραμματισμού.

Παράδειγμα 2.12

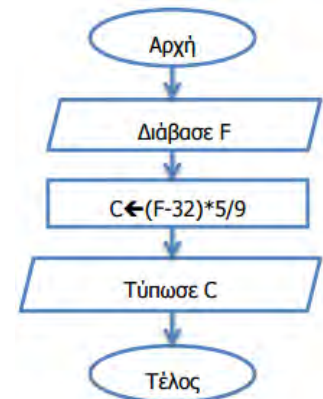
Στην Κύπρο η μονάδα μέτρησης της θερμοκρασίας που χρησιμοποιούμε είναι οι βαθμοί Κελσίου (°C). Υπάρχουν, όμως, άλλες χώρες, όπως οι ΗΠΑ, που χρησιμοποιούν τους βαθμούς

Φάρεναϊτ (°F). Ο τρόπος μετατροπής των βαθμών Φάρεναϊτ σε βαθμούς Κελσίου δίνεται από τον τύπο $C = (F - 32) * 5/9$.

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό που αντιστοιχεί στους βαθμούς Φάρεναϊτ και να υπολογίζει και να τυπώνει τους αντίστοιχους βαθμούς Κελσίου με ακρίβεια ενός δεκαδικού ψηφίου.

Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>
#include<iomanip>
using namespace std;
int F;
float C;
cout<<"Δώσε θερμοκρασία σε Φάρεναϊτ:";
cin>>F;
C=(float) (F-32)*5/9;
cout<<"Οι βαθμοί σε Κελσίου είναι "<<fixed<<setprecision(1)<<C;
return 0;
}
```



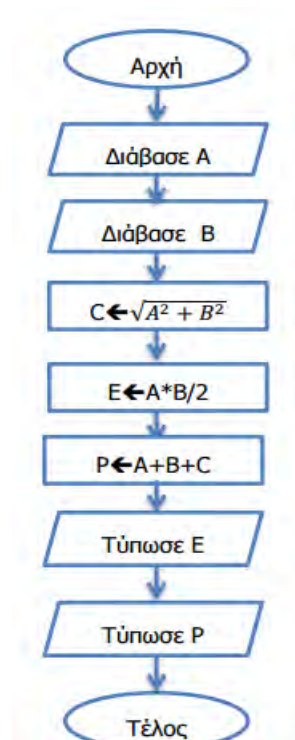
Παράδειγμα 2.13

Να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει το εμβαδόν και την περίμετρο ενός ορθογωνίου τριγώνου όταν είναι γνωστές οι δύο κάθετες πλευρές. Για να υπολογίσετε την περίμετρο, πρέπει να υπολογίσετε την υποτείνουσα, χρησιμοποιώντας το πυθαγόρειο θεώρημα.

(Σημείωση: $E = A*B/2$, $P = A+B+C$, $C = \sqrt{A^2 + B^2}$)

Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>
#include<cmath>
using namespace std;
int main() {
    float a,b,c,e,p;
    cout<<"Δώσε πλευρές α και β:"<<endl;
    cin>>a>>b;
    c=sqrt(a*a+b*b);
    e=a*b/2;
    p=a+b+c;
    cout<<"Εμβαδό="<<e<<endl;
    cout<<"Περίμετρος="<<p<<endl;
    return 0;}
}
```





Παράδειγμα 2.14 – (Ρέστα)

Ένα περιπτερο έχει στο ταμείο του χαρτονομίσματα των δέκα και πέντε ευρώ και νομίσματα των δύο και ενός ευρώ. Να γράψετε πρόγραμμα το οποίο να δέχεται δύο ακέραιους αριθμούς Χ και Υ, που αντιστοιχούν στο ποσό που πρέπει να πληρώσει ο πελάτης και το ποσό που δίνει ο πελάτης. Το πρόγραμμα να υπολογίζει και να τυπώνει τον ελάχιστο αριθμό χαρτονομισμάτων και νομισμάτων, που πρέπει να δώσει ο περιπτεράς.

Παράδειγμα Εισόδου

12 20

Παράδειγμα Εξόδου

3

Επεξήγηση: Ο περιπτεράς πρέπει να δώσει ρέστα οκτώ ευρώ (20-12). Θα δώσει ένα χαρτονόμισμα των πέντε, ένα νόμισμα των δύο και ένα νόμισμα του ενός ευρώ.

Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>
using namespace std;
int main() {
    int x,y,d,p,t,one;
    cin>>x>>y;
    x=y-x;           //ρέστα
    d=x/10;          //χαρτονομίσματα των 10 ευρώ
    p=x%10/5;        //χαρτονομίσματα των 5 ευρώ
    t=x%10%5/2;      //νομίσματα των 2 ευρώ
    one=x%10%5%2;    //νομίσματα του 1 ευρώ
    cout<<d+p+t+one<<endl;
    return 0;
}
```

Υλοποίηση στη γλώσσα προγραμματισμού Python

```
x=int(input())
y=int(input())
x=y-x
d=x//10
p=x%10//5
t=x%10%5//2
one=x%10%5%2
print(d+p+t+one)
```



Παράδειγμα 2.15

Ο πολυεκατομμυριούχος σείχης Αλ Φούρπος είναι λάτρης του κυπριακού ποδοσφαίρου και θέλει να αθλοθετήσει ένα νέο κύπελλο για την πρωταθλήτρια ομάδα. Το κύπελλο θα είναι μία ολόχρυση μπάλα. Ο σείχης θέλει να γράψετε ένα πρόγραμμα, το οποίο να δέχεται την ακτίνα της μπάλας και να υπολογίζει και να τυπώνει τον όγκο της μπάλας-κυπέλλου.

Σημείωση: Η μπάλα είναι μία σφαίρα και ο υπολογισμός του όγκου δίνεται από τον τύπο $\frac{4}{3}\pi R^3$, όπου $\pi=3.14$ και R η ακτίνα της σφαίρας.

Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>
#define p 3.14
using namespace std;
int main() {
    float R,V;
    cout<<"Δώσε την ακτίνα:";
    cin>>R;
    V=4.0/3.0*p*R*R*R;
    cout<<"Όγκος="<<V;
    return 0;
}
```

Υλοποίηση στη γλώσσα προγραμματισμού Pascal

```
program par2_15;
const p = 3.14;
var
    V,R:Real;
Begin
    write('Δώσε ακτίνα:');
    readln(R);
    V:=4/3*p*R*R*R;
    write('Όγκος=',V)
End.
```

*Παράδειγμα 2.16*

Μία χώρα βρήκε πλούσια κοιτάσματα φυσικού αερίου στη θάλασσά της. Μετά από σύσκεψη των αρμόδιων φορέων αποφασίστηκε ότι το 50% της ποσότητας του φυσικού αερίου θα μένει στη χώρα, για να καλύπτονται οι τοπικές ανάγκες, το 40% θα εξάγεται σε χώρες της Ευρώπης και το 10% θα εξάγεται στην Κίνα. Να γράψετε πρόγραμμα το οποίο να δέχεται την ποσότητα του φυσικού αερίου που παράγει η χώρα και να υπολογίζει και να τυπώνει με δύο δεκαδικά ψηφία τις ποσότητες των εξαγωγών σε Κίνα και Ευρώπη, καθώς και την ποσότητα που θα χρησιμοποιηθεί για τις ανάγκες της χώρας. Να γράψετε, επίσης, τον πίνακα προκαταρκτικής εκτέλεσης αν δοθεί ως τιμή εισόδου για τη μεταβλητή παραγωγή ο αριθμός 10,000.00.

Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>
#include<iomanip>
using namespace std;
int main() {
    float paragogi,china,europe,home;
    cout<<"Production=";
    cin>>paragogi;
    china=paragogi * 0.1;
    europe=paragogi * 0.4;
    home=paragogi * 0.5;
    cout<<"Home="<<fixed<<setprecision(2)<<home<<endl;
    cout<<"China="<<fixed<<setprecision(2)<<china<<endl;
    cout<<"Europe="<<fixed<<setprecision(2)<<europe<<endl;
    return 0;
}
```

Πίνακας προκαταρκτικής εκτέλεσης

Μεταβλητές				Παρουσίαση
paragogi	china	europe	home	Home=5000.00 China=1000.00 Europe=4000.00
10000.00	1000.00	4000.00	5000.00	



Ασκήσεις Κεφαλαίου

Άσκηση 2.1

- (1) Να γράψετε το πρόγραμμα του παραδείγματος 2.1, να το αποθηκεύσετε με το όνομα `hello.cpp` και να το εκτελέσετε.
- (2) Να διαγράψετε από το πρόγραμμα `hello.cpp` την εντολή `#include <iostream>` και να το εκτελέσετε ξανά. Τι παρατηρείτε;
- (3) Να τροποποιήσετε το πρόγραμμα `hello.cpp`, ώστε να τυπώνει το όνομά σας στην οθόνη.

Άσκηση 2.2

Να γράψετε προγράμματα, τα οποία να εμφανίζουν τα πιο κάτω στην οθόνη.

(1)

```
Have no fear...
The path is clear...

C++ is here...
```

(2)

```
 *
***
*****
*****
*****
```

Άσκηση 2.3

Να γράψετε προγράμματα, τα οποία να τυπώνουν τα πιο κάτω στην οθόνη.

(1)

```
*
**
***
****
*****
*****
```

(2)

```
 *
  *
   *
    *
   *
  *
 *
*
```

Άσκηση 2.4

Το πιο κάτω πρόγραμμα υπολογίζει και τυπώνει το γινόμενο δύο αριθμών. Να βάλετε τις εντολές στη σωστή σειρά.

```
using namespace std;
return 0;
result = num1 * num2;
int main() {
cout << result;
cin >> num1 >> num2;
}
int num1, num2, result;
#include<iostream>
```


Άσκηση 2.5

Το πιο κάτω πρόγραμμα έχει τέσσερα λάθη. Να τα εντοπίσετε και να τα διορθώσετε.

```
#include<iostream>
using namespace std;
int main{
    int A;
    cin A;
    B = A * 10;
    cout >> A >> A;
return 0;
}
```

Άσκηση 2.6

Να γράψετε τις εντολές δήλωσης των πιο κάτω μεταβλητών:

- (1) Μία μεταβλητή ακέραιου τύπου με όνομα myval.
- (2) Δύο μεταβλητές δεκαδικού τύπου με ονόματα dec1 και dec2, αντίστοιχα. Στη dec2 να ορίσετε ως αρχική τιμή το 9.9.
- (3) Μία μεταβλητή λογικού τύπου, με όνομα found και με αρχική τιμή true.
- (4) Μία μεταβλητή τύπου χαρακτήρα, με όνομα spot και με αρχική τιμή 'X'.

Άσκηση 2.7

Δίνεται το πιο κάτω πρόγραμμα γραμμένο στη γλώσσα προγραμματισμού C++. Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του προγράμματος. Στη θέση του διαστήματος να χρησιμοποιήσετε το σύμβολο «□».

```
#include<iostream>
#include<iomanip>
using namespace std;
int main() {
int A,B,C,D;
double E;
    A = 3;
    B = A + 9;
    C = A + B;
    D = 4 * A * B;
    E = (double) A/B;
    A = B / A;
    C = B % A;
    D = (A + B) / (A + 1);
cout << A << " " << B << " " << C << " " << D << endl;
cout << fixed << setprecision(2) << E << endl;
return 0;}
```

Άσκηση 2.8

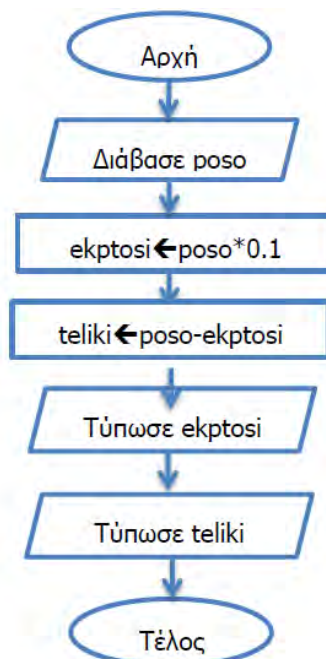
Δίνεται το πιο κάτω πρόγραμμα γραμμένο στη γλώσσα προγραμματισμού C++. Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του προγράμματος. Στη θέση του διαστήματος να χρησιμοποιήσετε το σύμβολο «□».

```
#include<iostream>
#include<iomanip>
using namespace std;

int main() {
    int index;
    double counter;
    bool what;
    char letter;
    index = 4;
    counter = 27.5678;
    what = false;
    letter = 'Z';
    cout << setw(4) << index << endl;
    cout << setw(5) << counter << endl;
    cout << setw(6) << what << setw(3) << letter << endl;
    return 0;
}
```

Άσκηση 2.9

Να μετατρέψετε το πιο κάτω λογικό διάγραμμα σε πρόγραμμα στη γλώσσα προγραμματισμού C++.



Άσκηση 2.10

Να σχεδιάσετε λογικό διάγραμμα και να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να υπολογίζει και να τυπώνει το δεκαπλάσιό του στην οθόνη.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
22	Δώσε έναν ακέραιο αριθμό: Δεκαπλάσιος= 220

Άσκηση 2.11

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να υπολογίζει και να υπολογίζει και να τυπώνει τον επόμενο και τον προηγούμενό του.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
43	Δώσε έναν ακέραιο αριθμό: Επόμενος=44 Προηγούμενος=42

Άσκηση 2.12

Να γράψετε πρόγραμμα το οποίο να δέχεται δύο ακέραιους αριθμούς και στη συνέχεια να υπολογίζει και να τυπώνει το άθροισμα και το γινόμενό τους. Η εκτύπωση θα πρέπει να γίνεται με δεξιά στοίχιση των δέκα χαρακτήρων, σύμφωνα με το παράδειγμα εξόδου.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
12 10	Δώσε δύο ακέραιους αριθμούς: <div style="text-align: right;"> 12 + 10 ----- 22 12 x 10 ----- 120 </div>

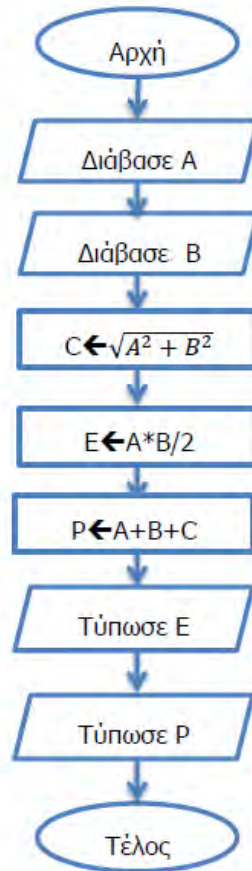
Άσκηση 2.13

Να γράψετε το αποτέλεσμα των πιο κάτω εντολών, οι οποίες είναι γραμμένες στη γλώσσα προγραμματισμού C++, σημειώνοντας τα βήματα που ακολουθήσατε για να φτάσετε στο συγκεκριμένο αποτέλεσμα:

1. `cout<<pow(2,3)+trunc(abs(-8.3))*sqrt(round(63.7));`
2. `cout<<abs(-15)*3+trunc(3.68)/2-6*pow(2,3);`

Άσκηση 2.14

Να μετατρέψετε το πιο κάτω λογικό διάγραμμα σε πρόγραμμα της C++. Να γίνει χρήση των συναρτήσεων `pow` και `sqrt` της βιβλιοθήκης `<cmath>`. Όλες οι μεταβλητές είναι τύπου `double`.

**Άσκηση 2.15**

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να υπολογίζει και να τυπώνει το τετράγωνό του και την τετραγωνική του ρίζα. Να γίνει χρήση των συναρτήσεων `pow` και `sqrt` της βιβλιοθήκης `<cmath>`.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
8	Δώσε έναν ακέραιο αριθμό: Τετράγωνο=64 Τετραγωνική Ρίζα=2.83

Άσκηση 2.16

Να γράψετε πρόγραμμα το οποίο να δέχεται ένα δεκαδικό αριθμό και να υπολογίζει και να τυπώνει το δεκαδικό του μέρος στην οθόνη.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
56.78	Δώσε ένα δεκαδικό αριθμό: Δεκαδικό Μέρος=0.78

Άσκηση 2.17

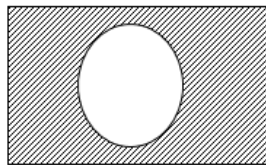
Το ετήσιο επιτόκιο δανεισμού από μία τράπεζα είναι 6%. Να σχεδιάσετε λογικό διάγραμμα και να γράψετε πρόγραμμα το οποίο να δέχεται το ποσό που θα δανειστεί κάποιος και να υπολογίζει και να τυπώνει τον τόκο για τον πρώτο χρόνο.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
1200	Δώσε το ποσό δανεισμού: Τόκος για τον 1 ^ο χρόνο=72.00

**Άσκηση 2.18**

Να γράψετε πρόγραμμα το οποίο να δέχεται τα μήκη a και b του ορθογωνίου και την ακτίνα R του κύκλου και να υπολογίζει και να τυπώνει το εμβαδόν του γραμμοσκιασμένου σχήματος. Το εμβαδό να τυπώνεται με ακρίβεια δύο δεκαδικών ψηφίων. Το π να δηλωθεί ως σταθερά.

Σημείωση: Εμβαδόν ορθογωνίου = $a*b$, Εμβαδόν Κύκλου = $\pi*R^2$ όπου $\pi=3.14$



Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
5 10 3	Δώσε τα μήκη του ορθογωνίου: Δώσε την ακτίνα του κύκλου: Εμβαδό γραμμοσκιασμένου=21.74

Άσκηση 2.19

Η δύναμη της τριβής υπολογίζεται από τον τύπο $F=nmg$, όπου n ο συντελεστής τριβής, m η μάζα του σώματος και g η ένταση του βαρυτικού πεδίου. Να γράψετε πρόγραμμα το οποίο να δέχεται τη μάζα του σώματος και να υπολογίζει και να τυπώνει την τριβή.

Σημείωση: Οι τιμές των n και g που έχουν τις τιμές 0.57 και 9.81 αντίστοιχα να δηλωθούν ως σταθερές.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
60	Δώσε τη μάζα του σώματος: Τριβή=335.50

Άσκηση 2.20

Ένας ψαράς αγόρασε δίχτυα και θέλει να προσθέσει βαρίδια. Γνωρίζει ότι η απόσταση των βαριδιών πρέπει να είναι 30 εκατοστά. Αν κάθε βαρίδιο στοιχίζει 0.20 ευρώ και το μήκος των δικτύων (M) δίνεται σε μέτρα, να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει το πλήθος των βαριδιών που θα χρειαστεί ο ψαράς, καθώς επίσης και το ποσό που πρέπει να πληρώσει.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
8	Δώσε το μήκος των δικτύων (μέτρα): Το πλήθος των βαριδιών είναι 26 Το ποσό που πρέπει να πληρωθεί είναι 5.20

Άσκηση 2.21

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να τυπώνει τους τρεις επόμενους αριθμούς στην οθόνη.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
34	Δώσε έναν ακέραιο αριθμό: Οι 3 επόμενοι αριθμοί είναι: 35 36 37

Άσκηση 2.22

Να γράψετε πρόγραμμα το οποίο να δέχεται το βάρος τριών ανθρώπων και να υπολογίζει και να τυπώνει τον μέσο όρο του βάρους τους. Το βάρος για κάθε άτομο είναι ακέραιος αριθμός και το αποτέλεσμα να τυπώνεται με ακρίβεια ενός δεκαδικού ψηφίου.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
62 59 87	Δώσε το βάρος τριών ανθρώπων: Ο μέσος όρος βάρους είναι 69.3

Άσκηση 2.23

Η Αρχή Ηλεκτρισμού Κύπρου (ΑΗΚ) χρεώνει 30 σεντ την κάθε KWh (κιλοβατώρα). Να σχεδιάσετε λογικό διάγραμμα και να γράψετε πρόγραμμα το οποίο να δέχεται την ένδειξη του μετρητή του προηγούμενου λογαριασμού, καθώς και την τρέχουσα τιμή του μετρητή. Στη συνέχεια, να υπολογίζει και να τυπώνει τις συνολικές KWh, που έχουν καταναλωθεί καθώς επίσης και το ποσό που πρέπει να πληρωθεί.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
2580 3220	Δώσε την προηγούμενη τιμή του μετρητή: Δώσε την τρέχουσα τιμή του μετρητή: Έχουν καταναλωθεί 640 KWh Το ποσό πληρωμής είναι 192.00 ευρώ

Άσκηση 2.24

Ένας γεωργός συσκευάζει τα μήλα σε κιβώτια των 15 και τα πορτοκάλια σε κιβώτια των 10. Αν για κάθε μήλο έχει κέρδος 5 σεντ και για κάθε πορτοκάλι 7 σεντ, να γράψετε πρόγραμμα το οποίο να δέχεται το πλήθος των κιβωτίων των μήλων (M) και το πλήθος των κιβωτίων των πορτοκαλιών (P) και να υπολογίζει και να τυπώνει το ποσό που θα εισπράξει ο γεωργός από τα μήλα και το ποσό που θα εισπράξει από τα πορτοκάλια.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
100 120	Πλήθος των κιβωτίων των μήλων Πλήθος των κιβωτίων των πορτοκαλιών Ποσό που θα εισπράξει από τα μήλα= 75.00 Ποσό που θα εισπράξει από τα πορτοκάλια= 84.00

Άσκηση 2.25

Να γράψετε πρόγραμμα το οποίο να δέχεται δύο δεκαδικούς αριθμούς και να υπολογίζει και να τυπώνει τον μέσο όρο τους χρησιμοποιώντας πέντε δεκαδικά ψηφία.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
5.0 7.1	Δώσε δύο δεκαδικούς αριθμούς: Ο μέσος όρος είναι 6.43182

**Άσκηση 2.26**

Ο δήμος Λάρνακας έχει δημιουργήσει έναν νέο χώρο στάθμευσης στην παραλία Μακένζυ, για να εξυπηρετήσει τους λουόμενους κατά τους καλοκαιρινούς μήνες. Για κάθε αυτοκίνητο ο δήμος εισπράττει 2 ευρώ. Ο δήμος διαθέτει το 10% των μηνιαίων εισπράξεών του για την πληρωμή των υπαλλήλων και το 25% για την αποπληρωμή του δανείου που χρειάστηκε να πάρει, για να ολοκληρώσει το έργο.

Να γράψετε πρόγραμμα το οποίο να δέχεται το πλήθος των αυτοκινήτων (cars) που χρησιμοποίησαν τον χώρο στάθμευσης μέσα σε έναν μήνα και να υπολογίζει και να τυπώνει το ποσό που θα δοθεί για πληρωμές (priors), το ποσό που θα δοθεί για το δάνειο (daneio) και το ποσό που θα μείνει στα ταμεία του δήμου (sinolo).

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
250	Δώσε το πλήθος των αυτοκινήτων: Ποσό πληρωμών= 50.00 ευρώ Ποσό δανείου= 125.00 ευρώ Ποσό που απομένει= 325.00 ευρώ

Άσκηση 2.27

Μία πολυκατοικία έχει τρία διαμερίσματα. Το διαμέρισμα του 1ου ορόφου πληρώνει το 27% των κοινοχρήστων, το διαμέρισμα του 2ου ορόφου το 33% των κοινοχρήστων και το διαμέρισμα του 3ου ορόφου το 40 %. Να γράψετε πρόγραμμα το οποίο να δέχεται το σύνολο των δαπανών και να υπολογίζει και να τυπώνει τα κοινόχρηστα κάθε διαμερίσματος.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
200	Δώσε το σύνολο των δαπανών: Κοινόχρηστα του 1 ^{ου} ορόφου= 54.00 ευρώ Κοινόχρηστα του 2 ^{ου} ορόφου= 66.00 ευρώ Κοινόχρηστα του 3 ^{ου} ορόφου= 80.00 ευρώ

Άσκηση 2.28

Ένας καθηγητής επιθυμεί να δημιουργήσει ένα πρόγραμμα το οποίο να δέχεται τρεις βαθμούς ενός μαθητή. Ο πρώτος βαθμός είναι ο προφορικός βαθμός του μαθητή, ο δεύτερος είναι ο βαθμός του κύριου διαγωνίσματος και ο τρίτος βαθμός είναι ο βαθμός ενός μικρού διαγωνίσματος. Το πρόγραμμα να υπολογίζει και να τυπώνει τον τελικό βαθμό του μαθητή, αν ο προφορικός βαθμός λαμβάνει το 45% της συνολικής βαθμολογίας, ο βαθμός του διαγωνίσματος το 40% και ο βαθμός του μικρού διαγωνίσματος το 15%.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
20 17 18	Δώσε τον προφορικό βαθμό: Δώσε τον βαθμό του κύριου διαγωνίσματος: Δώσε τον βαθμό του μικρού διαγωνίσματος: Τελικός βαθμός = 18.50

Άσκηση 2.29

Ένας μισθωτός έχει ακαθάριστες αποδοχές 22 ευρώ για κάθε ημέρα εργασίας. Από αυτά, το 20% το αποκόπτει η Εφορία ως φόρο και το 10% οι Κοινωνικές Ασφαλίσεις ως ασφαλιστικές εισφορές. Να γράψετε πρόγραμμα το οποίο να δέχεται τον αριθμό των ημερών εργασίας του μισθωτού, να υπολογίζει και να τυπώνει το καθαρό ποσό που θα πάρει ο μισθωτός, το ποσό του φόρου και το ποσό των ασφαλιστικών εισφορών.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
20	Δώσε τον αριθμό των ημερών εργασίας: Το καθαρό ποσό είναι 338.80 ευρώ Το ποσό του φόρου είναι 96.80 ευρώ Το ποσό των ασφαλιστικών είναι 48.40 ευρώ

Άσκηση 2.30

Σε ένα δημοτικό σχολείο, ο Διευθυντής αποφάσισε να υπολογίσει το ποσοστό των αγοριών και το ποσοστό των κοριτσιών κάθε τμήματος. Έτσι επισκέπτεται κάθε τμήμα και καταγράφει τον αριθμό των αγοριών και των κοριτσιών. Να γράψετε πρόγραμμα το οποίο να δέχεται το πλήθος των αγοριών και των κοριτσιών ενός τμήματος και να υπολογίζει και να τυπώνει το ποσοστό των αγοριών και των κοριτσιών.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
12 11	Δώσε το πλήθος των αγοριών: Δώσε το πλήθος των κοριτσιών: Το ποσοστό των αγοριών είναι 52.17% Το ποσοστό των κοριτσιών είναι 47.82%

Άσκηση 2.31

Σε μία αντιπροσωπεία αυτοκινήτων, η τιμή πώλησης ενός μοντέλου είναι 10000 ευρώ. Ο αντιπρόσωπος πρότεινε στην Αριάδνη να πληρώσει προκαταβολικά το 20% του ποσού και το

υπόλοιπο σε 12 ισόποσες δόσεις. Αν το ετήσιο επιτόκιο είναι 2% και υπολογίζεται στο ποσό μετά την αφαίρεση της προκαταβολής, ποιο θα είναι το ποσό της κάθε δόσης; Να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει το ποσό της κάθε δόσης.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
	Το ποσό της κάθε δόσης είναι 680 ευρώ

Άσκηση 2.32

Να γράψετε πρόγραμμα το οποίο να δέχεται δύο πραγματικούς αριθμούς, οι οποίοι θα εκχωρηθούν σε δύο μεταβλητές *a* και *b*. Στη συνέχεια, να γίνεται ανταλλαγή των τιμών τους και να τυπώνεται το περιεχόμενό τους.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
15 28	Δώσε δύο πραγματικούς αριθμούς: Η νέα τιμή του <i>a</i> είναι 28 Η νέα τιμή του <i>b</i> είναι 15

Άσκηση 2.33

Μία δεξαμενή χωράει *N* λίτρα νερό. Για να την γεμίσει κάποιος, χρησιμοποιεί έναν κουβά που χωράει *M* λίτρα νερό. Να γράψετε πρόγραμμα το οποίο να δέχεται τους αριθμούς *N* και *M* και να υπολογίζει και να τυπώνει το πλήθος των κουβάδων που χρειάζονται, για να γεμίσει η δεξαμενή.

Σημείωση: Να θεωρήσετε ότι το *N* είναι ακέραιο πολλαπλάσιο του *M*.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
1000 10	Δώσε τα λίτρα που χωράει η δεξαμενή: Δώσε τα λίτρα που χωράει ο κουβάς: Θα χρειαστούν 100 κουβάδες

Άσκηση 2.34

Ένα ελαιοτριβείο διαθέτει αυτόματο μηχάνημα εισαγωγής, επεξεργασίας και εμφιάλωσης ελαιόλαδου το οποίο συνδέεται απευθείας με τον υπολογιστή. Να γράψετε πρόγραμμα το οποίο να δέχεται την ποσότητα (*m*) του ελαιόλαδου σε kg, καθώς και την πυκνότητα (*d*) σε g/cm³ και να υπολογίζει και να τυπώνει τον συνολικό όγκο (*V*) και το πλήθος (*P*) των 5-λιτρων φιαλών που θα χρειαστεί, για να γίνει η εμφιάλωση.

Σημείωση 1: Ο όγκος υπολογίζεται από τον τύπο $V = m/d$.

Σημείωση 2: Να θεωρήσετε ότι το *V* είναι ακέραιο πολλαπλάσιο του *P*.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
1000 0.9	Δώσε την ποσότητα του ελαιόλαδου σε kg: Δώσε την πυκνότητα του ελαιόλαδου: Ο συνολικός όγκος είναι 1111.11 Το πλήθος των 5-λιτρων είναι 222

Άσκηση 2.35

Ένας μαθητής Λυκείου αγόρασε πρόσφατα κινητό τηλέφωνο με κάρτα και στέλνει σε όλους τους φίλους του μηνύματα. Όταν του τελειώσει η κάρτα, πήγε να εργαστεί στο μαγαζί του πατέρα του, έτσι ώστε με τα χρήματα που θα κερδίσει να μπορέσει να ξαναγοράσει κάρτα για το κινητό του, η οποία κοστίζει 10 ευρώ. Αν γνωρίζουμε πόσα χρήματα του δίνει ο πατέρας του για κάθε ώρα εργασίας, καθώς και τις ώρες που εργάστηκε ο μαθητής, να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει πόσες κάρτες θα μπορέσει να αγοράσει ο μαθητής.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
5 40	Δώσε ποσό για κάθε ώρα εργασίας: Δώσε τις ώρες εργασίας: Το πλήθος των καρτών είναι 20

Άσκηση 2.36

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν διψήφιο ακέραιο αριθμό και να υπολογίζει και να τυπώνει το άθροισμα και το γινόμενο των ψηφίων του.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
43	Δώσε ένα διψήφιο αριθμό: Το άθροισμα των ψηφίων του είναι 7 Το γινόμενο των ψηφίων του είναι 12

Άσκηση 2.37

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν διψήφιο ακέραιο αριθμό και να τον τυπώνει στην οθόνη αντίστροφα, δηλαδή με το ψηφίο των μονάδων στη θέση των δεκάδων και το αντίθετο.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
63	Δώσε ένα διψήφιο ακέραιο αριθμό: Ο αντίστροφος αριθμός είναι 36

Άσκηση 2.38

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να υπολογίζει και να τυπώνει το γινόμενο του αριθμού αυτού με το τελευταίο του ψηφίο.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
125	Δώσε έναν ακέραιο αριθμό: Το γινόμενο είναι 625



Άσκηση 2.39

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν αριθμό σε δευτερόλεπτα και να τον μετατρέπει σε ώρες, λεπτά και δευτερόλεπτα. Το πρόγραμμα να τυπώνει τον αριθμό των ωρών, τον αριθμό των λεπτών και τον αριθμό των δευτερολέπτων που αντιστοιχούν στα δευτερόλεπτα που δόθηκαν.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
17620	Δώσε τον αριθμό των δευτερολέπτων: 4 ώρες 53 λεπτά 40 δευτερόλεπτα

Άσκηση 2.40

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν τετραψήφιο αριθμό και να υπολογίζει και να τυπώνει την τετραγωνική ρίζα του αθροίσματος των ψηφίων του. Το αποτέλεσμα να τυπώνεται με ακρίβεια τριών δεκαδικών ψηφίων.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
1580	Δώσε έναν τετραψήφιο αριθμό: Η τετραγωνική ρίζα ισούται με 3.742

Άσκηση 2.41

Ένας εργολάβος μπορεί να προσλάβει εργάτες με συμβόλαιο των 10, 5, 1 ημερών. Να γράψετε πρόγραμμα το οποίο να δέχεται τις ημέρες που χρειάζεται για να γίνει ένα έργο και να υπολογίζει και να τυπώνει τον ελάχιστο αριθμό συμβολαίων, τα οποία θα χρειαστεί να υπογράψει ο εργολάβος.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
26	Δώσε τον αριθμό των ημερών: Ο ελάχιστος αριθμός συμβολαίων είναι 4



Άσκηση 2.42

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν 4-ψήφιο αριθμό στο οκταδικό σύστημα και να υπολογίζει και να τυπώνει τον αριθμό που προκύπτει από την μετατροπή του αριθμού στο δεκαδικό σύστημα.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
3714	Δώσε έναν τετραψήφιο αριθμό: Ο αριθμός στο δεκαδικό σύστημα είναι 1996

Επεξήγηση: $3 \cdot 8^3 + 7 \cdot 8^2 + 1 \cdot 8^1 + 4 \cdot 8^0 = 1996$

Άσκηση 2.43

Σε ένα κατάστημα ενοικιάσεων DVDs προσφέρονται δύο τρόποι πληρωμής:

- (1) Εγγραφή μέλους 25 € και τιμή ενοικίασης DVD 1.5 €.
- (2) Χωρίς εγγραφή, τιμή ενοικίασης DVD 2 €.

Να γράψετε πρόγραμμα το οποίο να δέχεται τον αριθμό των DVDs που ενοικίασε ένας πελάτης και να υπολογίζει και να τυπώνει το κόστος ενοικίασης χρησιμοποιώντας τους δύο τρόπους πληρωμής που αναφέρονται πιο πάνω.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
6	Δώσε αριθμό των DVDs: Ποσό 1ου τρόπου πληρωμής= 31 ευρώ Ποσό 2ου τρόπου πληρωμής= 12 ευρώ

**Άσκηση 2.44**

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν τριψήφιο ακέραιο αριθμό και να υπολογίζει και να τυπώνει το αποτέλεσμα της διαίρεσής του με το τελευταίο ψηφίο του. Το αποτέλεσμα να εμφανίζεται με ακρίβεια δύο δεκαδικών ψηφίων.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
389	Δώσε έναν τριψήφιο ακέραιο αριθμό: Το αποτέλεσμα είναι ίσο με 43.22

Άσκηση 2.45

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό N ($0 < N \leq 20000000$), που αντιστοιχεί στο χρόνο δευτερόλεπτα και να τον μετατρέπει σε μέρες, ώρες, λεπτά και δευτερόλεπτα.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
86500	Δώσε τον αριθμό των δευτερολέπτων: Μέρες=1 Ώρες= 0 Λεπτά=1 Δευτερόλεπτα= 40

Άσκηση 2.46

Να γράψετε ένα πρόγραμμα το οποίο να δέχεται τα ρέστα που θα πάρει ένας πελάτης σε λεπτά και να υπολογίζει και να τυπώνει το ποσό σε κέρματα των 50, 20, 10, 5, 2 και 1 λεπτών, ώστε αυτά να δίνονται με το μικρότερο δυνατό αριθμό κερμάτων.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
28	Δώσε τα ρέστα (σε λεπτά): Κέρματα των 50= 0 Κέρματα των 20= 1 Κέρματα των 10=0 Κέρματα των 5= 1 Κέρματα των 2= 1 Κέρματα των 1= 1

+ Ασκήσεις Εμπλουτισμού



Άσκηση 2.47

Να γράψετε ένα πρόγραμμα που να δέχεται 3 ακέραιους αριθμούς ($0 \leq x, y, z \leq 1000$) και να υπολογίζει και να τυπώνει το αποτέλεσμα της εξίσωσης $F = x^3 + 2y^2 + 3z$ στην οθόνη.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
2 3 5	Δώσε τρεις ακέραιους αριθμούς: Το αποτέλεσμα της εξίσωσης είναι ίσο με 41



Άσκηση 2.48

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να τον στρογγυλοποιεί στα δύο τελευταία ψηφία του. Το πρόγραμμα να τυπώνει τον στρογγυλοποιημένο αριθμό. Για παράδειγμα, αν δώσουμε 15732 μάς δίνει 15700, αν δώσουμε 15784 μάς δίνει 15800, αν δώσουμε 60 μάς δίνει 100, αν δώσουμε 530 μάς δίνει 500 και αν δώσουμε 17 μάς δίνει 0.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο)	Παράδειγμα Εξόδου 1 (οθόνη)
15732	Δώσε έναν ακέραιο αριθμό Ο αριθμός που προκύπτει είναι το 15700
Παράδειγμα Εισόδου 2 (πληκτρολόγιο)	Παράδειγμα Εξόδου 2 (οθόνη)
15784	Δώσε έναν ακέραιο αριθμό Ο αριθμός που προκύπτει είναι το 15800
Παράδειγμα Εισόδου 3 (πληκτρολόγιο)	Παράδειγμα Εξόδου 3 (οθόνη)
60	Δώσε έναν ακέραιο αριθμό Ο αριθμός που προκύπτει είναι το 100
Παράδειγμα Εισόδου 4 (πληκτρολόγιο)	Παράδειγμα Εξόδου 4 (οθόνη)
530	Δώσε έναν ακέραιο αριθμό Ο αριθμός που προκύπτει είναι το 500

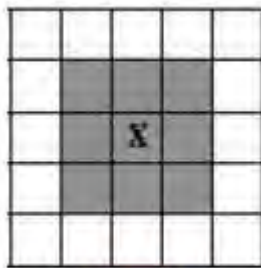
Παράδειγμα Εισόδου 5 (πληκτρολόγιο) 17	Παράδειγμα Εξόδου 5 (οθόνη) Δώσε έναν ακέραιο αριθμό Ο αριθμός που προκύπτει είναι το 0
--	--

Άσκηση 2.49

Να γράψετε πρόγραμμα το οποίο να δέχεται τη χωρητικότητα ενός σκληρού δίσκου σε Gb (Gb = 1024 Mb) και να την τυπώνει αναλυτικά σε Mb, Kb και bytes.

Σημείωση: Να γίνει χρήση τύπου long long int, για να αποφευχθεί η υπερχείλιση.

Παράδειγμα Εισόδου (πληκτρολόγιο) 500	Παράδειγμα Εξόδου (οθόνη) Δώσε τη χωρητικότητα σε Gb: Αντιστοιχεί σε 512000 MB Αντιστοιχεί σε 524288000 KB Αντιστοιχεί σε 536870912000 bytes
---	---

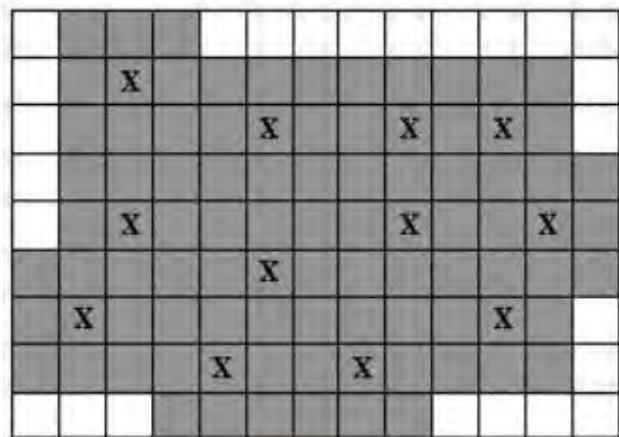
**Άσκηση 2.50**

Εικόνα 1

Είμαστε στα μέσα του Β' Παγκοσμίου Πολέμου και ο πόλεμος κάτω από την θάλασσα του Ατλαντικού μαίνεται σφοδρός. Τα Γερμανικά υποβρύχια βρίσκονται σε πλεονεκτική θέση, καθώς οι Σύμμαχοι δεν μπορούν να «σπάσουν» τους κωδικούς επικοινωνίας τους μέσω της συσκευής Enigma. Για καλή τύχη των Συμμάχων, το γερμανικό υποβρύχιο U-571, που μεταφέρει μία τέτοια συσκευή, παθαίνει ζημιά, βυθίζεται και περιμένει βοήθεια από άλλο γερμανικό υποβρύχιο. Οι Σύμμαχοι τότε αποφασίζουν ότι αυτή είναι η στιγμή να δράσουν και να πάρουν στα χέρια τους τη συσκευή. Έχοντας μπροστά τους μία

μεγάλη υποθαλάσσια έκταση να καλύψουν, οι Σύμμαχοι έχουν στη διάθεσή τους ανιχνευτικά σκάφη, εφοδιασμένα με ειδικά σόναρ. Τα ηχοεντοπιστικά συστήματα σόναρ είναι ηλεκτροακουστικές συσκευές, που εκμεταλλεύονται τη διάδοση των κυμάτων ηχητικής ενέργειας μέσα στη θαλάσσια μάζα, όπως ακριβώς τα συστήματα ραντάρ. Ο σκοπός των συστημάτων σόναρ είναι ο εντοπισμός και η παρακολούθηση υποβρυχίων σκαφών. Κάθε σκάφος, με το σόναρ που διαθέτει, μπορεί να καλύψει μία υποθαλάσσια περιοχή 9 km², όπως φαίνεται και στην πιο πάνω εικόνα (Εικόνα 1).

Οι Σύμμαχοι γνωρίζοντας τις διαστάσεις της περιοχής που πρέπει να καλύψουν, θέλουν να μπορούν να υπολογίσουν τον ελάχιστο αριθμό ανιχνευτικών σκαφών που θα χρειαστούν για να καλύψουν ολόκληρη την περιοχή. Για παράδειγμα, σε μία περιοχή 117km², όπως φαίνεται στην πιο πάνω εικόνα (Εικόνα 2), θα χρειαστούν συνολικά 12 ανιχνευτικά



Εικόνα 2

σκάφη. Τα σημεία που εφάπτονται στα όρια της περιοχής δεν χρειάζεται να ανιχνευτούν, καθώς εκεί το θαλάσσιο βάθος δεν είναι αρκετά μεγάλο, για να καταφύγει ένα υποβρύχιο. Ο χρόνος πιέζει και το Γερμανικό υποβρύχιο πλησιάζει. Θα καταφέρουν οι Σύμμαχοι να εντοπίσουν το υποβρύχιο και να πάρουν τη συσκευή πριν να είναι πολύ αργά;

Δεδομένα εισόδου

Δύο ακέραιοι αριθμοί x, y ($3 \leq x, y \leq 10000$), οι διαστάσεις της περιοχής.

Δεδομένα εξόδου

Ένας ακέραιος αριθμός, ο ελάχιστος αριθμός ανιχνευτικών σκαφών που θα χρειαστούν.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
9 13	Δώσε τις διαστάσεις: Ο αριθμός των σκαφών είναι 12



Άσκηση 2.51

Σας δίνεται ένας 5-ψήφιος δυαδικός αριθμός. Να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει το συμπλήρωμά του. Το συμπλήρωμα ενός δυαδικού αριθμού υπολογίζεται, μετατρέποντας τα 0 σε 1 και τα 1 σε 0.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο)	Παράδειγμα Εξόδου 1 (οθόνη)
10100	Δώσε δυαδικό αριθμό: Το συμπλήρωμα είναι 01011
Παράδειγμα Εισόδου 2 (πληκτρολόγιο)	Παράδειγμα Εξόδου 2 (οθόνη)
00111	Δώσε δυαδικό αριθμό: Το συμπλήρωμα είναι 11000



Άσκηση 2.52

Ο Παντελής έχει αγοράσει μία αριθμητική κλειδαριά, την οποία βλέπετε στην πιο κάτω εικόνα:



Η κλειδαριά, όπως βλέπετε και στην εικόνα, αποτελείται από 40 διαφορετικά σημεία, αριθμημένα από το 0 μέχρι το 39. Ο συνδυασμός της κλειδαριάς αποτελείται από τρεις αριθμούς (π.χ. 13, 34, 8), στο διάστημα $[0...39]$. Για να ανοίξει η κλειδαριά, ο Παντελής πρέπει να κάνει τα εξής βήματα:

- Κάνε δύο πλήρεις, δεξιόστροφες περιστροφές της κλειδαριάς.
- Συνέχισε δεξιόστροφα μέχρι να φτάσεις στον πρώτο αριθμό του συνδυασμού της κλειδαριάς.
- Κάνε μία πλήρη αριστερόστροφη περιστροφή της κλειδαριάς.
- Συνέχισε την αριστερόστροφη περιστροφή μέχρι να φτάσεις στον δεύτερο αριθμό του συνδυασμού.
- Γύρισε δεξιόστροφα την κλειδαριά μέχρι να φτάσεις στον τρίτο αριθμό του συνδυασμού.
- Τράβηξε την κλειδαριά, για να ξεκλειδώσει.

Αν σας δοθεί το αρχικό σημείο που δείχνει η κλειδαριά, να βρείτε πόσες μοίρες συνολικά πρέπει να περιστραφεί (δεξιόστροφα και αριστερόστροφα), για να ξεκλειδώσει η κλειδαριά.

Δεδομένα εισόδου

Τέσσερις ακέραιοι αριθμοί (x, a, b, c) στο διάστημα $[0...39]$. Όπου x , το σημείο που δείχνει αρχικά η κλειδαριά και a, b, c , ο συνδυασμός που ξεκλειδώνει την κλειδαριά.

Δεδομένα εξόδου

Ένας ακέραιος, το σύνολο των μοιρών που πρέπει να περιστραφεί η κλειδαριά, για να ξεκλειδώσει.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
0 13 34 8	Δώσε 4 αριθμούς: Το σύνολο των μορίων είναι 1746

B7.1.3 Δομές Διακλάδωσης

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ❖ Να αναγνωρίζουμε τους συγκριτικούς τελεστές και να αναφέρουμε λεκτικά τι αναπαριστά ο καθένας
- ❖ Να προβλέπουμε το αποτέλεσμα (true ή false) μίας απλής συνθήκης με έναν συγκριτικό τελεστή δεδομένων των τιμών των μεταβλητών που χρησιμοποιούνται
- ❖ Να διατυπώνουμε απλές συνθήκες με έναν συγκριτικό τελεστή, που να ανταποκρίνονται σε συγκεκριμένη εκφώνηση
- ❖ Να αναγνωρίζουμε τους λογικούς τελεστές AND (&&) OR (||) και NOT (!) και να αναφέρουμε λεκτικά τι αναπαριστά ο καθένας
- ❖ Να προβλέπουμε το αποτέλεσμα (true ή false) μίας συνθήκης με μέχρι δύο συγκριτικούς και έναν λογικό τελεστή δεδομένων των τιμών των μεταβλητών που χρησιμοποιούνται
- ❖ Να διατυπώνουμε συνθήκες με μέχρι δύο συγκριτικούς τελεστές και έναν λογικό τελεστή που να ανταποκρίνονται σε συγκεκριμένη εκφώνηση
- ❖ Να χρησιμοποιούμε τους κανόνες προτεραιότητας, για να προβλέπουμε το αποτέλεσμα (true ή false) μίας συνθήκης με πολλούς συγκριτικούς και λογικούς τελεστές δεδομένων των τιμών των μεταβλητών που χρησιμοποιούνται
- ❖ Να διατυπώνουμε συνθήκες με περισσότερους από έναν λογικούς και συγκριτικούς τελεστές, που να ανταποκρίνονται σε συγκεκριμένη εκφώνηση
- ❖ Να διατυπώνουμε τη δομή και το λογικό διάγραμμα της δομής διακλάδωσης if...
- ❖ Να διατυπώνουμε τη δομή και το λογικό διάγραμμα της δομής διακλάδωσης if...else.
- ❖ Να σχεδιάζουμε το λογικό διάγραμμα και τις αντίστοιχες εντολές προγράμματος, που να ανταποκρίνονται σε συγκεκριμένη εκφώνηση που αναφέρεται σε δομή διακλάδωσης
- ❖ Να αναγνωρίζουμε την ένθετη (nested) δομή διακλάδωσης (if .. else if ..) ως λύση σε προβλήματα με σύνθετη λογική.
- ❖ Να μετατρέπουμε εκφώνηση προβλήματος με σύνθετη λογική σε ένθετη δομή διακλάδωσης.
- ❖ Να αναγνωρίζουμε την περιπτωσιακή δομή (switch) ως λύση σε προβλήματα με σύνθετη λογική
- ❖ Να μετατρέπουμε εκφώνηση προβλήματος με σύνθετη λογική σε περιπτωσιακή δομή
- ❖ Να μετατρέπουμε ένθετη δομή διακλάδωσης σε περιπτωσιακή δομή και αντίστροφα
- ❖ Να αναγνωρίζουμε τους περιορισμούς και τα πλεονεκτήματα της περιπτωσιακής δομής σε σχέση με την ένθετη δομή διακλάδωσης
- ❖ Να καθορίζουμε το πρόβλημα με ακρίβεια, συγκεκριμένα: να εντοπίζουμε/διακρίνουμε τα Δεδομένα, τις Πληροφορίες και την Επεξεργασία.
- ❖ Να δημιουργούμε το Λογικό Διάγραμμα για την επίλυση του προβλήματος
- ❖ Να επιλέγουμε κατάλληλη δομή διακλάδωσης, ανάλογα με τις δυνατότητες, τους περιορισμούς και τα χαρακτηριστικά της για επίλυση του προβλήματος
- ❖ Να μεταφράζουμε ένα λογικό διάγραμμα σε πρόγραμμα, με τη χρήση του προγραμματιστικού περιβάλλοντος, ώστε να επιλυθεί το πρόβλημα
- ❖ Να επιλέγουμε κατάλληλα δεδομένα και στρατηγική για έλεγχο του προγράμματος
- ❖ Να ελέγχουμε την ορθότητα της λύσης του προβλήματος, χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης και της χρήσης του debugger για επαλήθευση
- ❖ Να μελετούμε έτοιμο πρόγραμμα, το οποίο περιλαμβάνει δομές διακλάδωσης και να εντοπίζουμε βασικά μέρη του, τα οποία συνδέονται με πτυχές του προβλήματος που επιλύει
- ❖ Να συμπληρώνουμε ένα έτοιμο πρόγραμμα με δομή/δομές διακλάδωσης, ώστε να αποτελεί λύση ενός διαφοροποιημένου προβλήματος
- ❖ Να εντοπίζουμε και να αναγνωρίζουμε σε ένα πρόγραμμα πρότυπα σχεδίασης και στρατηγικές (design patterns, τμήματα κώδικα)
- ❖ Να χρησιμοποιούμε πρότυπα σχεδίασης και στρατηγικές που εντοπίσαμε ως εργαλεία επίλυσης προβλημάτων σε νέα προγράμματά μας.

1. Εισαγωγή

Τα προγράμματα που υλοποιήσαμε μέχρι στιγμής εκτελούν όλες τις εντολές τους, ξεκινώντας από την αρχή και φτάνοντας στο τέλος κάθε προγράμματος. Στην πραγματικότητα αυτό συμβαίνει σπάνια. Τα προγράμματα παίρνουν αποφάσεις για το ποιες εντολές θα εκτελέσουν, εξετάζοντας κάποια συνθήκη. Σε αυτό το κεφάλαιο θα εξετάσουμε πώς μπορούμε να εισάγουμε συνθήκες σε ένα πρόγραμμα. Ένα πρόγραμμα μπορεί να πάρει μία ή πολλές

αποφάσεις, βασισμένες σε απλές ή σύνθετες συνθήκες. Σε αυτό το κεφάλαιο θα εξετάσουμε τις εντολές με τις οποίες ένα πρόγραμμα λαμβάνει αποφάσεις, καθώς και τους λογικούς τελεστές που χρησιμοποιούμε, για να φτιάξουμε τις συνθήκες, βάσει των οποίων θα ληφθούν οι αποφάσεις.

2. Απλή δομή διακλάδωσης

Θα ξεκινήσουμε με το εξής απλό παράδειγμα: Σας δίνεται ένας αριθμός και θέλετε να ελέγξετε αν αυτός ο αριθμός είναι άρτιος ή περιττός.

Παράδειγμα 3.1

```
#include<iostream>
using namespace std;
int main() {
    int num;
    cout<<"Num=";
    cin>>num;           // Εισαγωγή αριθμού
    if (num%2==0) {      // Έλεγχος συνθήκης. Αν ο αριθμός
                        // διαιρείται ακριβώς με το 2
        cout<<"Άρτιος"; // Εκτελείται όταν ισχύει η συνθήκη
    }
    else {
        cout<<"Περιττός"; // Εκτελείται όταν δεν ισχύει η συνθήκη
    }
    return 0;
}
```

Η εντολή που χρησιμοποιούμε για να πάρει μία απόφαση το πρόγραμμά μας είναι η εντολή if. Η σύνταξη της εντολής έχει ως ακολούθως:

```
if (λογική έκφραση) {
    εντολές που εκτελούνται, αν ισχύει η λογική έκφραση
}
else {
    εντολές που εκτελούνται, αν δεν ισχύει η λογική έκφραση
}
```

Παρατηρήσεις:

- Μπορεί να έχουμε την εντολή if χωρίς το else. Δηλαδή, σε περίπτωση που δεν ισχύει η λογική έκφραση να μην εκτελείται άλλη εντολή.
- Τα άγκιστρα μπορούν να παραλείπονται, αν έχουμε μόνο μία εντολή.

3. Συγκριτικοί τελεστές (==, !=, >, <, >=, <=)

Προκειμένου να συγκρίνουμε δύο αριθμητικές παραστάσεις ως προς την τιμή τους χρησιμοποιούμε τους συγκριτικούς τελεστές:

==	ίσο
!=	διάφορο / άνισο
>	μεγαλύτερο
<	μικρότερο
>=	μεγαλύτερο ή ίσο
<=	μικρότερο ή ίσο

Οι συγκριτικοί τελεστές χρησιμοποιούνται, για να σχηματίσουμε λογικές συνθήκες. Μία λογική συνθήκη απαρτίζεται από τρία μέρη: τον συγκριτικό τελεστή στη μέση και τις εκφράσεις τις οποίες συγκρίνουμε στα δύο άκρα του. Το αποτέλεσμα είναι μία λογική τιμή, είτε αληθής (true) είτε ψευδής (false).

```
cout << (7 == 5);           // 0 (false)
cout << (5 > 4);            // 1 (true)
cout << (3 != 2);           // 1 (true)
cout << (6 >= 6);           // 1 (true)
cout << (5 < 5);            // 0 (false)
```

Σημείωση: Ο τελεστής (=) είναι τελεστής ανάθεσης τιμής, αποδίδει ό,τι υπάρχει στα δεξιά του στη μεταβλητή που βρίσκεται στα αριστερά του. Ο τελεστής (==) είναι τελεστής σύγκρισης και συγκρίνει εάν τα δύο μέλη στα δεξιά και αριστερά του είναι ίσα, επιστρέφοντας true ή false.

Παράδειγμα 3.2

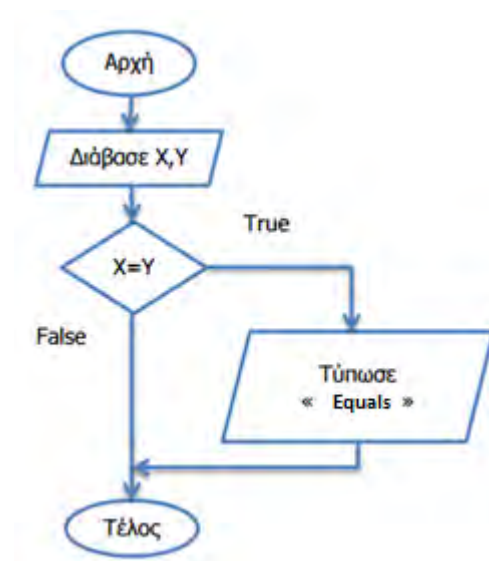
Να γράψετε πρόγραμμα το οποίο να δέχεται δύο ακέραιους αριθμούς. Αν οι αριθμοί είναι ίσοι να τυπώνει το μήνυμα «Equals».

Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>

using namespace std;

int main() {
    int x,y;
    cout<<"Give X,Y:";
    cin>>x>>y;
    if(x==y)
        cout<<"Equals";
    return 0;
}
```



Υλοποίηση στη γλώσσα προγραμματισμού Pascal



```
program par3_2;
var
    x,y:integer;
Begin
    write('Give X,Y');
    readln(x,y);
    if x=y then
        write('Equals')
End.
```

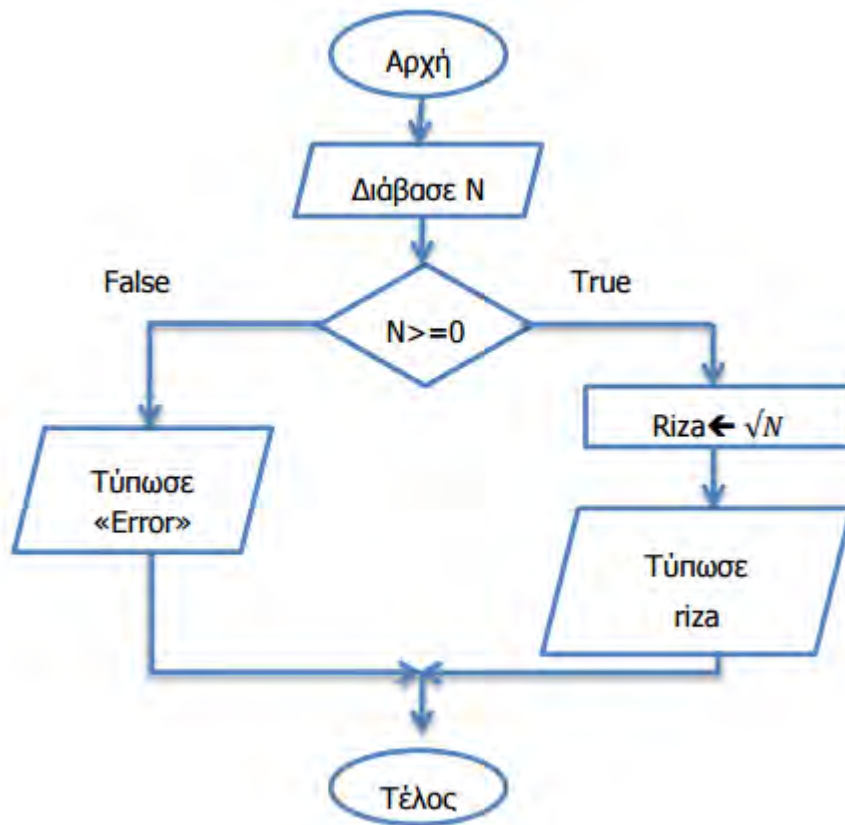
Παράδειγμα 3.3

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να τυπώνει την τετραγωνική του ρίζα με ακρίβεια 2 δεκαδικών ψηφίων. Σε περίπτωση που δοθεί αρνητικός αριθμός, να τυπώνει την λέξη «Λάθος αριθμός».

Υλοποίηση στη γλώσσα προγραμματισμού C++



```
#include<iostream>
#include<cmath>
#include<iomanip>
using namespace std;
int main() {
    int N;
    float Riza;
    cout<<"N=";
    cin>>N;
    if(N>=0) {
        Riza=sqrt(N);
        cout<<"Ρίζα="<<fixed<<setprecision(2)<<Riza;
    }
    else
        cout<<"Λάθος αριθμός";
    return 0;
}
```



Παράδειγμα 3.4

Ένα σχολείο δημιουργεί τους τριψήφιους αριθμούς μητρώου των μαθητών του ως εξής:

- Αν το ψηφίο των μονάδων είναι άρτιος αριθμός, τότε σημαίνει ότι ο μαθητής είναι αγόρι, διαφορετικά είναι κορίτσι.
- Αν το ψηφίο των δεκάδων είναι το 5, τότε σημαίνει ότι ο μαθητής είναι εκτοπισμένος, διαφορετικά δεν είναι.
- Αν ο αριθμός των εκατοντάδων είναι μεγαλύτερος του 7, τότε σημαίνει ότι ο μαθητής είναι από πολύτεκνη οικογένεια, διαφορετικά δεν είναι.

Να γράψετε πρόγραμμα το οποίο να δέχεται τον αριθμό μητρώου (AM) του μαθητή και να τυπώνει τις πληροφορίες του.

Υλοποίηση στη γλώσσα προγραμματισμού C++

```

#include<iostream>
using namespace std;

int main() {
    int AM, mon, dek, ek;
    cout<<"AM=";
    cin>>AM;
    mon=AM%10;
    ek=AM/100;
    dek=AM%100/10;
  
```

```

    if (mon%2==0)
        cout<<"Αγόρι"<<endl;
    else
        cout<<"Κορίτσι"<<endl;
    if (dek==5)
        cout<<"Εκτοπισμένος"<<endl;
    else
        cout<<"Δεν είναι εκτοπισμένος"<<endl;
    if (ek>7)
        cout<<"Πολύτεκνος"<<endl;
    else
        cout<<"Δεν είναι πολύτεκνος"<<endl;
return 0;
}

```

Υλοποίηση στη γλώσσα προγραμματισμού Python

```

AM = int(input())
mon = AM % 10
ek = AM // 100
dek = AM % 100 // 10
if mon % 2==0 :
    print('Agori')
else:
    print('Koritsi')
if dek == 5:
    print('Ektopismenos')
else :
    print('Den einai ektopismenos')
if ek > 7:
    print('Politeknos')
else:
    print('Den einai politeknos')

```



4. Λογικοί Τελεστές

Στα παραδείγματα που παρουσιάσαμε μέχρι στιγμής, οι λογικές εκφράσεις ήταν απλές και εξέταζαν μόνο μία συνθήκη. Πολλές φορές, όμως, είναι αναγκαίο να εξετάσουμε ταυτόχρονα περισσότερες από μία συνθήκες. Αυτό επιτυγχάνεται με τη χρήση των λογικών τελεστών.

Οι λογικοί τελεστές, οι οποίοι θα εξετάσουμε, είναι ο τελεστής And (&&), ο τελεστής Or (||) και ο τελεστής Not (!). Ο πίνακας αληθείας των λογικών τελεστών είναι ο ακόλουθος:

X	Y	X && Y	X Y	!X
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

Η προτεραιότητα των τελεστών είναι (ξεκινώντας από τη μεγαλύτερη):

1. !(Not)
2. && (And)
3. || (Or)

Παράδειγμα 3.5

Ένα τρίγωνο χαρακτηρίζεται ορθογώνιο, αν έχει μία ορθή γωνία (90 μοίρες). Επιπρόσθετα, το άθροισμα των γωνιών ενός τριγώνου πρέπει να είναι 180. Να γράψετε πρόγραμμα το οποίο να δέχεται το μέγεθος των γωνιών του τριγώνου και να τυπώνει «Ορθογώνιο Τρίγωνο», αν το τρίγωνο είναι ορθογώνιο, διαφορετικά να τυπώνει «Όχι ορθογώνιο Τρίγωνο».

Να γράψετε, επίσης, την προκαταρκτική εκτέλεση αν δοθούν για τις τρεις γωνιές οι τιμές εισόδου:

α) 100, 60, 20 και β) 90, 50, 40.

Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>
using namespace std;
int main() {
    int a,b,c,total;
    cout<<"Γωνιές=";
    cin>>a>>b>>c;
    total=a+b+c;

    if((a==90 || b==90 || c==90) && total==180)
        cout<<" Ορθογώνιο Τρίγωνο ";
    else
        cout<<" Όχι Ορθογώνιο Τρίγωνο ";

    return 0;
}
```


Πίνακας προκαταρκτικής εκτέλεσης

Μεταβλητές			Αποφάσεις	T/F	Παρουσίαση
a	b	c	((a==90 b==90 c==90) && total==180)		
100	60	20	((100==90 60==90 20==90) && 180==180)	F	Γωνίες=
					Όχι ορθογώνιο Τρίγωνο
90	50	40	((90==90 50==90 40==90) && 180==180)	T	Ορθογώνιο Τρίγωνο

Παράδειγμα 3.6

Ένα έτος χαρακτηρίζεται δίσεκτο, όταν διαιρείται ακριβώς με το 4, αλλά δεν διαιρείται με το 100 ή αν διαιρείται ακριβώς με το 400. Να γράψετε πρόγραμμα το οποίο να δέχεται το έτος και να τυπώνει τον αριθμό 1, αν το έτος είναι δίσεκτο, διαφορετικά να τυπώνει τον αριθμό 0.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 2000	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε έναν ακέραιο αριθμό: 1
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 1900	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε έναν ακέραιο αριθμό: 0

Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>
using namespace std;
int main() {
    int etos;
    cout<<"Year=";
    cin>>etos;
    if(etos%4==0 && etos%100 !=0 || etos%400==0)
        cout<<"1";
    else
        cout<<"0";
    return 0;
}
```

5. Ένθετη δομή διακλάδωσης – Nested if

Μέχρι στιγμής έχουμε εξετάσει τις παρακάτω περιπτώσεις:

- Το πρόγραμμα έχει μία συνθήκη και δεν εκτελούνται εντολές, αν δεν ισχύει η συνθήκη.
- Το πρόγραμμα έχει μία συνθήκη και εκτελούνται εντολές, αν δεν ισχύει η συνθήκη
- Υπάρχουν περισσότερες από μία συνθήκες στο πρόγραμμα.

- Υπάρχουν σύνθετες λογικές εκφράσεις μέσα στη συνθήκη.

Στις περιπτώσεις που πρέπει να συνδυάσουμε διάφορες λογικές εκφράσεις, για να αποφασίσουμε ποιες εντολές θα χρησιμοποιηθούν, τότε έχουμε ένθετη δομή διακλάδωσης. Μόλις βρεθεί λογική έκφραση που είναι αληθής (true), εκτελούνται οι αντίστοιχες εντολές και η δομή διακλάδωσης τερματίζεται. Είναι σαν να έχουμε πολλά if, το ένα μετά το άλλο και θα εκτελεστούν οι εντολές του πρώτου, του οποίου η λογική έκφραση θα είναι true. Οι υπόλοιπες εντολές θα αγνοηθούν.

Σύνταξη εντολής if..else if

```
if (λογική έκφραση 1) {  
    εντολές που εκτελούνται, αν ισχύει η λογική έκφραση 1  
}  
else if (λογική έκφραση 2) {  
    εντολές που εκτελούνται, αν ισχύει η λογική έκφραση 2  
}  
else {  
    εντολές που εκτελούνται, αν δεν ισχύει η λογική έκφραση 1 ή 2  
}
```

Παράδειγμα 3.7

Να γράψετε πρόγραμμα το οποίο να δέχεται την τελική βαθμολογία (V) ενός μαθητή. Αν ο βαθμός δεν είναι μεταξύ του 1-20 συμπεριλαμβανομένων, να τυπώνει μήνυμα λάθους, αλλιώς να τυπώνει τη βαθμολογία βάσει των πιο κάτω στοιχείων :

1-9 → E, 10 - 12 → Δ, 13 - 15 → Γ, 16 - 18 → Β, 19 - 20 → Α

Υλοποίηση στη γλώσσα προγραμματισμού C++

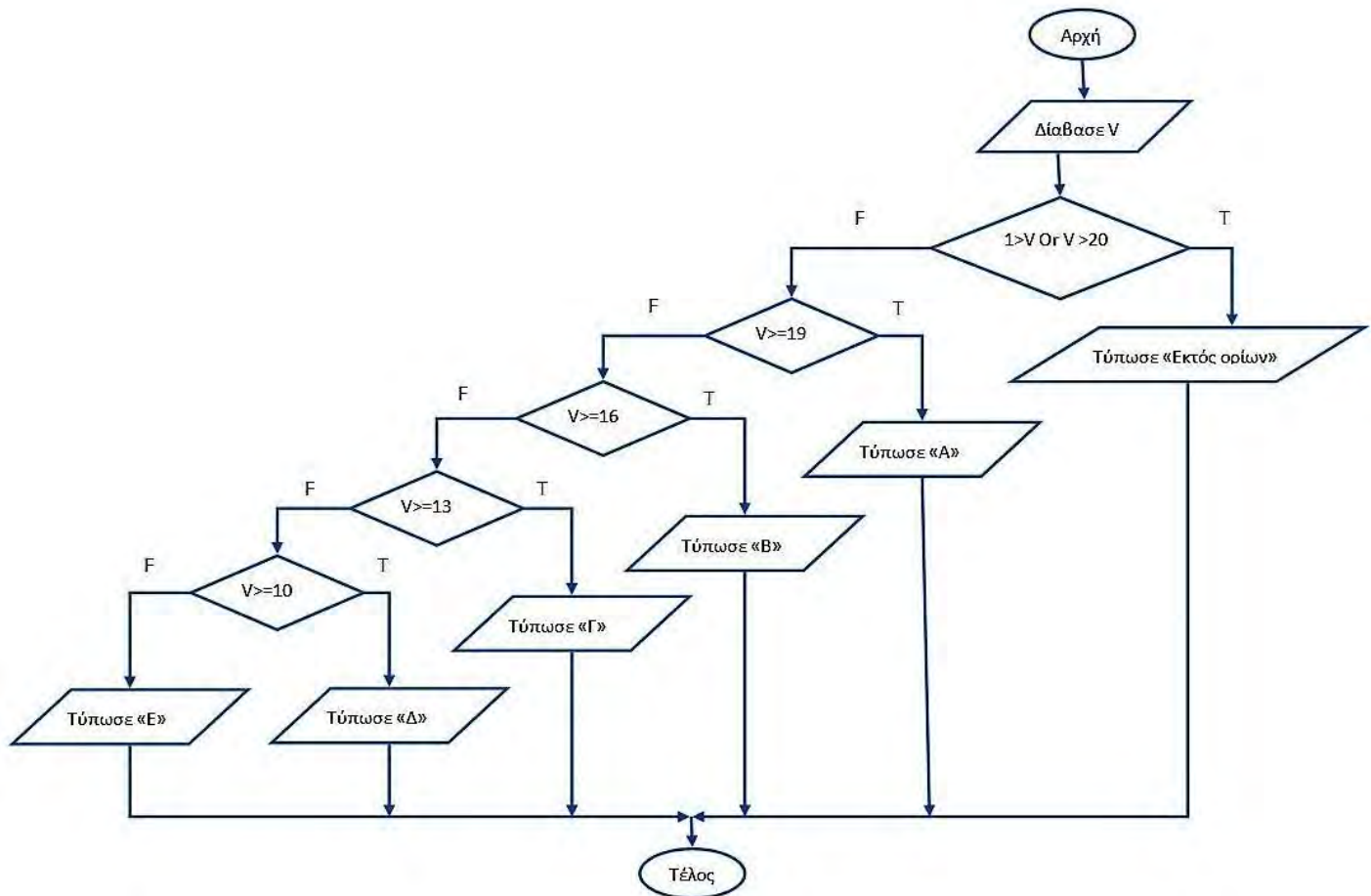
```
#include<iostream>  
  
using namespace std;  
  
int main() {  
    int V;  
    cout<<"V=";  
    cin>>V;  
    if(V<1 || V>20)  
        cout<<"Wrong Input";  
    else if(V>=19)  
        cout<<"A";  
    else if(V>=16)  
        cout<<"B";  
    else if(V>=13)  
        cout<<"C";  
    else if(V>=10)  
        cout<<"D";
```



```

else
    cout<<"E ";
return 0;
}

```



Παράδειγμα 3.8

Ένα τυπογραφείο χρεώνει κλιμακωτά τους πελάτες του ως εξής:

- Τα πρώτα 100 βιβλία [1-100] προς 8 ευρώ το ένα.
- Τα επόμενα 400 βιβλία [101-500] προς 6.40 ευρώ το ένα.
- Τα υπόλοιπα βιβλία (περισσότερα από 500) προς 3.50 ευρώ το ένα.

Να γράψετε πρόγραμμα το οποίο να δέχεται το πλήθος των βιβλίων και να υπολογίζει και να τυπώνει το κόστος.

Υλοποίηση στη γλώσσα προγραμματισμού C++

```

#include<iostream>
using namespace std;
int main() {
    int books;
    float total;
    cout<<"Δώσε αριθμό βιβλίων";

```

```
cin>>books;
if (books<=100)
    total=books*8;
else if (books<=500)
    total=(books-100)*6.4+8*100;
else
    total=(books-500)*3.5+400*6.4+8*100;
cout<<"Κόστος="<<total;
return 0;
}
```

6. Περιπτωσιακή Δομή – Η εντολή switch

Με την εντολή switch μπορούμε να εκτελέσουμε μία ομάδα εντολών, αναλόγως της τιμής που θα έχει η μεταβλητή ελέγχου. Η διαφορά με το if είναι ότι δεν εξετάζει κάποια λογική έκφραση, αλλά τις τιμές που μπορεί να πάρει μία μεταβλητή ελέγχου.

Γενικά, η εντολή switch δεν χρησιμοποιείται ευρέως, παρά μόνο σε ειδικές περιπτώσεις.

Σύνταξη Εντολής

```
switch (μεταβλητή_ελέγχου) {
    case περίπτωση 1:
        εντολές;
        break;
    case περίπτωση 2:
        εντολές;
        break;
    // Μπορείτε να έχετε όσες περιπτώσεις θέλετε
    default:
        εντολές;
}
```

Επεξήγηση

- Η μεταβλητή ελέγχου πρέπει να είναι διακριτού τύπου (integer, char).
- Όταν βρεθεί περίπτωση που θα έχει την ίδια τιμή με την μεταβλητή ελέγχου, τότε εκτελούνται οι εντολές που βρίσκονται από κάτω, μέχρι να βρεθεί η εντολή break. Όταν εκτελεστεί η εντολή break, η switch τερματίζεται.
- Οι εντολές οι οποίες βρίσκονται μετά το default, θα εκτελεστούν, εφόσον δεν βρεθεί καμία περίπτωση που να είναι ίση με την τιμή της μεταβλητής ελέγχου.

Παράδειγμα 3.9

Η ποδοσφαιρική ομάδα «We are the champions» έχει ανακοινώσει ότι τα εισιτήρια διαρκείας θα χωριστούν στις πιο κάτω κατηγορίες με τις αντίστοιχες τιμές.

Κατηγορία	Τιμή
A	200
B	150
C	100
D	50

Να γράψετε πρόγραμμα το οποίο να δέχεται την κατηγορία και να τυπώνει την τιμή. Αν δεν δοθεί σωστή κατηγορία να εμφανίζεται μήνυμα λάθους.

Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>
using namespace std;
int main() {
    char katigoria;
    cout<<"Δώσε κατηγορία:";
    cin>>katigoria;
    switch (katigoria) {
        case 'A':
            cout<<200;
            break;
        case 'B':
            cout<<150;
            break;
        case 'C':
            cout<<100;
            break;
        case 'D':
            cout<<50;
            break;
        default:
            cout<<"Λάθος";
    }
    return 0;
}
```



7. Εισαγωγή στα strings

Μέχρι στιγμής στα προγράμματά μας χρησιμοποιούσαμε ως μεταβλητές μόνο αριθμούς ή έναν χαρακτήρα. Τι γίνεται, όμως, αν θέλουμε να αποθηκεύσουμε σε μία μεταβλητή μία σειρά από χαρακτήρες (π.χ. ένα όνομα ή μία συμβολοσειρά); Αυτό μπορεί να επιτευχθεί με τη δομή **string**. Στη δομή string θα αναφερθούμε με λεπτομέρεια και στη Γ' Λυκείου. Σε αυτό το σημείο, θα εξηγήσουμε τον τρόπο με τον οποίο μπορούμε να δηλώσουμε μία μεταβλητή, η οποία θα αποθηκεύει μία συμβολοσειρά.

Στο παράδειγμα που ακολουθεί, το πρόγραμμα δέχεται ένα όνομα. Αν το όνομα που θα δοθεί είναι «Theseas», τότε θα εμφανίζεται το μήνυμα «Welcome Mr. Theseas», διαφορετικά θα εμφανίζεται το μήνυμα «You are not Mr. Theseas».

Παράδειγμα 3.10

```
#include<iostream>
using namespace std;
int main() {
    string onoma;
    cout<<"Onoma=";
    cin>>onoma;
    if (onoma=="Theseas")
        cout<<"Welcome Mr.Theseas";
    else
        cout<<"You are not Mr. Theseas";
    return 0;
}
```

Με την εντολή **string onoma;** ορίζουμε ότι η μεταβλητή θα αποθηκεύει μία συμβολοσειρά. Όταν έχουμε συμβολοσειρές (string), χρησιμοποιούμε εισαγωγικά ("), ενώ όταν έχουμε απλούς χαρακτήρες (char) χρησιμοποιούμε μονά εισαγωγικά ('). Στο παράδειγμα που ακολουθεί δηλώνονται δύο μεταβλητές. Η μία τύπου string (st) και η άλλη τύπου char (ch). Για τις εντολές εκχώρησης χρησιμοποιούμε τα σύμβολα, τα οποία αναφέρονται πιο πάνω.

Παράδειγμα 3.11

```
#include<iostream>
using namespace std;
int main() {
    string st;
    char ch;
    st="abcd";
    ch='a';
    cout<<st;    //τυπώνει abcd
    cout<<ch;    //τυπώνει a
    return 0;
}
```

**Ασκήσεις Κεφαλαίου****Άσκηση 3.1**

Να γράψετε τα αποτελέσματα των πιο κάτω λογικών εκφράσεων, αν $x=5$, $y=-5$ και $z=10$:

- (1) $(x > z) \ \&\& \ (z == y)$
- (2) $(x - y == 0) \ || \ (y \% 2 == 0)$
- (3) $(z \% y == 0) \ \&\& \ (x \% y == 0)$
- (4) $(z == (x + y)) \ \&\& \ (z != 2/y) \ || \ (z > (y + x))$
- (5) $(z != (4 * x)) \ \&\& \ !(z < y)$
- (6) $!((z == 2 * x) \ || \ (y == 0))$
- (7) $((x - y) > 0) \ \&\& \ !((z - y) > 0)$

Άσκηση 3.2

Αν pA , pB και pC αντιστοιχούν στον πληθυσμό των χωρών A , B , C , να μετατρέψετε τις πιο κάτω λεκτικές προτάσεις σε λογικές εκφράσεις.

- (1) Η χώρα A έχει μεγαλύτερο ή ίσο πληθυσμό από το άθροισμα των χωρών B και C .
- (2) Η χώρα A δεν έχει τον μικρότερο πληθυσμό.
- (3) Η χώρα B είναι μικρότερη από την χώρα C ή είναι μεγαλύτερη από τη χώρα A .
- (4) Η χώρα C έχει τον μεγαλύτερο πληθυσμό.
- (5) Η χώρα B έχει τον 2^ο μεγαλύτερο πληθυσμό.

Άσκηση 3.3

Σας δίνονται τρεις ακέραιοι αριθμοί a , b , c . Να γράψετε τη λογική έκφραση για κάθε μία από τις παρακάτω περιπτώσεις:

- (1) Είναι ο αριθμός a ο μεγαλύτερος αριθμός από τους τρεις;
- (2) Είναι ο αριθμός b μέσα στο διάστημα a μέχρι c , συμπεριλαμβανομένων;
- (3) Είναι κάποιος από τους τρεις αριθμούς άρτιος (ζυγός);

Άσκηση 3.4

Αν A , B , C είναι ακέραιοι αριθμοί, να μετατρέψετε τις πιο κάτω προτάσεις σε λογικές εκφράσεις.

- (1) Ο αριθμός A είναι ο μεγαλύτερος από τους τρεις.
- (2) Ο αριθμός B δεν είναι ο μικρότερος από τους τρεις.
- (3) Όλοι οι αριθμοί είναι άρτιοι.
- (4) Υπάρχει τουλάχιστον ένας άρτιος αριθμός.
- (5) Το άθροισμα του A και του B είναι μεγαλύτερο από το άθροισμα του A και του C ή το γινόμενο όλων των αριθμών είναι μεγαλύτερο από το άθροισμά τους.

Άσκηση 3.5

Να μετατρέψετε τις πιο κάτω λεκτικές προτάσεις στις αντίστοιχες λογικές εκφράσεις στη γλώσσα προγραμματισμού C++:

- (1) Η μεταβλητή **grade** να είναι μεταξύ του 1 και του 100 συμπεριλαμβανομένων.
- (2) Η μεταβλητή **z** να είναι ζυγός αριθμός και να μην ισούται με 8.

Άσκηση 3.6

Αν $X = 10$, $Y = 5$ και $Z = -2$, να γράψετε το αποτέλεσμα των πιο κάτω λογικών εκφράσεων:

(1) $(X < Y) \ \&\& \ (Z > Y)$	(2) $(X == Y) \ \ (X != Z)$
(3) $(Y < 0) \ \ (Z > 0)$	(4) $(X > Y) \ \ (X > Z) \ \&\& \ (Y > X)$
(5) $!(X == (2 * Y)) \ \&\& \ !(Z == 2)$	(6) $!(X != Y) \ \ !(Y == Z)$
(7) $(X < 0) \ \&\& \ (Y > 0) \ \ (Z > 0)$	(8) $(Y == (X / 2)) \ \&\& \ (Z > 0)$

Άσκηση 3.7

Οι μεταβλητές πραγματικού τύπου x , y και z , έχουν τις ακόλουθες τιμές:

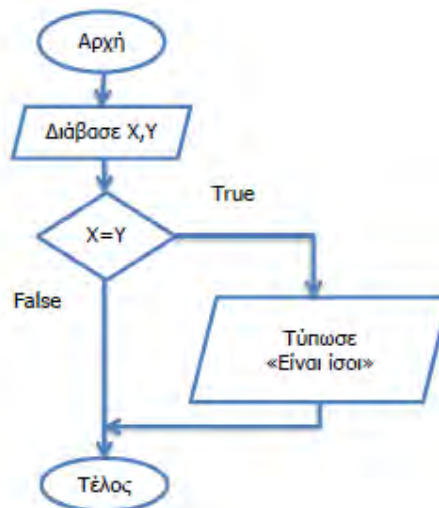
```
float x = 5.5, y = 6.0, z = 7.5;
```

Να γράψετε το αποτέλεσμα των πιο κάτω εντολών της C++:

- (1) `cout << ((1 > abs(x-z)) && (y > x));`
- (2) `cout << ((round(x) >= y) || (y != 6) || !(x > z) && (trunc(z) < y));`

Άσκηση 3.8

Να μετατρέψετε το πιο κάτω λογικό διάγραμμα σε πρόγραμμα της C++.

**Άσκηση 3.9**

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να τυπώνει το μήνυμα «One-digit number» σε περίπτωση που ο αριθμός είναι μονοψήφιος.

Παράδειγμα Εισόδου (πληκτρολόγιο) 9	Παράδειγμα Εξόδου (οθόνη) Δώσε έναν ακέραιο αριθμό: One-digit number
---	---

Άσκηση 3.10

Να γράψετε πρόγραμμα το οποίο να δέχεται ένα ακέραιο αριθμό και να τυπώνει το μήνυμα «Even Number», αν ο αριθμός είναι ζυγός και το μήνυμα «Multiple of seven», αν ο αριθμός είναι πολλαπλάσιο του επτά.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 14	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε έναν ακέραιο αριθμό: Even Number Multiple of seven
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 21	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε έναν ακέραιο αριθμό: Multiple of seven

Άσκηση 3.11

Ένας μαθητής θεωρείται ότι πέτυχε στο μάθημα της Πληροφορικής αν ο τελικός βαθμός του για το μάθημα είναι μεγαλύτερος ή ίσος με 10 και αν ο αριθμός των απουσιών του στο μάθημα είναι μικρότερος από 29. Να σχεδιάσετε λογικό διάγραμμα που να δέχεται τον τελικό βαθμό και τον αριθμό απουσιών ενός μαθητή και να τυπώνει ανάλογα το μήνυμα «επιτυχία» ή «αποτυχία».

Άσκηση 3.12

Να γράψετε πρόγραμμα το οποίο να δέχεται τα μήκη A και B των πλευρών ενός ορθογωνίου παραλληλογράμμου και στην περίπτωση που είναι και τα δύο μήκη θετικά, να υπολογίζει και να τυπώνει το εμβαδό (E) του ορθογωνίου παραλληλογράμμου.

Παράδειγμα Εισόδου (πληκτρολόγιο) 10 4	Παράδειγμα Εξόδου (οθόνη) Δώσε τα μήκη των πλευρών: Το εμβαδό είναι 40
--	---

Άσκηση 3.13

Να γράψετε πρόγραμμα το οποίο να δέχεται δύο ακέραιους αριθμούς X και Y και να τυπώνει τον αριθμό 1 αν το Y είναι πολλαπλάσιο του X, διαφορετικά να τυπώνει τον αριθμό 0.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 4 10	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε δύο ακέραιους αριθμούς: 0
--	---

Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 4 12	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε δύο ακέραιους αριθμούς: 1
--	---

Άσκηση 3.14

Ένα κατάστημα πώλησης ειδών ένδυσης προσφέρει εκπτώσεις ανάλογα με το ποσό αγορών. Αν ο πελάτης αγοράσει προϊόντα αξίας μικρότερης ή ίσης των €100, τότε προσφέρεται έκπτωση 20% πάνω στο ποσό αγοράς, διαφορετικά, προσφέρεται έκπτωση 45% υπολογιζόμενο στο ποσό αγοράς. Να σχεδιάσετε λογικό διάγραμμα το οποίο να δέχεται το ποσό των αγορών ενός πελάτη και να υπολογίζει και να τυπώνει την τελική τιμή που θα πληρώσει ο πελάτης.

Άσκηση 3.15

Ένα ξενοδοχείο εφαρμόζει για τις κρατήσεις του το ακόλουθο σχέδιο εκπτώσεων: για κάθε κράτηση δωματίου για περισσότερες από 3 διανυκτερεύσεις και τιμή ανά διανυκτέρευση €100 και άνω θα παρέχει στον πελάτη 30% έκπτωση στη συνολική τιμή, διαφορετικά θα παρέχει έκπτωση μόνο 10% (συνολική τιμή = διανυκτερεύσεις * τιμή ανά διανυκτέρευση).

- (1) Να σχεδιάσετε λογικό διάγραμμα, το οποίο να δέχεται τον αριθμό των διανυκτερεύσεων και την τιμή ανά διανυκτέρευση. Στη συνέχεια, να υπολογίζει και να τυπώνει την τελική τιμή που πρέπει να πληρώσει ο πελάτης (τελική τιμή = συνολική τιμή - έκπτωση).
- (2) Να μετατρέψετε το λογικό διάγραμμα του ερωτήματος (1) σε πρόγραμμα.

Παράδειγμα Εισόδου (πληκτρολόγιο) 4 100	Παράδειγμα Εξόδου (οθόνη) Δώσε αριθμό διανυκτερεύσεων: Δώσε τιμή: Τελική Τιμή= 280 ευρώ
--	---

Άσκηση 3.16

Τα ταχυδρομικά τέλη για αποστολή ενός μικροδέματος υπολογίζονται με βάση το βάρος του σε γραμμάρια. Συγκεκριμένα, για μικροδέματα με βάρος μεταξύ 20 και 500 γραμμαρίων, συμπεριλαμβανομένων, ο πελάτης χρεώνεται 0.017 σεντ ανά γραμμάριο.

Να γράψετε πρόγραμμα το οποίο να δέχεται το βάρος του μικροδέματος σε γραμμάρια και να υπολογίζει και να τυπώνει τη χρέωση για την αποστολή του μικροδέματος.

Σε περίπτωση που το βάρος είναι εκτός των ορίων που καθορίστηκαν πιο πάνω, να εμφανίζεται το μήνυμα «ΒΑΡΟΣ ΕΚΤΟΣ ΟΡΙΩΝ».

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 300	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε το βάρος: Η χρέωση είναι 5.10 ευρώ
---	--

Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 550	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε το βάρος: ΒΑΡΟΣ ΕΚΤΟΣ ΟΡΙΩΝ
---	---

Άσκηση 3.17

Δίνεται το πιο κάτω πρόγραμμα γραμμένο στη γλώσσα προγραμματισμού C++. Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του προγράμματος. Στη θέση του διαστήματος να χρησιμοποιήσετε το σύμβολο «□».

```
#include<iostream>
using namespace std;
int main(){
    int x, y;
    x = 2;
    y = 4;
    if (x<0)
        x=x+1;
    y=y+3;
    cout << x << " " << y;
    return 0;
}
```

Άσκηση 3.18

Δίνεται το πιο κάτω πρόγραμμα γραμμένο στη γλώσσα προγραμματισμού C++. Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του προγράμματος. Στη θέση του διαστήματος να χρησιμοποιήσετε το σύμβολο «□».

```
#include<iostream>
using namespace std;
int main(){
    int x, y;
    x = -2;
    y = 4;
    if (x<0)
        x=x+1;
    y=y+3;
    cout << x << " " << y;
    return 0;
}
```

Άσκηση 3.19

Δίνεται το πιο κάτω πρόγραμμα γραμμένο στη γλώσσα προγραμματισμού C++. Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του προγράμματος. Στη θέση του διαστήματος να χρησιμοποιήσετε το σύμβολο «□».

```
#include<iostream>
using namespace std;
int main() {
    int a,b,c;
    a = 1;
    b = 2;
    c = 3;

    if (c == a+b)
        cout << "Line 1" << endl;
    if (b == c-a)
        if (a != c-b)
            cout << "Line 2" << endl;
    if ((b == a+1) && (c != b+2))
        cout << "Line 3";
    return 0;
}
```

Άσκηση 3.20

Δίνεται το πιο κάτω πρόγραμμα γραμμένο στη γλώσσα προγραμματισμού C++. Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του προγράμματος για αρχικές τιμές:

α) a=80, b=40, c=60 και β) a=90, b=60, c=20

Στη θέση του διαστήματος να χρησιμοποιήσετε το σύμβολο «□».

```
#include<iostream>
using namespace std;
int main() {
    int a,b,c,total;
    cin>>a>>b>>c;
    total=a+b+c;
    if((a==90 || b==90 || c==90) && total==180)
        cout<<"Right Triangle";
    else
        cout<<"Not Right Triangle";
    return 0;
}
```

Άσκηση 3.21

Να σχεδιάσετε λογικό διάγραμμα και να γράψετε πρόγραμμα το οποίο να δέχεται δύο μη μηδενικούς ακέραιους αριθμούς και να τυπώνει τον αριθμό 1, αν είναι ομόσημοι (έχουν το ίδιο πρόσημο), διαφορετικά να τυπώνει τον αριθμό 0.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 3 100	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε δύο αριθμούς: 1
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) -3 100	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε δύο αριθμούς: 0

Άσκηση 3.22

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν τριψήφιο αριθμό και να τυπώνει το μήνυμα «Έγκυρος αριθμός», αν δεν υπάρχει επανάληψη ψηφίου ανάμεσα στα ψηφία του αριθμού. Για παράδειγμα, ο αριθμός 456 είναι έγκυρος, ενώ ο αριθμός 454 όχι.

Παράδειγμα Εισόδου (πληκτρολόγιο) 456	Παράδειγμα Εξόδου (οθόνη) Δώσε έναν αριθμό: Έγκυρος αριθμός
--	--

Άσκηση 3.23

Ένας μαθητής μετά το τέλος του σχολικού έτους παραμένει στάσιμος, αν οι αδικαιολόγητες απουσίες είναι περισσότερες από 45 ή οι συνολικές του απουσίες είναι περισσότερες από 180, διαφορετικά μπορεί να συμμετάσχει στις τελικές εξετάσεις. Να γράψετε πρόγραμμα το οποίο να δέχεται τις αδικαιολόγητες απουσίες (Α) και τις συνολικές απουσίες (Τ) ενός μαθητή για το τρέχον σχολικό έτος και να τυπώνει το ανάλογο μήνυμα.

Παράδειγμα Εισόδου (πληκτρολόγιο) 46 160	Παράδειγμα Εξόδου (οθόνη) Δώσε αδικαιολόγητες και συνολικές απουσίες: Παραμένει στάσιμος
---	---

Άσκηση 3.24

Ένας ωρομίσθιος υπάλληλος για κάθε ώρα που δουλεύει πληρώνεται με 15 ευρώ την ώρα. Αν το ποσό που θα εισπράξει ξεπερνά τα 1000 ευρώ, θα του επιβληθεί φορολογία 13%. Να γράψετε πρόγραμμα το οποίο να δέχεται τις ώρες που δούλεψε ο υπάλληλος μέσα σε μία εβδομάδα και να τυπώνει τον μισθό του.

Παράδειγμα Εισόδου (πληκτρολόγιο) 70	Παράδειγμα Εξόδου (οθόνη) Δώσε τις ώρες εργασίας: Μισθός=892.50 ευρώ
---	---

Άσκηση 3.25

Η μετεωρολογική υπηρεσία εκδίδει απαγορευτικό για τα πλοία, στις πιο κάτω περιπτώσεις:

- (1) όταν η ταχύτητα του ανέμου είναι μεγαλύτερη από 6 μποφόρ
- (2) η ταχύτητα του ανέμου είναι μεταξύ 4 και 6 μποφόρ συμπεριλαμβανομένων, και η θερμοκρασία είναι μικρότερη από 0 βαθμούς

Να γράψετε πρόγραμμα το οποίο να δέχεται τη θερμοκρασία και την ένταση του ανέμου και σε περίπτωση που υπάρχει λόγος, να τυπώνει το μήνυμα «Extreme weather».

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
5 -3	Δώσε θερμοκρασία: Δώσε ένταση ανέμου: Extreme weather

**Άσκηση 3.26**

Ο Θησέας δουλεύει σε μία εταιρεία κατασκευής λογισμικού. Για κάθε ώρα που δουλεύει, μέχρι τις 40 ώρες την εβδομάδα, πληρώνεται 15 ευρώ ανά ώρα. Υπάρχουν, όμως, εβδομάδες κατά τις οποίες πρέπει να δουλέψει υπερωρίες (περισσότερες από 40 ώρες), για τις οποίες πληρώνεται 20 ευρώ την ώρα. Να γράψετε πρόγραμμα το οποίο να δέχεται τις ώρες που δούλεψε ο Θησέας σε μία εβδομάδα και να τυπώνει τον μισθό του.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο)	Παράδειγμα Εξόδου 1 (οθόνη)
35	Δώσε ώρες εργασίας: Μισθός= 525 ευρώ
Παράδειγμα Εισόδου 2 (πληκτρολόγιο)	Παράδειγμα Εξόδου 2 (οθόνη)
45	Δώσε ώρες εργασίας: Μισθός= 700 ευρώ

Άσκηση 3.27

Ένα μουσείο πωλεί τα εισιτήριά του προς 10 ευρώ. Το μουσείο κάνει έκπτωση 2 ευρώ για κάθε εισιτήριο κάθε Σάββατο και Κυριακή. Να γράψετε πρόγραμμα το οποίο να δέχεται το πλήθος των ατόμων που θέλουν να αγοράσουν εισιτήριο, καθώς και την ημέρα της επίσκεψης (1-Δευτέρα, 2-Τρίτη, 3-Τετάρτη, 4-Πέμπτη, 5-Παρασκευή, 6-Σάββατο, 7-Κυριακή). Το πρόγραμμα να υπολογίζει και να τυπώνει το ποσό που πρέπει να πληρωθεί.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
5 6	Δώσε αριθμό ατόμων: Δώσε μέρα επίσκεψης (1-7): Ποσό=40.00 ευρώ

Άσκηση 3.28

Για σκοπούς ασφαλείας, σε ένα στρατόπεδο χρησιμοποιούν την εξής μέθοδο: το σύνθημα και το παρασύνθημα που χρησιμοποιούν οι σκοποί είναι τετραψήφιοι ακέραιοι αριθμοί. Αν το άθροισμα των ψηφίων του συνθήματος είναι το ίδιο με το άθροισμα των ψηφίων του παρασυνθήματος, τότε επιτρέπεται η είσοδος στο στρατόπεδο. Να γράψετε πρόγραμμα το οποίο να δέχεται το σύνθημα και το παρασύνθημα και να τυπώνει τη λέξη «GO» αν υπάρχει ταύτιση αλλιώς να τυπώνει τη λέξη «STOP».

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 1234 4321	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε σύνθημα: Δώσε παρασύνθημα: GO
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 1234 4444	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε σύνθημα: Δώσε παρασύνθημα: STOP

Άσκηση 3.29

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο τριψήφιο αριθμό και να τυπώνει τον αριθμό 1, στην περίπτωση που ο αριθμός είναι παλίνδρομος, διαφορετικά να τυπώνει τον αριθμό 0. Παλίνδρομος θεωρείται ένας αριθμός, ο οποίος μπορεί να διαβαστεί ως ο ίδιος, τόσο από αριστερά όσο και από δεξιά.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 121	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε αριθμό: 1
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 122	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε αριθμό: 0

Άσκηση 3.30

Μία προπονήτρια ενόργανης γυμναστικής θα διδάξει φέτος σε δύο ομάδες κοριτσιών ανάλογα με την ηλικία. Η πρώτη ομάδα θα είναι ηλικίες από 4 μέχρι 8 και η δεύτερη από 10 μέχρι 14. Να σχεδιάσετε λογικό διάγραμμα το οποίο να δέχεται το φύλο (M ή F) και την ηλικία ενός παιδιού και να τυπώνει μόνο ένα από τα μηνύματα «Group 1», «Group 2» ή «No registration», αντίστοιχα.

Άσκηση 3.31

Να σχεδιάσετε λογικό διάγραμμα το οποίο να δέχεται τον βαθμό ενός μαθητή για ένα μάθημα. Αν ο βαθμός είναι μεγαλύτερος από 20, να εμφανίζεται το μήνυμα «Too big». Αν ο

βαθμός είναι μικρότερος από 1, τότε να τυπώνει το μήνυμα «Too small», αλλιώς να τυπώνει το μήνυμα «Grade is valid».

Άσκηση 3.32

Δύο μαθητές στο μάθημα της χημείας μελετούν μία χημική αντίδραση και σημειώνουν το χρώμα του διαλύματος, το οποίο δημιουργείται μετά την πάροδο συγκεκριμένου χρόνου. Ο πιο κάτω πίνακας παρουσιάζει τα αποτελέσματα που έχουν καταγράψει:

Χρόνος από την ανάμειξη	Χρώμα	Κωδικός χρώματος
3-5	Κίτρινο	1
6-8	Πορτοκαλί	2
11-13	Κόκκινο	3

Σε οποιαδήποτε άλλη χρονική στιγμή το χρώμα είναι απροσδιόριστο. Να γράψετε πρόγραμμα το οποίο να δέχεται τα λεπτά που έχουν περάσει από την ανάμειξη του διαλύματος και να τυπώνει τον κωδικό του κάθε χρώματος, ή τον αριθμό -1 αν το χρώμα είναι απροσδιόριστο.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 7	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε λεπτά: 2
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 1	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε λεπτά: -1

Άσκηση 3.33

Να μετατρέψετε την πιο κάτω ένθετη δομή διακλάδωσης σε περιπτωσιακή δομή (switch).

```
#include<iostream>
using namespace std;

int main(){
    int epid;
    cin >> epid;
    if (epid>=1 && epid<=3)
        cout << "ΜΕΤΡΙΑ" << endl;
    else if (epid>=4 && epid<=7)
        cout << "ΚΑΛΗ" << endl;
    else if (epid>=8 && epid<=10)
        cout << "ΑΡΙΣΤΗ" << endl;
    else
        cout << "Λανθασμένη τιμή" << endl;
    return 0;}
```


Άσκηση 3.34

Να μετατρέψετε την πιο κάτω περιπτωσιακή δομή (switch) σε ένθετη δομή διακλάδωσης.

```
#include<iostream>
#include<cmath>
using namespace std;

int main(){
    int num;
    cin >> num;
    switch (num){
        case 2:
        case 5:    cout << pow(num,2) << endl;
                  break;

        case 8:
        case 10:   cout << pow(num,3) << endl;
                  break;

        case 11:
        case 14:   cout << num << endl;
                  break;

        default:   cout << "Λανθασμένη τιμή" << endl;

    }
    return 0;}
```

Άσκηση 3.35

Σας δίνεται ο πιο κάτω πίνακας που περιλαμβάνει τον κωδικό και την τιμή ενός προϊόντος.

Κωδικός Προϊόντος	Τιμή
1	€ 4.00
2, 3, 4	€ 4.50
5, 10, 15	€ 5.00
7, 9	€ 2.00
12	€ 1.50

Να γράψετε πρόγραμμα, το οποίο να δέχεται τον κωδικό (code) και την ποσότητα ενός προϊόντος (qty) και να υπολογίζει και να τυπώνει το συνολικό ποσό (total), με ακρίβεια δύο δεκαδικών ψηφίων.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
2 10	Δώσε κωδικό: Δώσε ποσότητα: Συνολικό ποσό= 45.00

Άσκηση 3.36

Να γράψετε πρόγραμμα το οποίο να δέχεται τα τέρματα που πέτυχε η κάθε ομάδα σε έναν αγώνα ποδοσφαίρου και να τυπώνει το αποτέλεσμα (1-Νίκη γηπεδούχου, Χ-Ισοπαλία, 2-Νίκη φιλοξενούμενου).

Παράδειγμα Εισόδου 1 (πληκτρολόγιο)	Παράδειγμα Εξόδου 1 (οθόνη)
2 2	Δώσε τέρματα: X
Παράδειγμα Εισόδου 2 (πληκτρολόγιο)	Παράδειγμα Εξόδου 2 (οθόνη)
2 1	Δώσε τέρματα: 1
Παράδειγμα Εισόδου 3 (πληκτρολόγιο)	Παράδειγμα Εξόδου 3 (οθόνη)
1 2	Δώσε τέρματα: 2

Άσκηση 3.37

Το ταχυδρομείο χρεώνει τα δέματα ανάλογα με το βάρος τους. Αν ένα δέμα ζυγίζει μέχρι και 560 γραμμάρια, τότε χρεώνεται προς 1 σεντ το γραμμάριο, διαφορετικά για κάθε γραμμάριο που ξεπερνά τα 560 ο πελάτης το χρεώνεται προς 2 σεντ. Για παράδειγμα, αν το δέμα ζυγίζει 100 γραμμάρια, ο πελάτης θα πληρώσει 100 σεντ, ενώ αν ζυγίζει 1000 γραμμάρια, θα πληρώσει $560 + (1000 - 560) * 2 = 1440$ σεντ. Να σχεδιάσετε λογικό διάγραμμα το οποίο να δέχεται το βάρος του δέματος (B) και να υπολογίζει και να τυπώνει το ποσό που πρέπει να πληρώσει ο πελάτης (A).

Άσκηση 3.38

Μία εταιρεία παρέχει υπηρεσίες ήχου και εικόνας υψηλής πιστότητας. Μία από τις υπηρεσίες που παρέχει είναι η φωτογράφιση προϊόντων και χρεώνει ως εξής:

Ποσότητα	Τιμή ανά Τεμάχιο
1 – 25	13.49 €
26 – 50	12.74 €
Πάνω από 50	11.99 €

Να γράψετε πρόγραμμα το οποίο να δέχεται την ποσότητα (P) και να υπολογίζει και να τυπώνει τη συνολική τιμή (T).

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
20	Δώσε ποσότητα: Συνολική τιμή: 269.80 ευρώ



Άσκηση 3.39

Σας δίνεται ο πιο κάτω πίνακας που περιλαμβάνει τον κωδικό και την τιμή ενός προϊόντος.

Κωδικός Προϊόντος	Τιμή
1	€ 4.00
2	€ 4.50
3	€ 5.00
4	€ 2.00
5	€ 1.50

Να γράψετε πρόγραμμα το οποίο να δέχεται τον κωδικό (code) και την ποσότητα ενός προϊόντος (qty) και να υπολογίζει και να τυπώνει το συνολικό ποσό (total), με ακρίβεια δύο δεκαδικών ψηφίων.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
3 10	Δώσε κωδικό: Δώσε ποσότητα: Συνολικό ποσό:50.00 ευρώ

Άσκηση 3.40

Οι υπάλληλοι μίας εταιρείας, συμφώνησαν για τον μήνα Δεκέμβριο, να κρατηθούν από το μισθό τους δύο ποσά, ένα για την ενίσχυση του ερυθρού σταυρού και ένα για την ενίσχυση του κοινωνικού παντοπωλείου. Ο υπολογισμός του ποσού των εισφορών εξαρτάται από τον αρχικό μισθό του κάθε υπαλλήλου και υπολογίζεται με βάση τα παρακάτω όρια μισθών:

Μισθός	Εισφορά 1	Εισφορά 2
Μέχρι 750 ευρώ	5%	4%
751-1000 ευρώ	7.5%	6%
1000-1500 ευρώ	9.5%	8%
Μεγαλύτερο από 1500 ευρώ	12%	11%

Να γράψετε πρόγραμμα το οποίο να δέχεται τον μισθό του υπαλλήλου και στη συνέχεια να υπολογίζει και να τυπώνει το ποσό των δύο εισφορών και το καθαρό ποσό που θα πάρει ο υπάλληλος.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
1200	Δώσε μισθό: Εισφορά 1:90.00 ευρώ Εισφορά 2:72.00 ευρώ Καθαρός Μισθός:1038.00 ευρώ

Άσκηση 3.41

Σε ένα σχολείο της Λευκωσίας διεξάγεται δημοψήφισμα σχετικά με το αν συμφωνούν οι μαθητές/τριες με τα νέα κόμιστρα των λεωφορείων του ΟΣΕΛ. Οι μαθητές/τριες πρέπει να απαντήσουν στην ερώτηση «Συμφωνείτε με τα αυξημένα κόμιστρα (Ναι/Όχι)». Για να έχει ισχύ το δημοψήφισμα, πρέπει να υπάρξει συμμετοχή τουλάχιστον 80% των μαθητών/τριών. Για να γίνουν αποδεκτά τα νέα κόμιστρα, πρέπει οι απαντήσεις των μαθητών/τριών που απάντησαν «Ναι» να ξεπερνούν το 50% των συμμετεχόντων. Να γράψετε πρόγραμμα το οποίο να δέχεται το πλήθος των μαθητών/τριών του σχολείου, τον αριθμό των μαθητών/τριών που απάντησαν «Ναι» και τον αριθμό των μαθητών/τριών που απάντησαν «Όχι» και να τυπώνει τον αριθμό 1, αν εγκρίθηκε η πρόταση ή τον αριθμό -1, αν δεν εγκρίθηκε.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
520 395 120	Δώσε τον αριθμό μαθητών/τριων του σχολείου: Δώσε τον αριθμό των μαθητών/τριων που απάντησαν ΝΑΙ: Δώσε τον αριθμό των μαθητών/τριων που απάντησαν ΟΧΙ: 1

**Άσκηση 3.42**

Όπως είναι γνωστό, οι αριθμοί των σταθερών τηλεφώνων σε μία επαρχία έχουν κοινά τα πρώτα 2 ψηφία. Στον πιο κάτω πίνακα φαίνονται τα πρώτα 2 ψηφία των τηλεφώνων της κάθε επαρχίας.

Δύο πρώτα ψηφία	Επαρχία
22	Λευκωσία
23	Αμμόχωστος
24	Λάρνακα
25	Λεμεσός
26	Πάφος

Επιπλέον, αν το 3^ο και 4^ο ψηφίο ενός αριθμού στη Λευκωσία σχηματίζουν το 80, τότε σημαίνει ότι πρόκειται για αριθμό κυβερνητικής υπηρεσίας. Να γράψετε πρόγραμμα το οποίο να δέχεται έναν οκταψήφιο αριθμό και να τυπώνει την επαρχία. Επίσης, αν είναι κυβερνητική υπηρεσία στη Λευκωσία, να τυπώνει τη φράση «Κυβερνητική Υπηρεσία».

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
22806565	Δώσε έναν οκταψήφιο αριθμό: Λευκωσία Κυβερνητική Υπηρεσία

Άσκηση 3.43

Ένα τυπογραφείο χρεώνει κλιμακωτά τους πελάτες του ως εξής:

- Για εκτύπωση των πρώτων 100 σελίδων [1-100] προς 0.28 ευρώ τη μία.
- Για εκτύπωση των επόμενων 400 σελίδων [101-500] προς 0.24 ευρώ τη μία.
- Για εκτύπωση των υπόλοιπων σελίδων (περισσότερων από 500 σελίδων) προς 1.50 ευρώ τη μία.

Να γράψετε πρόγραμμα το οποίο να δέχεται τον αριθμό των σελίδων και να υπολογίζει και να τυπώνει το ποσό χρέωσης.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
400	Δώσε αριθμό σελίδων: Ποσό χρέωσης= 104.80 ευρώ

Άσκηση 3.44

Οι υπάλληλοι μίας εταιρείας, στο τέλος κάθε μήνα φορολογούνται με βάση τον μηνιαίο ακάθαρτο τους μισθό. Ο φόρος εισοδήματος υπολογίζεται κλιμακωτά με βάση τα παρακάτω όρια μισθών:

Μισθός	Ποσοστό Φόρου Εισοδήματος
1 – 1000 ευρώ	0%
10001-1800 ευρώ	10%
1800-2500 ευρώ	20%
2501-3500 ευρώ	30%
Μεγαλύτερο από 3500 ευρώ	40%

Να γράψετε πρόγραμμα το οποίο να δέχεται τον μισθό του υπαλλήλου και στη συνέχεια να υπολογίζει και να τυπώνει το ποσό του φόρου εισοδήματος και το καθαρό ποσό που θα πάρει ο υπάλληλος.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
1200	Δώσε μισθό: Φόρος Εισοδήματος: 20.00 ευρώ Καθαρός Μισθός: 1180.00 ευρώ

Επεξήγηση: Στο πιο πάνω παράδειγμα, για τα πρώτα 1000 ευρώ ο φόρος είναι 0 ευρώ και για τα υπόλοιπα 200 (1200-1000) ο φόρος είναι 20 (200*10%) ευρώ.

Άσκηση 3.45

Ένας πωλητής της αυτοκινητοβιομηχανίας XYZ αμείβεται μηνιαίως με ένα ποσό των 750 €, ανεξάρτητα από τον αριθμό των αυτοκινήτων που πούλησε. Σε αυτόν το μισθό προστίθεται κι ένα επιπρόσθετο ποσό ανάλογα με τον αριθμό αυτοκινήτων που πούλησε στο μήνα ως εξής :

Αριθμός Αυτοκινήτων	Επιπρόσθετο ποσό(για κάθε αυτοκίνητο)
1-9	0 €
10-20	15 €
21-30	30 €
30 και πάνω	60 €

Να γράψετε πρόγραμμα, το οποίο θα δέχεται τον αριθμό των αυτοκινήτων που πώλησε ο πωλητής και να υπολογίζει και να τυπώνει το επιπρόσθετο ποσό και τον συνολικό μισθό για τον μήνα αυτό.

Σημείωση: Το επιπρόσθετο ποσό να υπολογίζεται κλιμακωτά.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
25	Δώσε αριθμό αυτοκινήτων: Επιπρόσθετο ποσό: 315 ευρώ Συνολικός μισθός: 1065 ευρώ

Άσκηση 3.46

Στον διαγωνισμό για θέσεις καθηγητών στην εκπαίδευση, οι υποψήφιοι καθηγητές εξετάζονται σε δύο αντικείμενα, στο μάθημα της ειδικότητας τους και σε μάθημα που σχετίζεται με παιδαγωγικά θέματα. Σε κάθε μάθημα παίρνουν έναν βαθμό από 1-100. Ο συνολικός βαθμός είναι το άθροισμα των δύο επιμέρους βαθμών. Για να είναι επιτυχής η εξέτασή τους, θα πρέπει να συγκεντρώσουν τουλάχιστον 120 βαθμούς συνολικά, αλλά και η διαφορά των βαθμών στα δύο επιμέρους μαθήματα να μην ξεπερνά το 20. Να γράψετε πρόγραμμα το οποίο να τυπώνει το μήνυμα "ΕΠΙΤΥΧΗΣ" αν η εξέταση ενός υποψηφίου ήταν επιτυχής.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
75 90	Δώσε τον βαθμό της ειδικότητας: Δώσε τον βαθμό των παιδαγωγικών: ΕΠΙΤΥΧΗΣ

Άσκηση 3.47

Μία εταιρεία εμπορίας αυτοκινήτων πρότεινε σε έναν υποψήφιο αγοραστή τα παρακάτω προγράμματα χρηματοδότησης :

1^ο πρόγραμμα: Προκαταβολή του 40% της αρχικής αξίας του αυτοκινήτου και 12 δόσεις ίσες με το 1/18 της αρχικής αξίας του αυτοκινήτου.

2^ο πρόγραμμα: Προκαταβολή του 50% της αρχικής αξίας του αυτοκινήτου και 24 δόσεις ίσες με το 1/40 της αρχικής αξίας του αυτοκινήτου.

Να γράψετε πρόγραμμα το οποίο:

- (1) να δέχεται την αρχική αξία του αυτοκινήτου και

- (2) να τυπώνει το μήνυμα «Program 1» ή «Program 2», ανάλογα με το ποιο από αυτά είναι πιο φθηνό για τον υποψήφιο αγοραστή (δηλαδή αντιστοιχεί στο μικρότερο συνολικό ποσό πληρωμής).

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
22000	Δώσε την αξία του αυτοκινήτου: Program 1



Άσκηση 3.48

Να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει τον βαθμό ενός μαθητή σε ένα εξεταζόμενο μάθημα στις Εισαγωγικές Εξετάσεις. Η διαδικασία υπολογισμού είναι η εξής:

- Το γραπτό το βαθμολογούν δύο βαθμολογητές. Αν η διαφορά των δύο βαθμών είναι μικρότερη ή ίση από 15 μόρια, τότε ο τελικός βαθμός του μαθητή είναι το άθροισμα των δύο βαθμών.
- Αν η διαφορά των δύο βαθμών είναι μεγαλύτερη από 15 μόρια, τότε το γραπτό το βαθμολογεί και τρίτος βαθμολογητής.

Στην περίπτωση ύπαρξης τρίτου βαθμολογητή:

- Αν ο τρίτος βαθμός είναι μικρότερος από τους δύο πρώτους βαθμούς τότε αγνοείται και ο τελικός βαθμός του μαθητή θα είναι το άθροισμα των δύο πρώτων βαθμών.
- Αν ο τρίτος βαθμός είναι ανάμεσα στους δύο πρώτους βαθμούς τότε ο τελικός βαθμός του μαθητή θα είναι το άθροισμα του τρίτου βαθμού και του πιο κοντινού του βαθμού, από τους δύο πρώτους.
- Αν ο τρίτος βαθμός είναι μεγαλύτερος και από τους δύο πρώτους βαθμούς, τότε ο τελικός βαθμός του μαθητή θα είναι το άθροισμα του τρίτου βαθμού και του μεγαλύτερου βαθμού, από τους δύο πρώτους.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο)	Παράδειγμα Εξόδου 1 (οθόνη)
85 92	Δώσε τον βαθμό του 1ου βαθμολογητή: Δώσε τον βαθμό του 2ου βαθμολογητή: Βαθμολογία= 177
Παράδειγμα Εισόδου 2 (πληκτρολόγιο)	Παράδειγμα Εξόδου 2 (οθόνη)
85 92 90	Δώσε τον βαθμό του 1ου βαθμολογητή: Δώσε τον βαθμό του 2ου βαθμολογητή: Δώσε τον βαθμό του 3ου βαθμολογητή: Βαθμολογία= 182



Άσκηση 3.49

Ο τελικός βαθμός ενός μαθητή σε ένα μάθημα υπολογίζεται με βάση την προφορική και τη γραπτή του βαθμολογία, με την ακόλουθη διαδικασία: Αν η διαφορά των δύο βαθμών είναι μεγαλύτερη από πέντε μονάδες, τότε ο προφορικός βαθμός προσαρμόζεται (δηλαδή αυξάνεται ή μειώνεται), έτσι ώστε η αντίστοιχη διαφορά να μειωθεί στις τρεις μονάδες,

διαφορετικά ο προφορικός βαθμός παραμένει αμετάβλητος. Ο τελικός βαθμός είναι ο μέσος όρος του προφορικού και του γραπτού βαθμού.

Παράδειγμα προσαρμογής του προφορικού βαθμού: Αν ο γραπτός βαθμός είναι 18 και ο προφορικός 11, τότε ο προφορικός γίνεται 15, ενώ αν ο γραπτός βαθμός είναι 10 και ο προφορικός 19, τότε ο προφορικός γίνεται 13.

Να γράψετε πρόγραμμα το οποίο να δέχεται 2 ακέραιους αριθμούς X, Y ($0 \leq X, Y \leq 20$), οι οποίοι αντιστοιχούν στον γραπτό και στον προφορικό βαθμό και να υπολογίζει και να τυπώνει την τελική βαθμολογία με ακρίβεια 2 δεκαδικών ψηφίων.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 18 11	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε τον προφορικό βαθμό: Δώσε τον γραπτό βαθμό: Τελική Βαθμολογία=16.50
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 10 12	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε τον προφορικό βαθμό: Δώσε τον γραπτό βαθμό: Τελική Βαθμολογία=11.00

Άσκηση 3.50

Να γράψετε πρόγραμμα το οποίο να δέχεται τρεις ακέραιους αριθμούς και να υπολογίζει και να τυπώνει τον μεγαλύτερο από τους τρεις.

Παράδειγμα Εισόδου (πληκτρολόγιο) 14 5 19	Παράδειγμα Εξόδου (οθόνη) Δώσε 3 ακέραιους αριθμούς: Ο μεγαλύτερος είναι: 19
---	---

Άσκηση 3.51

Να γράψετε πρόγραμμα το οποίο να δέχεται τρεις ακέραιους αριθμούς και να τους τυπώνει ταξινομημένους σε φθίνουσα σειρά.

Παράδειγμα Εισόδου (πληκτρολόγιο) 14 5 19	Παράδειγμα Εξόδου (οθόνη) Δώσε 3 ακέραιους αριθμούς: Οι ταξινομημένοι αριθμοί είναι: 19 14 5
---	--



Άσκηση 3.52

Να γράψετε πρόγραμμα το οποίο να δέχεται τρεις ακέραιους αριθμούς και να υπολογίζει και να τυπώνει τον μεσαίο σε μέγεθος.

Παράδειγμα Εισόδου πληκτρολόγιο) 14 5 19	Παράδειγμα Εξόδου (οθόνη) Δώσε 3 ακέραιους αριθμούς: Ο μεσαίος είναι:14
--	--

Άσκηση 3.53

Οι λύσεις της εξίσωσης δευτέρου βαθμού $ax^2 + bx + c = 0$ είναι $x_{1,2} = \frac{-b \pm \sqrt{d}}{2a}$, όπου $d = b^2 - 4ac$. Να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει τις λύσεις της πιο πάνω εξίσωσης, λαμβάνοντας υπόψη και τα εξής:

- Αν $a, b, c = 0$, τότε η εξίσωση είναι «Αόριστη».
- Αν $a, b = 0$ και $c \neq 0$, τότε η εξίσωση είναι «Αδύνατη».
- Αν η διακρίνουσα (d) είναι μικρότερη από το μηδέν, τότε έχουμε «Μη πραγματικές λύσεις».

Παράδειγμα Εισόδου 1 (πληκτρολόγιο)	Παράδειγμα Εξόδου 1 (οθόνη)
3 4 5	Δώσε 3 συντελεστές: Μη πραγματικές λύσεις

Άσκηση 3.54

Να γράψετε πρόγραμμα το οποίο να δέχεται τρεις ακέραιους αριθμούς και να τους τυπώνει σε αύξουσα σειρά.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
7 25 3	Δώσε 3 ακέραιους αριθμούς: Οι αριθμοί σε αύξουσα σειρά: 3 7 25

Άσκηση 3.55

Ο Μάκης και ο Τάκης παίζουν ένα παιχνίδι. Ο Μάκης γράφει στο τετράδιό του δύο διψήφιους ακέραιους αριθμούς. Ο Τάκης γράφει στο δικό του τετράδιο δύο μονοψήφιους ακέραιους αριθμούς.

Αν ο Τάκης έχει βρει τουλάχιστον ένα ψηφίο, με το οποίο να ξεκινά ένας διψήφιος, ή τουλάχιστον ένα ψηφίο που να καταλήγει ένας διψήφιος, τότε κερδίζει το παιχνίδι. Να γράψετε πρόγραμμα το οποίο να δέχεται τους δύο διψήφιους αριθμούς του Μάκη και τους δύο μονοψήφιους αριθμούς του Τάκη και να τυπώνει το όνομα αυτού που έχει κερδίσει το παιχνίδι. Για παράδειγμα, αν ο Μάκης γράψει τους αριθμούς 13, 46 και ο Τάκης τους αριθμούς 2 και 6, τότε έχει κερδίσει ο Τάκης. Αν ο Μάκης γράψει τους αριθμούς 10, 27 και ο Τάκης τους αριθμούς 5 και 8, τότε έχει κερδίσει ο Μάκης.

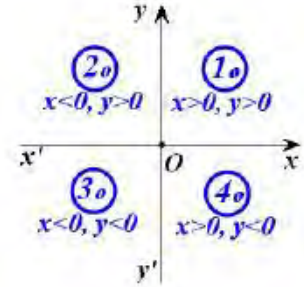
Παράδειγμα Εισόδου 1 (πληκτρολόγιο)	Παράδειγμα Εξόδου 1 (οθόνη)
13 46 2 6	Δώσε τους 2 αριθμούς του Μάκη: Δώσε τους 2 αριθμούς του Τάκη: Τάκης

Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 10 27 5 8	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε τους 2 αριθμούς του Μάκη: Δώσε τους 2 αριθμούς του Τάκη: Μάκης
--	---

Άσκηση 3.56

Να γράψετε πρόγραμμα το οποίο να δέχεται τις συντεταγμένες ενός σημείου με τη μορφή 2 δεκαδικών αριθμών (X,Y) σε ένα καρτεσιανό σύστημα συντεταγμένων. Στη συνέχεια να τυπώνει:

- Q1, αν το σημείο βρίσκεται στο 1^ο τεταρτημόριο
- Q2, αν το σημείο βρίσκεται στο 2^ο τεταρτημόριο
- Q3, αν το σημείο βρίσκεται στο 3^ο τεταρτημόριο
- Q4, αν το σημείο βρίσκεται στο 4^ο τεταρτημόριο
- X, αν βρίσκεται πάνω στον άξονα X
- Y, αν βρίσκεται πάνω στον άξονα Y
- Start, αν βρίσκεται στην αρχή των αξόνων



Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 4.5 -2.5	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε 2 αριθμούς: Q4
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 0.0 1.0	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε 2 αριθμούς: Y
Παράδειγμα Εισόδου 3 (πληκτρολόγιο) 0.0 0.0	Παράδειγμα Εξόδου 3 (οθόνη) Δώσε 2 αριθμούς: Start

Άσκηση 3.57

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν αριθμητικό τελεστή (+, -, *, /) και δύο δεκαδικούς αριθμούς και τυπώνει να το αποτέλεσμα της πράξης.

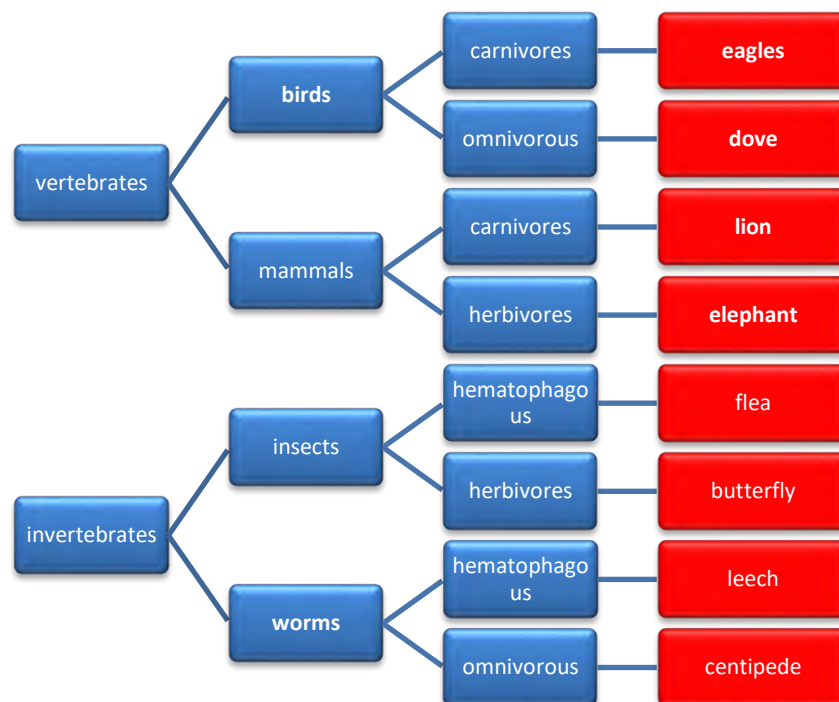
Τα αποτελέσματα να εμφανίζονται με ακρίβεια 2 δεκαδικών ψηφίων.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) + 7.1 10.0	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε τελεστή: Δώσε 2 αριθμούς: Αποτέλεσμα =17.10
---	--

Παράδειγμα Εισόδου 2 (πληκτρολόγιο) * 5.2 2.5	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε τελεστή: Δώσε 2 αριθμούς: Αποτέλεσμα =13.00
--	--

Άσκηση 3.58

Δίνεται η πιο κάτω ομαδοποίηση ζώων:



Να γράψετε πρόγραμμα το οποίο να δέχεται την περιγραφή του ζώου και να τυπώνει το όνομα του.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) vertebrates mammals herbivores	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε περιγραφή ζώου: elephant
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) invertebrates worms hematophagous	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε περιγραφή ζώου: leech

+ Ασκήσεις Εμπλουτισμού



Άσκηση 3.59

Μία ομάδα εισβολέων συστημάτων με την επωνυμία Ερονηγμους έχουν κοινοποιήσει ένα βίντεο στα μέσα μαζικής ενημέρωσης. Το βίντεο ενημερώνει τους αρμόδιους φορείς ότι την 12η Δεκεμβρίου του 2012, η ώρα 12:12 μ.μ. θα ξεκινήσει μία αντίστροφη μέτρηση, η οποία θα διαρκέσει 12 ώρες ακριβώς. Αν μέσα σε αυτές τις 12 ώρες δεν ικανοποιηθούν όλα τα αιτήματά τους, μεταξύ άλλων και η δωρεάν πρόσβαση στο διαδίκτυο για το κοινό, θα προσβάλουν με ένα επικίνδυνο ιό, επονομαζόμενο Flame, όλες τις κυβερνητικές υπηρεσίες. Η Υπηρεσία Πληροφοριών έχει αναθέσει στην Αριάδνη τον εντοπισμό και την εξουδετέρωση του Flame. Η Αριάδνη ως έμπειρη προγραμματίστρια, ειδική στην αντιμετώπιση κρίσεων, φέρνει σε πέρας την αποστολή και εντοπίζει τον ιό. Έχει προλάβει, όμως, το χρονικό περιθώριο των 12 ωρών;

Δεδομένα εισόδου

Τρεις ακέραιοι, D ($12 \leq D \leq 31$), H ($0 \leq H \leq 23$) και M ($0 \leq M \leq 59$) που υποδηλώνουν την ημερομηνία και την ώρα που εξουδετερώθηκε ο ιός. Για παράδειγμα, 12 23 34 σημαίνει ότι ο ιός εξουδετερώθηκε την 12η Δεκεμβρίου, η ώρα 11:34 μ.μ.

Δεδομένα εξόδου

Ο συνολικός αριθμός των λεπτών που έχουν περάσει από τις 12:12 μ.μ. της 12^{ης} Δεκεμβρίου 2012 και το μήνυμα «You made it!», αν δεν έχουν περάσει 12 ώρες, ή «Too late», σε αντίθετη περίπτωση. Το πρόγραμμα να τυπώνει -1, αν υπάρχει λάθος στα δεδομένα εισόδου.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 12 23 34	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε 3 ακέραιους αριθμούς: Λεπτά =682 You made it!
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 13 12 11	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε 3 ακέραιους αριθμούς: Λεπτά =1439 Too late
Παράδειγμα Εισόδου 3 (πληκτρολόγιο) 0 12 11	Παράδειγμα Εξόδου 3 (οθόνη) Δώσε 3 ακέραιους αριθμούς: -1



Άσκηση 3.60

Το σχολείο της Αριάδνης ετοιμάζεται να επισκεφθεί τον αρχαιολογικό χώρο της Χοιροκοιτίας. Η Αριάδνη είναι υπεύθυνη να πάρει τηλέφωνο και να ρωτήσει για την έκπτωση που δικαιούται το σχολείο. Ο υπεύθυνος του αρχαιολογικού χώρου ενημέρωσε την Αριάδνη ότι αν πάει τουλάχιστον Α αριθμός ατόμων, το σχολείο θα έχει 10% έκπτωση, αν πάει Β αριθμός

ατόμων, θα έχει 20% και αν πάει C αριθμός ατόμων, θα έχει 30%. Η κανονική τιμή του εισιτηρίου είναι 10 ευρώ.

Ο διευθυντής του σχολείου ενημέρωσε την Αριάδνη ότι θα πάνε στην εκδρομή N άτομα. Βοηθήστε την Αριάδνη να γράψει πρόγραμμα που να υπολογίζει το ποσό που πρέπει να πληρώσει το σχολείο. Προσοχή σε ορισμένες περιπτώσεις μπορεί να συμφέρι να αγοραστούν περισσότερα εισιτήρια.

Δεδομένα εισόδου

Η πρώτη γραμμή θα περιέχει έναν ακέραιο αριθμό N ($N \leq 10000$). Η δεύτερη γραμμή θα περιέχει τρεις ακραίους A, B, και C ($A, B, C \leq 10000$).

Δεδομένα εξόδου

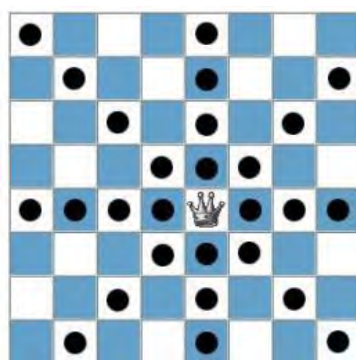
Ένας ακέραιος που θα παρουσιάζει την ελάχιστη συνολική τιμή.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 30 10 20 50	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε ακέραιο αριθμό: Δώσε 3 ακέραιους αριθμούς: Ελάχιστη συνολική τιμή=240
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 48 10 20 50	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε ακέραιο αριθμό: Δώσε 3 ακέραιους αριθμούς: Ελάχιστη συνολική τιμή=350



Άσκηση 3.61

Η βασίλισσα είναι το ισχυρότερο κομμάτι στο σκάκι. Μπορεί να κινείται οριζόντια, κάθετα ή διαγώνια, για όσα τετράγωνα απόσταση θέλει, όπως φαίνεται στην πιο κάτω εικόνα.



Ας υποθέσουμε ότι έχουμε μία σκακιέρα διαστάσεων 8x8 τετραγώνων. Τοποθετούμε πάνω σε αυτή δύο βασίλισσες σε τυχαίες θέσεις. Πόσες είναι οι ελάχιστες κινήσεις που χρειάζεται να κάνει μία από τις δύο βασίλισσες, για να βρεθεί στο ίδιο τετράγωνο με την άλλη βασίλισσα;

Δεδομένα εισόδου

Τέσσερις ακέραιοι αριθμοί x_1, y_1, x_2, y_2 ($1 \leq x_1, y_1, x_2, y_2 \leq 8$). Όπου (x_1, y_1) , η θέση που βρίσκεται η πρώτη βασίλισσα. Το x_1 αντιστοιχεί στη γραμμή και το y_1 στη στήλη. Όπου $(x_2,$

γ2), η θέση της δεύτερης βασίλισσας. Για παράδειγμα, η βασίλισσα στην Εικόνα 1 βρίσκεται στη θέση (5, 5). Οι βασίλισσες μπορεί να τοποθετηθούν αρχικά και στο ίδιο σημείο.

Δεδομένα εξόδου

Ένας ακέραιος, οι ελάχιστες δυνατές κινήσεις που χρειάζεται να κάνει μία από τις δύο βασίλισσες, για να βρεθεί στο ίδιο τετράγωνο με την άλλη.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
4 4 6 2	Δώσε 4 ακέραιους αριθμούς: 1



Άσκηση 3.62

Στο άθλημα της επιτραπέζιας αντισφαίρισης (ping-pong) ο κάθε πόντος ξεκινά με σέρβις από έναν παίκτη. Αρχικά, σερβίρει ο πρώτος παίκτης για τους πρώτους δύο πόντους και ακολούθως, ο δεύτερος παίκτης για τους επόμενους δύο. Αυτό συνεχίζεται μέχρι να φτάσει ένας παίκτης στους 11 πόντους, ενώ ο άλλος έχει 9 ή λιγότερους, οπότε και τελειώνει το σετ. Αν το σκορ είναι στο 10-10, τότε οι παίκτες συνεχίζουν να παίζουν, κάνοντας από ένα σέρβις ο καθένας, μέχρι ένας παίκτης να αποκτήσει διαφορά δύο πόντων. Στο επόμενο σετ ξεκινά να σερβίρει ο παίκτης ο οποίος στο προηγούμενο σετ έκανε πρώτος υποδοχή.

Ας υποθέσουμε ότι αγωνίζονται ο Andy εναντίον του Roger, με τον Andy να σερβίρει πρώτος. Να γράψετε πρόγραμμα το οποίο να δέχεται τον αριθμό του σετ ($1 \leq S \leq 5$) και το μέχρι στιγμής σκορ στο οποίο βρίσκεται το παιγνίδι και θα τυπώνει το όνομα του παίκτη που πρέπει να σερβίρει.

Δεδομένα εισόδου

- Ακέραιος αριθμός S ($1 \leq S \leq 5$), ο αριθμός των σετ.
- Το επιμέρους σκορ του αγώνα με πρώτο τον Andy και δεύτερο τον Roger ($0 \leq A, R \leq 100$).

Δεδομένα εξόδου

Το όνομα του παίκτη που σερβίρει τον επόμενο πόντο (Andy ή Roger).

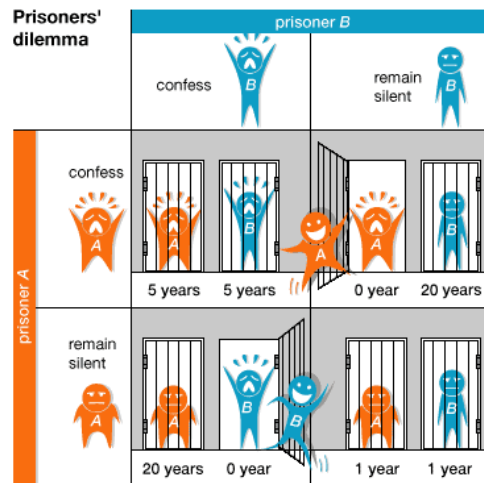
Παράδειγμα Εισόδου 1 (πληκτρολόγιο)	Παράδειγμα Εξόδου 1 (οθόνη)
1 5 3	Δώσε ακέραιο αριθμό: Δώσε σκορ: Andy
Παράδειγμα Εισόδου 2 (πληκτρολόγιο)	Παράδειγμα Εξόδου 2 (οθόνη)
4 16 16	Δώσε ακέραιο αριθμό: Δώσε σκορ: Roger



Άσκηση 3.63

Το πρόβλημα αυτό είναι γνωστό ως δίλημμα του φυλακισμένου. Η αστυνομία έχει συλλάβει δύο συνεργούς σε αδίκημα. Στον καθένα προτείνεται ο εξής συμβιβασμός:

- Ομολόγησε και πρόδωσε τον συνεργάτη σου όσο αυτός δεν έχει ομολογήσει ακόμη. Αν συμβεί αυτό, εσύ θα αφεθείς ελεύθερος, ενώ ο συνεργός σου θα φυλακιστεί για 20 χρόνια.
- Αν ομολογήσει και σε προδώσει αυτός, τότε θα αφεθεί αυτός ελεύθερος και θα φυλακιστείς εσύ για 20 χρόνια.
- Αν ομολογήσετε και προδώσετε και οι δύο ταυτόχρονα, τότε θα φυλακιστείτε και οι δύο για 5 χρόνια ο καθένας.
- Αν δεν ομολογήσει κανείς από τους δύο, τότε, λόγω έλλειψης στοιχείων, θα καταδικαστείτε και οι δύο για ελαφρύτερα αδικήματα, σε 1 χρόνο ο καθένας.



Το πρόβλημα είναι τι πρέπει να κάνει ο κάθε φυλακισμένος, χωρίς να γνωρίζει τι θα κάνει ο άλλος. Αν μπορούσαν να συνεννοηθούν, θα προτιμούσαν να μην προδώσει κανένας και να γλιτώσουν με μικρότερη ποινή. Να γράψετε πρόγραμμα το οποίο να δέχεται αν ομολόγησαν ή όχι οι δύο συνεργοί ρωτώντας για τον καθένα χωριστά και έπειτα να τυπώνει τα χρόνια φυλακής που καταδικάστηκε ο καθένας.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 1 1	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε 2 αριθμούς: Τα χρόνια φυλάκισης είναι 5 και 5
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 1 0	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε 2 αριθμούς: Τα χρόνια φυλάκισης είναι 0 και 20

B7.1.4 Δομές Επανάληψης

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ❖ Να διατυπώνουμε τη δομή και το λογικό διάγραμμα της δομής επανάληψης while
- ❖ Να διατυπώνουμε τη δομή και το λογικό διάγραμμα της δομής επανάληψης for
- ❖ Να χρησιμοποιούμε τελεστές αυξομείωσης (++) και --) στη μεταβλητή ελέγχου της δομής for
- ❖ Να διατυπώνουμε τη δομή και το λογικό διάγραμμα της δομής επανάληψης do .. while
- ❖ Να επιλέγουμε την κατάλληλη δομή επανάληψης με βάση την περιγραφή συγκεκριμένου προβλήματος
- ❖ Να σχεδιάζουμε το λογικό διάγραμμα και τις αντίστοιχες εντολές προγράμματος, που να ανταποκρίνονται σε συγκεκριμένη εκφώνηση που αναφέρεται σε δομή επανάληψης
- ❖ Να καθορίζουμε το πρόβλημα με ακρίβεια, συγκεκριμένα: να εντοπίζουμε/διακρίνουμε τα Δεδομένα, τις Πληροφορίες και την Επεξεργασία
- ❖ Να δημιουργούμε το Λογικό Διάγραμμα για την επίλυση του προβλήματος
- ❖ Να επιλέγουμε κατάλληλη δομή επανάληψης ανάλογα με τις δυνατότητες, τους περιορισμούς και τα χαρακτηριστικά της για επίλυση του προβλήματος
- ❖ Να μεταφράζουμε ένα λογικό διάγραμμα σε πρόγραμμα με τη χρήση του προγραμματιστικού περιβάλλοντος, ώστε να επιλυθεί το πρόβλημα
- ❖ Να επιλέγουμε κατάλληλα δεδομένα και στρατηγική για έλεγχο του προγράμματος
- ❖ Να ελέγχουμε την ορθότητα της λύσης του προβλήματος, χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης και της χρήσης του debugger για επαλήθευση
- ❖ Να μελετούμε έτοιμο πρόγραμμα, το οποίο περιλαμβάνει δομές επανάληψης και να εντοπίζουμε βασικά μέρη του, τα οποία συνδέονται με πτυχές του προβλήματος που επιλύει
- ❖ Να συμπληρώνουμε ένα έτοιμο πρόγραμμα με δομή/δομές επανάληψης, ώστε να αποτελεί λύση ενός διαφοροποιημένου προβλήματος
- ❖ Να εντοπίζουμε και να αναγνωρίζουμε σε ένα πρόγραμμα πρότυπα σχεδίασης και στρατηγικές (design patterns, τμήματα κώδικα)
- ❖ Να χρησιμοποιούμε πρότυπα σχεδίασης και στρατηγικές που εντοπίσαμε ως εργαλεία επίλυσης προβλημάτων σε νέα προγράμματά μας.

1. Εισαγωγή

Σε αυτό το κεφάλαιο θα δούμε τις δομές επανάληψης (iterative statements). Δομές επανάληψης ή βρόχοι (loops) ονομάζονται μέρη του κώδικα που εκτελούνται περισσότερες από μία φορές, ανάλογα με τη συνθήκη που έχουμε δηλώσει.

Οι βρόχοι αναγκάζουν ένα τμήμα κώδικα να επαναλαμβάνεται. Η επανάληψη συνεχίζεται για όσο μία λογική συνθήκη είναι αληθής. Όταν η συνθήκη είναι ψευδής, ο βρόχος τελειώνει και ο έλεγχος του προγράμματος συνεχίζεται παρακάτω. Στη C++ υπάρχουν τρία είδη βρόχων: ο βρόχος for, ο βρόχος while και ο βρόχος do.

2. Τελεστές αύξησης/μείωσης (++, --)

Με τους τελεστές αύξησης (++) και μείωσης (--), μπορούμε να αυξήσουμε ή να μειώσουμε αντίστοιχα κατά μία μονάδα την τιμή που είναι αποθηκευμένη στη μεταβλητή. Έτσι, τα παρακάτω είναι ισοδύναμα:

```
a++;  
a += 1;  
a = a + 1;
```

Θα πρέπει να δοθεί προσοχή στο εάν ο τελεστής αύξησης ή μείωσης χρησιμοποιείται ως πρόθεμα (++a) ή ως επίθεμα (a++). Εάν χρησιμοποιείται ως πρόθεμα, πρώτα αυξάνεται ή μειώνεται κατά ένα η τιμή της και μετά πραγματοποιούνται όποιες πράξεις υπάρχουν στην

εντολή. Ενώ όταν χρησιμοποιείται ως επίθεμα, πρώτα εκτελούνται οι πράξεις της εντολής και στο τέλος αλλάζει κατά μία μονάδα η τιμή της μεταβλητής.

```
b = 4;  
a = ++b;    // a = 5
```

Μετά τη δεύτερη εντολή $a=5$ και $b=5$, γιατί πρώτα το b αυξήθηκε κατά ένα και μετά εκτελέστηκε η εντολή. Ενώ αντίθετα:

```
b = 4;  
a = b++;    // a = 4
```

Μετά τη δεύτερη εντολή $a=4$ και $b=5$, γιατί πρώτα εκτελέστηκε η εντολή και το a πήρε την τιμή του b και μετά το b αυξήθηκε κατά ένα.

3. Ο βρόχος for

Θα ξεκινήσουμε με το εξής απλό παράδειγμα. Θέλουμε να εμφανίσουμε τους αριθμούς από το 1 μέχρι το 10 στην οθόνη.

Παράδειγμα 4.1

```
#include<iostream>  
using namespace std;  
int main() {  
    int i;  
    for (i = 1; i <= 10; i++) {  
        cout << i << endl;  
    }  
    return 0;  
}
```

Ο βρόχος for είναι η πιο απλή επαναληπτική δήλωση, καθώς μας επιτρέπει να δημιουργούμε επαναληπτικές δομές με την ελάχιστη δυνατή χρήση κώδικα. Ο βρόχος for χρησιμοποιείται όταν γνωρίζουμε από πριν τον ακριβή αριθμό των επαναλήψεων. Μετά τη δεσμευμένη λέξη for ακολουθούν μέσα σε παρένθεση οι οδηγίες εκτέλεσης του βρόχου. Μία μεταβλητή παίζει τον ρόλο του μετρητή των επαναλήψεων. Έτσι, η μεταβλητή-μετρητής λαμβάνει αρχική τιμή. Αυτή η εντολή εκτελείται μόνο την πρώτη φορά, στην αρχή του βρόχου. Ακολουθεί μία λογική συνθήκη. Όσο η λογική συνθήκη είναι αληθής, ο βρόχος θα επαναλαμβάνει τις εντολές που περιέχει. Όταν η λογική συνθήκη γίνει ψευδής, ο έλεγχος του προγράμματος συνεχίζει με το υπόλοιπο πρόγραμμα εκτός βρόχου. Η εξέταση της λογικής συνθήκης γίνεται πριν ο έλεγχος του προγράμματος εισέλθει στον βρόχο. Η τελευταία παράμετρος μέσα στις παρενθέσεις είναι η μεταβολή του μετρητή. Είδαμε ότι ο τελεστής ++ αυξάνει την τιμή της μεταβλητής κατά μία μονάδα.

Στο παράδειγμα 4.1, η εντολή for, με τη χρήση ενός μετρητή, μας επιτρέπει μέσα σε μία γραμμή κώδικα να:

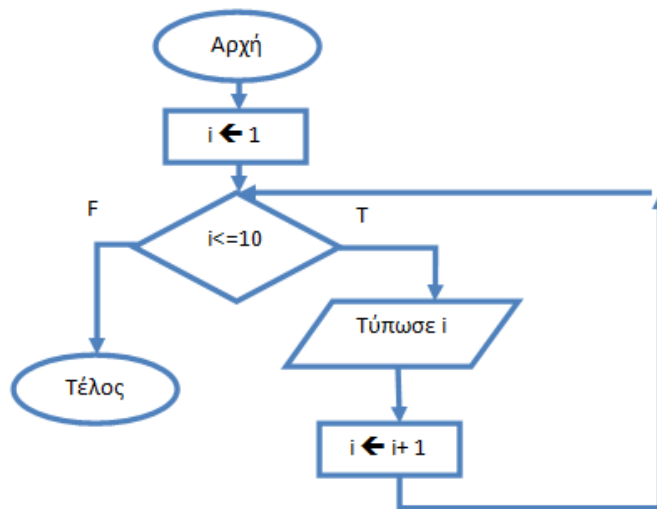
- 1) Ορίσουμε την αρχική τιμή του μετρητή ($i = 1$).
- 2) Ορίσουμε τη συνθήκη, για την οποία θα εκτελείται ο βρόχος ($i <= 10$).

3) Ορίσουμε την τιμή μεταβολής του μετρητή ($i++$).

Η γενική μορφή της εντολής έχει ως εξής:

①	②	③
for	(αρχική τιμή μετρητή; συνθήκη ελέγχου μετρητή; μεταβολή τιμής μετρητή) {	
	εντολές εντός του βρόχου	
	}	

Το πιο κάτω λογικό διάγραμμα μάς παρουσιάζει πώς ακριβώς εκτελείται ο κώδικας του παραδείγματος 4.1.



Ο μετρητής που χρησιμοποιεί ένας βρόχος μπορεί να μην είναι πάντοτε ακέραιος αριθμός, όπως βλέπουμε και στο πιο κάτω παράδειγμα:

Παράδειγμα 4.2

```
#include<iostream>
using namespace std;
int main() {
    char letter;
    for (letter = 'a'; letter <= 'z'; letter++){
        cout << letter << endl;
    }
    for (double fx = 0.5; fx < 3.5; fx += 0.5)
        cout << fx << endl;
    return 0;
}
```

Παρατηρήσεις:

- Ο μετρητής μας μπορεί να είναι χαρακτήρας ή δεκαδικός αριθμός.

- Μπορεί να δηλώσουμε και ταυτόχρονα να αρχικοποιήσουμε μία μεταβλητή-μετρητή μέσα στον βρόχο.
- Τα άγκιστρα μπορούν να παραλείπονται, αν έχουμε μόνο μία εντολή.

Παράδειγμα 4.3

Να γράψετε πρόγραμμα το οποίο να τυπώνει στην οθόνη τα πολλαπλάσια του 7, ξεκινώντας από τον αριθμό 21 μέχρι τον αριθμό 777.

```
#include<iostream>
using namespace std;
int main() {
    int i;
    for (i = 21; i <= 777; i+=7) {
        cout << i << endl;
    }
    return 0;
}
```

Παράδειγμα 4.4

Να γράψετε πρόγραμμα το οποίο να διαβάζει τη βάση ($b \geq 0$) και τον εκθέτη ($e \geq 0$) μίας δύναμης και να υπολογίζει και να τυπώνει τη βάση υψωμένη στον εκθέτη. Π.χ. $2^3 = 8$. Να γράψετε, επίσης, τον πίνακα προκαταρκτικής εκτέλεσης για τιμές $b=5$ και $e=3$.

Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>
using namespace std;
int main() {
    int b,e,i,result = 1;
    cin >> b >> e;
    for (i = 1; i <= e; i++) {
        result = result * b;
    }
    cout << result << endl;
    return 0;
}
```

Πίνακας προκαταρκτικής εκτέλεσης

Μεταβλητές				Αποφάσεις	T/F	Παρουσίαση
b	e	i	result	$i \leq e$		125
5	3		1		T	
		1	5	$1 \leq 3$	T	
		2	25	$2 \leq 3$	T	
		3	125	$3 \leq 3$	T	
		4		$4 \leq 3$	F	

Υλοποίηση στη γλώσσα προγραμματισμού Pascal

```
program par4_4;
var
  i,b,e,result: integer;
Begin
  readln(b,e);
  result := 1;
  for i := 1 to e do
    result := result * b;
  writeln(result)
End.
```



Υλοποίηση στη γλώσσα προγραμματισμού Python

```
result=1
b=int(input())
e=int(input())
for i in range(1,e+1):
    result *= b
print(result)
```



Υλοποίηση στη γλώσσα προγραμματισμού Java

```
import java.util.Scanner;
public class main {
    public static void main(String arg[]){
        int e,b,i,result=1;
        Scanner in = new Scanner(System.in);
        b=in.nextInt();
        e=in.nextInt();
        for(i = 1; i <= e; i++) {
            result = result * b;
        }
        System.out.print(result);
        in.close();
    }
}
```



4. Φωλιασμένες εντολές for – Nested for loops

Σε πολλές περιπτώσεις μπορούμε να χρησιμοποιήσουμε επαναληπτικές εντολές μέσα σε άλλες επαναληπτικές εντολές. Στην περίπτωση αυτή, για κάθε επανάληψη της εξωτερικής εντολής, η εσωτερική θα εκτελέσει όλες τις δικές της επαναλήψεις. Ας δούμε παραδείγματα:

Παράδειγμα 4.5

```

#include<iostream>
using namespace std;
int main() {                                // Output
    for (int i=1;i<=3;i++){                // 1 4
        for (int j=4;j<=5;j++){            // 1 5
            cout << i << " " << j << endl; // 2 4
        }                                  // 2 5
    }                                       // 3 4
    return 0;                              // 3 5
}

```

Παράδειγμα 4.6

```

#include<iostream>
using namespace std;
int main() {                                // Output

    for (int i = 1; i <= 8; i++){            // *
        for (int j = 1; j <= i; j++){        // **
            cout << "*";                     // ***
        }                                    // ****
        cout << endl;                        // *****
    }                                       // ****
    return 0;                              // *****
}                                           // *****

```

5. Ο βρόχος while

Είδαμε ότι ο βρόχος for χρησιμοποιείται όταν ξέρουμε τον ακριβή αριθμό των επαναλήψεων μίας δομής. Τι γίνεται, όμως, όταν θέλουμε να επαναλαμβάνονται κάποιες εντολές, χωρίς να ξέρουμε από πριν τον αριθμό των επαναλήψεων; Για παράδειγμα, να διαβάζουμε αριθμούς μέχρι να δοθεί ο αριθμός μηδέν. Σε αυτές τις περιπτώσεις χρησιμοποιούμε τον βρόχο while.

Για να επαναλάβει τις εντολές του ο βρόχος while, ελέγχει τη λογική συνθήκη και εάν αυτή είναι αληθής, τότε επαναλαμβάνει τις εντολές που του έχουμε ορίσει. Όταν εκτελέσει όλες τις εντολές του, ο έλεγχος μεταβιβάζεται πάλι στην αρχή και επανεξετάζει τη λογική συνθήκη. Όσο η λογική συνθήκη παραμένει αληθής, ο βρόχος θα επαναλαμβάνεται. Όταν η λογική συνθήκη γίνει ψευδής, τότε ο έλεγχος του προγράμματος θα συνεχίσει με το υπόλοιπο πρόγραμμα μετά από αυτόν.

Παράδειγμα 4.7

```
#include<iostream>
using namespace std;
int main() {

    int n;
    cin >> n;
    while (n != 0) {
        cout << "You pressed: " << n << endl;
        cin >> n;
    }

    return 0;
}
```

Στο παράδειγμα 4.7 ζητάμε από τον χρήστη να δώσει έναν ακέραιο αριθμό και τον τυπώνουμε στην οθόνη. Ο βρόχος σταματά, όταν ο χρήστης δώσει το μηδέν. Σε αυτή την περίπτωση δεν γνωρίζουμε πόσες φορές θα εκτελεστεί ο βρόχος. Όσο ο χρήστης επιμένει να μην δίνει για είσοδο το μηδέν, ο βρόχος θα εκτελείται ξανά και ξανά. Επειδή δεν γνωρίζουμε τον αριθμό των επαναλήψεων, δεν μπορούμε να χρησιμοποιήσουμε τον βρόχο for. Επίσης, η επανάληψη των εντολών βασίζεται σε μία λογική συνθήκη. Αυτό μας υποδεικνύει τη χρησιμοποίηση βρόχου while για την επανάληψη.

Ας χρησιμοποιήσουμε τον βρόχο while, για να τυπώσουμε τους αριθμούς από το 1 μέχρι το 10 στην οθόνη (παράδειγμα 4.1):

```
int i = 1;                // αρχική τιμή μετρητή
while(i <= 10) {          // λογική συνθήκη ελέγχου μετρητή
    cout << i << " ";
    i++;                  // μεταβολή τιμής μετρητή
}
```

Η γενική μορφή της εντολής while συγκριτικά με τη δομή for έχει ως εξής:

Βρόχος for

①	②	③
for (αρχική τιμή μετρητή; συνθήκη ελέγχου μετρητή; μεταβολή τιμής μετρητή) {		
εντολές εντός του βρόχου		
}		

Βρόχος while

```

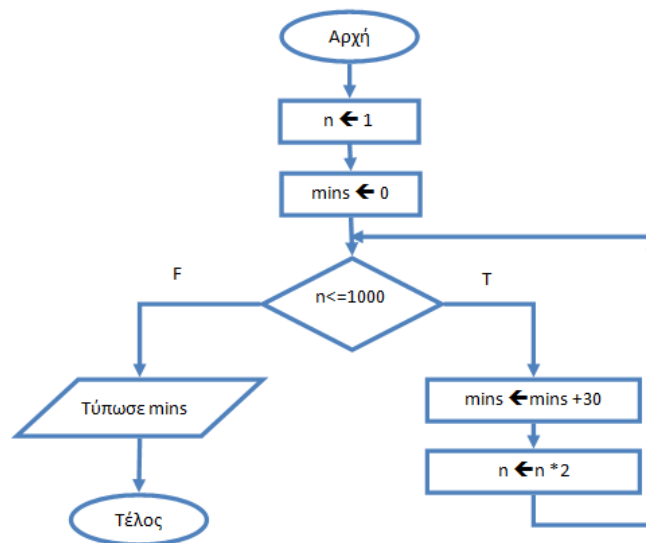
αρχική τιμή μετρητή ①

while (συνθήκη ελέγχου μετρητή) { ②
    εντολές εντός του βρόχου
    μεταβολή τιμής μετρητή ③
}

```

Παράδειγμα 4.8

Οι μονοκύτταροι οργανισμοί, όπως η αμοιβάδα, αναπαράγονται συνήθως με μονογονία. Στην αρχή διπλασιάζεται το γενετικό υλικό της αμοιβάδας και στη συνέχεια, με διαίρεση, δημιουργούνται δύο νέες όμοιες αμοιβάδες. Αυτό συμβαίνει κάθε 30 λεπτά. Να σχεδιάσετε λογικό διάγραμμα και να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει πόσα λεπτά χρειάζονται, ώστε από μία μόνο αμοιβάδα να δημιουργηθούν περισσότερες από 1000 αμοιβάδες.

*Υλοποίηση στη γλώσσα προγραμματισμού C++*

```

#include<iostream>
using namespace std;
int main() {
    int n = 1, mins = 0;
    while (n <= 1000) {
        mins += 30;
        n *= 2;
    }
    cout << mins << endl;
    return 0;
}

```



Υλοποίηση στη γλώσσα προγραμματισμού Pascal

```
program par4_8;
var
    n, mins : integer;
begin
n:= 1;
mins:= 0;
while n <=1000 do
    begin
        mins:=mins+30;
        n:=n*2;
    end;
    writeln(mins)
end.
```

*Υλοποίηση στη γλώσσα προγραμματισμού Python*

```
n=1
mins=0
while (n<=1000):
    mins+=30
    n*=2
print(mins)
```

*Υλοποίηση στη γλώσσα προγραμματισμού Java*

```
package par4_8;
public class main {
    public static void main(String arg[]){
        int n = 1, mins = 0;
        while (n <=1000){
            mins+=30;
            n*=2;
        }
        System.out.print(mins);
    }
}
```



Παράδειγμα 4.9

Να γράψετε πρόγραμμα το οποίο να δέχεται αριθμούς, μέχρι που το άθροισμα τους να ξεπεράσει το 999. Όταν γίνει αυτό, να υπολογίζει και να τυπώνει το πλήθος των αριθμών που έχουν δοθεί.

```
#include<iostream>
using namespace std;
int main() {
    int plithos = 0, sum = 0, num;
    while (sum <= 999){
        cin >> num;
        plithos+=1;
        sum+=num;
    }
    cout << plithos << endl;
    return 0;
}
```

Παράδειγμα 4.10

Ο πληθυσμός μίας αποικίας μυρμηγκιών αυξάνεται κατά 12% κάθε μήνα. Ταυτόχρονα, λόγω δυσμενών συνθηκών περίπου 10,000 μυρμηγκία πεθαίνουν στο ίδιο διάστημα. Αν ο αρχικός πληθυσμός της αποικίας ήταν 85,000, να υπολογίσετε σε πόσους μήνες η αποικία θα ξεπεράσει τις 200,000.

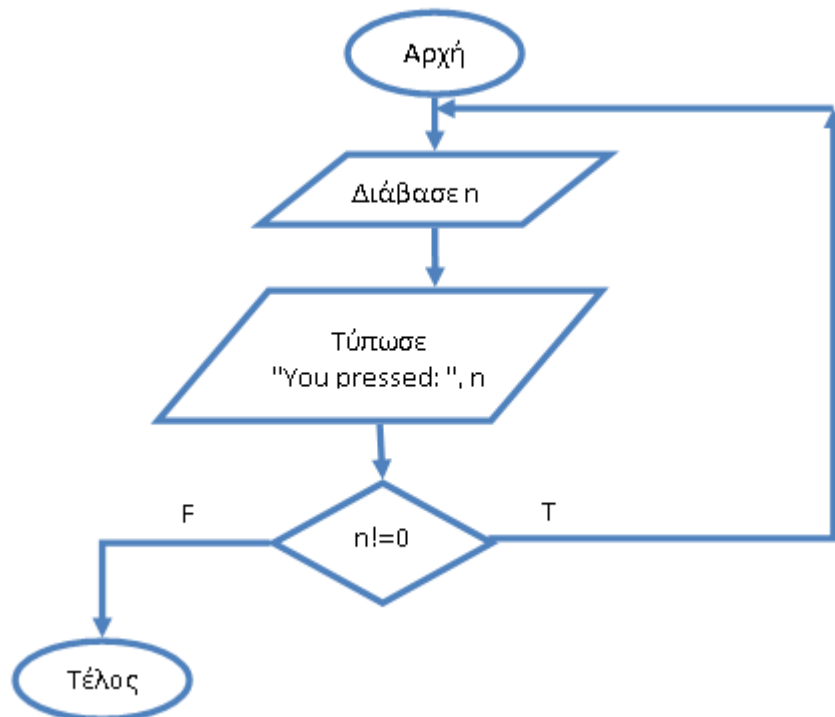
```
#include<iostream>
using namespace std;
int main() {
    int pop = 85000, month = 0;
    while (pop <= 200000){
        month+=1;
        pop = pop * 112/100 - 10000;
    }
    cout << month << endl;
    return 0;
}
```

6. Ο βρόχος do

Όπως προαναφέρθηκε, σε έναν βρόχο while η λογική συνθήκη ελέγχεται στην αρχή. Αν αυτή είναι ψευδής, τότε το σώμα του βρόχου δεν εκτελείται καθόλου. Σε μερικές περιπτώσεις, όμως, θέλουμε να εξασφαλίσουμε ότι οι εντολές του βρόχου θα εκτελεστούν έστω και μία φορά, ανεξάρτητα από την τιμή της λογικής συνθήκης. Σε αυτή την περίπτωση χρησιμοποιούμε τον βρόχο do, ο οποίος κάνει τον έλεγχο της λογικής συνθήκης στο τέλος του βρόχου. Η διαφορά, δηλαδή, έγκειται στο ότι ο βρόχος while κάνει τον έλεγχο πριν ξεκινήσει να εκτελεί τις εντολές, ενώ ο βρόχος do στο τέλος.

Παράδειγμα 4.11

Να σχεδιάσετε λογικό διάγραμμα και να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να τον τυπώνει στην οθόνη. Το πρόγραμμα θα σταματήσει να δέχεται αριθμούς, όταν δοθεί ο αριθμός 0.



Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>
using namespace std;
int main() {
    int n;

    do {
        cin >> n;
        cout << "You pressed: "<<n <<endl;
    }
    while (n != 0);

    return 0;
}
```

Αρχικά, η δεσμευμένη λέξη `do` σηματοδοτεί την έναρξη του βρόχου. Έπειτα, ακολουθεί το σώμα του βρόχου. Στο τέλος υπάρχει η λέξη `while` με τη λογική συνθήκη που ελέγχει, μέσα σε παρενθέσεις. Προσοχή στο ερωτηματικό που υπάρχει μετά τη συνθήκη, το οποίο σηματοδοτεί το τέλος της πρότασης. Όσο η λογική συνθήκη είναι αληθής, ο βρόχος θα εκτελεί τις εντολές του. Όταν η λογική συνθήκη γίνει ψευδής, ο έλεγχος του προγράμματος θα συνεχίσει παρακάτω. Όπως βλέπετε, ο έλεγχος της λογικής συνθήκης με τη δεσμευμένη

λέξη `while` γίνεται στο τέλος του βρόχου. Αυτό εξασφαλίζει την εκτέλεση του βρόχου τουλάχιστον μία φορά, ακόμα και όταν η λογική συνθήκη είναι ψευδής.

Παρατήρηση: Δώστε προσοχή στη μη αρχικοποίηση της μεταβλητής `n`, σε αντίθεση με το παράδειγμα 4.7. Στον βρόχο `while`, η μεταβλητή που συμμετέχει στη λογική συνθήκη θα πρέπει να έχει ήδη λάβει τιμή, ώστε να εκτιμηθεί η συνθήκη ως αληθής ή ψευδής.

Παράδειγμα 4.12

Να γράψετε πρόγραμμα το οποίο να δέχεται δύο ακέραιους αριθμούς και να τυπώνει στην οθόνη τις επιλογές: 1. Addition, 2. Multiplication, 3. Subtraction, 4. Division, 5. Modulo, 6. Exit. Με βάση την επιλογή του χρήστη, να υπολογίζει και να τυπώνει το αποτέλεσμα της αντίστοιχης πράξης.

Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>
using namespace std;
int main() {
    int op,a,b;
    cout << "Enter two numbers (a>b): ";
    cin >> a >> b;
    do {
        cout << "1. Addition" << endl;
        cout << "2. Multiplication" << endl;
        cout << "3. Subtraction" << endl;
        cout << "4. Division" << endl;
        cout << "5. Modulo" << endl;
        cout << "6. Exit" << endl;
        cout << "Please make a choice: ";
        cin >> op;
        switch(op) {
            case 1:
                cout << a << "+" << b << "=" << a+b << endl;
                break;
            case 2:
                cout << a << "*" << b << "=" << a*b << endl;
                break;
            case 3:
                cout << a << "-" << b << "=" << a-b << endl;
                break;
            case 4:
                cout << a << "/" << b << "=" << a/b << endl;
                break;
            case 5:
```

```

        cout << a << "%" << b << "=" << a%b << endl;
        break;
    case 6:
        cout << "Program ended" << endl;
        break;
    default:
        cout << "Wrong input" << endl;
    }
} while (op != 6);
return 0;
}

```

7. Η Βιβλιοθήκη <climits>

Η βιβλιοθήκη <climits> καθορίζει ως σταθερές (INT_MAX, INT_MIN), τα αριθμητικά όρια για μεταβλητές ακέραιου τύπου.

```

#include <iostream>
#include <climits>           // Integer limits
using namespace std;
int main() {
    cout << INT_MAX << endl;      // 2147483647
    cout << INT_MIN << endl;     // -2147483648
    return 0;
}

```

Παράδειγμα 4.13

Να γράψετε πρόγραμμα το οποίο να δέχεται 100 ακέραιους αριθμούς και να υπολογίζει και τυπώνει τον μικρότερο αριθμό και τον μεγαλύτερο αριθμό.

```

#include<iostream>
#include<climits>
using namespace std;
int main(){
    int num,max=INT_MIN,min=INT_MAX;
    for (int i = 1; i <= 100; i++){
        cin >> num;
        if (num>max)
            max=num;
        if (num<min)
            min=num;}
    cout<<"Minimum=" << min << endl;
    cout<<"Maximum=" << max;
    return 0;}

```

8. Έλεγχος ορθότητας δεδομένων

Πολλές φορές μας ζητείται να ελέγχουμε κατά πόσο τα δεδομένα που δίνονται από τον χρήστη τηρούν τις επιτρεπόμενες τιμές. Σε αυτή την περίπτωση χρησιμοποιούμε δομή επανάληψης.

Παράδειγμα 4.14

Να γράψετε πρόγραμμα το οποίο να δέχεται τις βαθμολογίες είκοσι μαθητών μιας τάξης. Στην περίπτωση που η βαθμολογία δεν είναι μεταξύ των τιμών 1 και 100 συμπεριλαμβανομένων, το πρόγραμμα να τυπώνει μήνυμα λάθους και να ζητά από τον χρήστη να την δώσει ξανά. Στο τέλος, να υπολογίζει και να τυπώνει το πλήθος των μαθητών που πήραν βαθμολογία πάνω από το 50.

```
#include<iostream>
using namespace std;
int main(){
    int pl = 0, B;
    for (int i = 1; i <= 20; i++){
        cout << " Δώσε βαθμολογία:" ;
        cin >> B;
        while (B < 1 || B > 100){
            cout << "Λάθος τιμή. Δώσε βαθμολογία:";
            cin >> B;
        }
        if (B>50)
            pl++;
    }
    cout << "Πλήθος=" << pl << endl;
    return 0;
}
```

9. Έλεγχος Προγράμματος

Ο έλεγχος του προγράμματος είναι μία πολύ σημαντική διαδικασία. Θα μπορούσαμε να τον χωρίσουμε σε 2 κατηγορίες: α) συντακτικός, β) λογικός. Ο συντακτικός έλεγχος μπορεί να πραγματοποιηθεί εύκολα με τη βοήθεια του compiler της γλώσσας που χρησιμοποιούμε, όπως είδαμε και στο Κεφάλαιο 1. Συνηθισμένα συντακτικά λάθη στη C++ είναι η μη δήλωση μεταβλητών και βιβλιοθηκών και η απουσία του ερωτηματικού ';' στο τέλος μίας εντολής. Ο λογικός έλεγχος είναι πιο δύσκολος και τα λάθη δεν εντοπίζονται εύκολα. Θα μπορούσαμε να χωρίσουμε τον λογικό έλεγχο σε 2 βήματα: α) επιλογή εισόδων που θα καλύπτουν όλες τις πιθανές περιπτώσεις, β) έλεγχος αποτελεσμάτων.

Ας δοκιμάσουμε να πραγματοποιήσουμε λογικό έλεγχο στο παράδειγμα 4.4, που ζητούσε να γράψουμε ένα πρόγραμμα για τον υπολογισμό της δύναμης, όταν ο εκθέτης και η βάση είναι θετικοί αριθμοί.

- Επιλογή Εισόδων

- Επιλέγουμε εισόδους για τις μη ακραίες περιπτώσεις, δηλαδή για τη συνηθέστερη λειτουργία του προγράμματος, π.χ. $b=3$ $e=4$
- Επιλέγουμε εισόδους για τις ακραίες περιπτώσεις. Στην περίπτωση μας έχουμε τα εξής:
 - $b=$ θετικός ακέραιος, $e=0$
 - $b=0$, $e=$ θετικός ακέραιος
 - $b=0$, $e=0$
 - $b=$ μεγάλος ακέραιος (π.χ. 10,000), $e=$ μεγάλος ακέραιος (π.χ. 10,000)

Αφού επιλέξαμε τις εισόδους, πρέπει να ελέγχουμε ότι για τις συγκεκριμένες εισόδους το πρόγραμμα δίνει το σωστό αποτέλεσμα. Υπάρχουν διάφοροι τρόποι να γίνει αυτό, όπως:

α) με έναν αναλυτικό πίνακα προκαταρκτικής εκτέλεσης, που παρουσιάζει όλα τα βήματα, τις τιμές των μεταβλητών και των αποφάσεων του προγράμματος,

β) με έλεγχο του τελικού αποτελέσματος (αν γνωρίζουμε από πριν πιο είναι αυτό),

γ) με την εκτύπωση ενδιάμεσων αποτελεσμάτων (η μέθοδος αυτή είναι πολύ χρήσιμη, όταν εντοπίσουμε κάποιο λάθος).

Θα παρουσιάσουμε και τους τρεις τρόπους (οι οποίοι δεν είναι οι μοναδικοί) πιο κάτω.

α) Προκαταρκτική εκτέλεση με αναλυτικό πίνακα. Είσοδος $b=3$ $e=4$.

Μεταβλητές				Αποφάσεις		Παρουσίαση
b	e	i	result	$i \leq e$	T/F	81
3	4		1			
		1	3	$1 \leq 4$	T	
		2	9	$2 \leq 4$	T	
		3	27	$3 \leq 4$	T	
		4	81	$4 \leq 4$	T	
		5		$5 \leq 4$	F	

β) Έλεγχος τελικού αποτελέσματος.

b	e	result
5	0	5 0 1
0	5	0 5 0
0	0	0 0 1

γ) Εκτύπωση ενδιάμεσων αποτελεσμάτων

Αν δοκιμάσετε να υπολογίσετε με το πρόγραμμα που παρουσιάσαμε στο παράδειγμα 4.4 τη δύναμη 10^{15} , θα πάρετε λανθασμένο αποτέλεσμα, όπως φαίνεται πιο κάτω:

```
10 15
-1530494976
```

Το σωστό αποτέλεσμα είναι 1,000,000,000,000,000. Ο λόγος που εμφανίζεται το λάθος είναι επειδή γίνεται υπερχείλιση του αριθμού, δηλαδή το αποτέλεσμα δεν χωράει να αποθηκευτεί μέσα στο result που είναι τύπου integer. Για να δούμε πού έγινε η υπερχείλιση, εκτυπώνουμε το αποτέλεσμα μετά από κάθε πολλαπλασιασμό:

```
for (int i = 1; i <= e; i++) {  
    result = result * b;  
    cout << "For i=" << i << " result=" << result << endl;  
}
```

Παρατηρούμε ότι για $i \geq 10$ δεν τυπώνει το σωστό αποτέλεσμα λόγω της υπερχείλισης. Άρα το πρόγραμμα που φτιάξαμε έχει ένα όριο στο μέγεθος του αποτελέσματος.

```
10 15  
For i=1 result=10  
For i=2 result=100  
For i=3 result=1000  
For i=4 result=10000  
For i=5 result=100000  
For i=6 result=1000000  
For i=7 result=10000000  
For i=8 result=100000000  
For i=9 result=1000000000  
For i=10 result=1410065408  
For i=11 result=1215752192  
For i=12 result=-727379968  
For i=13 result=1316134912  
For i=14 result=276447232  
For i=15 result=-1530494976
```



Ασκήσεις Κεφαλαίου

Άσκηση 4.1

Ποια από τις πιο κάτω δομές επανάληψης τυπώνει το σύμβολο «*» πέντε φορές στην οθόνη;

(α) <pre>for (i=8; i>=3; i--) cout << "*";</pre>	(γ) <pre>for (i=10; i<21; i+=2) cout << "*";</pre>
(β) <pre>for (i=6; i>=2; i--) cout << "*";</pre>	(δ) <pre>for (i=0; i<=6; i++) cout << "*";</pre>

Άσκηση 4.2

Να γράψετε πρόγραμμα το οποίο να τυπώνει όλα τα κεφαλαία γράμματα του αγγλικού αλφαβήτου (από το A μέχρι και το Z) με ένα κενό χαρακτήρα ανάμεσά τους.

Παράδειγμα Εξόδου (οθόνη)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Άσκηση 4.3

Να γράψετε πρόγραμμα το οποίο να τυπώνει τους αριθμούς όπως παρουσιάζονται στο πιο κάτω παράδειγμα εξόδου.

Παράδειγμα Εξόδου (οθόνη)

1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

Άσκηση 4.4

Να γράψετε πρόγραμμα το οποίο να τυπώνει όλους τους τετραψήφιους αριθμούς στην οθόνη, χωρισμένους με κενό.

Παράδειγμα Εξόδου (οθόνη) (για τους 6 πρώτους τετραψήφιους αριθμούς)

1000 1001 1002 1003 1004 1005

Άσκηση 4.5

Να γράψετε πρόγραμμα το οποίο να δέχεται δύο ακέραιους αριθμούς X1 και X2 και να τυπώνει όλα τα πολλαπλάσια του 3, μεταξύ των αριθμών X1 και X2, συμπεριλαμβανομένων.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
2 16	Δώσε 2 αριθμούς: Τα πολλαπλάσια του 3 είναι: 3 6 9 12 15

Άσκηση 4.6

Να γράψετε το πρόγραμμα το οποίο να τυπώνει όλους τους τριψήφιους ακέραιους αριθμούς για τους οποίους το πρώτο ψηφίο να είναι άρτιος αριθμός και το τρίτο ψηφίο να είναι περιττός αριθμός.

Παράδειγμα Εξόδου (οθόνη) (για τους 6 πρώτους τριψήφιους αριθμούς)
201 203 205 207 209 211

Άσκηση 4.7

Δίνεται το πιο κάτω πρόγραμμα γραμμένο στη γλώσσα προγραμματισμού C++. Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του προγράμματος. Στη θέση του διαστήματος να χρησιμοποιήσετε το σύμβολο «□».

```
int i, a;
for (i=5; i<=10; i++){
    a = 10 * i;
    cout << a << " " << i << endl;
}
```

Άσκηση 4.8

Να σχεδιάσετε λογικό διάγραμμα και να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό N και να υπολογίζει και να τυπώνει στην οθόνη το άθροισμα όλων των αριθμών από το 1 μέχρι το N.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
10	Δώσε αριθμό: Άθροισμα= 55

Άσκηση 4.9

Να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει το γινόμενο των ζυγών αριθμών από το 10 μέχρι το 20.

Παράδειγμα Εξόδου 1 (οθόνη)
Γινόμενο=9676800

Άσκηση 4.10

Να σχεδιάσετε λογικό διάγραμμα το οποίο να υπολογίζει και να τυπώνει το άθροισμα όλων των τριψηφίων αριθμών των οποίων τουλάχιστον ένα από τα ψηφία τους είναι ο αριθμός 1.

Άσκηση 4.11

Να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει το άθροισμα όλων των τριψηφίων αριθμών των οποίων τουλάχιστον δύο από τα ψηφία τους είναι ο αριθμός 8.

Παράδειγμα Εξόδου (οθόνη)

```
Αθροισμα= 20891
```

Άσκηση 4.12

Να γράψετε πρόγραμμα το οποίο να τυπώνει στην οθόνη όλα τα πολλαπλάσια των αριθμών 17 και 43, στο πεδίο τιμών [1...10000].

Παράδειγμα Εξόδου (οθόνη)

```
Πολλαπλάσια των αριθμών 17 και 43
731 1462 2193 2924 3655 4386 5117 5848 6579 7310 8041 8772
9503
```

Άσκηση 4.13

Να γράψετε πρόγραμμα το οποίο να δέχεται το πλήθος N αριθμών και στη συνέχεια τους N ακέραιους αριθμούς. Ακολουθώντας, να τυπώνει το μήνυμα «Όλοι θετικοί», αν όλοι οι αριθμοί που καταχωρήθηκαν ήταν θετικοί, διαφορετικά να τυπώνει το μήνυμα «Όχι όλοι θετικοί».

Παράδειγμα Εισόδου 1 (πληκτρολόγιο)

```
10
3 6 9 18 27 35 69 2 8 63
```

Παράδειγμα Εξόδου 1 (οθόνη)

```
Δώσε πλήθος αριθμών:
Δώσε 10 αριθμούς:
Όλοι θετικοί
```

Παράδειγμα Εισόδου 2 (πληκτρολόγιο)

```
5
3 -6 9 -18 27
```

Παράδειγμα Εξόδου 2 (οθόνη)

```
Δώσε πλήθος αριθμών:
Δώσε 5 αριθμούς:
Όχι όλοι θετικοί
```

Άσκηση 4.14

Να σχεδιάσετε λογικό διάγραμμα το οποίο να δέχεται 50 ακέραιους αριθμούς και να υπολογίζει και να τυπώνει τον μεγαλύτερο από αυτούς.

Άσκηση 4.15

Η Αριάδνη είχε ως άσκηση στο σχολείο να γράψει ένα πρόγραμμα, που θα τυπώνει όλους τους θετικούς διαιρέτες ενός θετικού ακέραιου αριθμού. Αφού έλυσε το πρόβλημα, πήγε να ξεκουραστεί, αλλά άφησε τον υπολογιστή της ανοικτό. Ο Παντελής, γνωστός για τις

σκανταλιές του, άλλαξε τη σειρά των εντολών του προγράμματος. Βοηθήστε την Αριάδνη, ξαναγράφοντας το πρόγραμμα με τις εντολές στη σωστή σειρά.

```
#include<iostream>
using namespace std;
int main() {
    if(num%i==0)
        for(i=1;i<=num;i++)
            cout<<i<<endl;
    cin>>num;
    return 0;
    int num,i;
}
```

Άσκηση 4.16

Δίνεται το πιο κάτω πρόγραμμα γραμμένο στη γλώσσα προγραμματισμού C++. Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του προγράμματος με τις ακόλουθες τιμές εισόδου:

α) X=5, Y=2, β) X=6, Y=3 και γ) X=4, Y=8

Στη θέση του διαστήματος να χρησιμοποιήσετε το σύμβολο «□».

```
#include<iostream>
#define p 3
using namespace std;
int main() {
    int x,y,i,result=0;
    for (i=1;i<=3;i++) {
        cin>>x>>y;
        if(x%y==0)
            result=result+p*(x+y);
        else
            result=result+p*(x*y);
    }
    cout<<result;
    return 0;
}
```

Άσκηση 4.17

Το πιο κάτω πρόγραμμα δέχεται δέκα ακέραιους αριθμούς και υπολογίζει και τυπώνει τον μεγαλύτερο αριθμό και τη σειρά στην οποία δόθηκε. Σε περίπτωση που υπάρχουν περισσότεροι από έναν μέγιστοι αριθμοί, τυπώνεται ο τελευταίος που δόθηκε. Στο πρόγραμμα υπάρχουν πέντε λογικά ή/και συντακτικά λάθη/παραλείψεις. Να ξαναγράψετε το πρόγραμμα, διορθώνοντας τα λάθη/παραλείψεις.

```
#include<iostream>
using namespace std;
int main() {
    int num, max, maxth;
    cin>>num;
    max=num;
    for (i=1;i<=10;i++) {
        cin>>num;
        if (num>max) {
            max=num;
            maxth=num;
        }
    }
    cout<<"Max="<<max<<" thesi="<<maxth;
return 0;
}
```

Άσκηση 4.18

Το πιο κάτω πρόγραμμα δέχεται N ακέραιους αριθμούς και υπολογίζει και τυπώνει το πλήθος των θετικών, των αρνητικών και των μηδενικών αριθμών. Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του προγράμματος με τις ακόλουθες τιμές εισόδου:

N=5 και α) num=22, β) num=6, γ) num=-3, δ) num=-42 ε) num=0.

Στη θέση του διαστήματος να χρησιμοποιήσετε το σύμβολο «□».

```
#include<iostream>
using namespace std;
int main() {
    int pos=0,neg=0,zeros=0,i,N,num;
    cin>>N;
    for (i=1;i<=N;i++) {
        cin>>num;
        if (num>0)
            pos+=1;
        else if (num<0)
            neg+=1;
        else
            zeros+=1;
    }
    cout<<pos<<endl;
    cout<<neg<<endl;
    cout<<zeros<<endl;
return 0;}
```

Άσκηση 4.19

Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του πιο κάτω προγράμματος. Στη θέση του διαστήματος να χρησιμοποιήσετε το σύμβολο «□».

```
#include<iostream>
using namespace std;
int main() {
    int k = 3, n = 0, m = 0;
    for (int i = 6; i>=0; i-=2) {
        k = k + 1;
        if (k > 4)
            n = k + i;
        else
            n = i - k;
        m = k * n;
    }
    cout << k << " " << n << " " << m << endl;
    return 0;
}
```

Άσκηση 4.20

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται 100 ακέραιους αριθμούς.
- (2) Να υπολογίζει και να τυπώνει τον μέσο όρο των πολλαπλασίων του 8.
- (3) Να υπολογίζει και να τυπώνει τον μέσο όρο των πολλαπλασίων του 12.
- (4) Να υπολογίζει και να τυπώνει τον μεγαλύτερο ζυγό αριθμό από αυτούς που έχουν δοθεί.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 8 αριθμούς)	Παράδειγμα Εξόδου (οθόνη) (για 8 αριθμούς)
3 12 16 24 40 48 56 60	Δώσε αριθμούς: Μέσος όρος πολλαπλασίων 8= 34.00 Μέσος όρος πολλαπλασίων 12= 36.00 Μεγαλύτερος ζυγός= 60

Άσκηση 4.21

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται το επώνυμο και τον βαθμό [1..100] για 50 φοιτητές.
- (2) Να υπολογίζει και να τυπώνει τον μέσο όρο των βαθμών με ακρίβεια δύο δεκαδικών ψηφίων.

- (3) Να υπολογίζει και να τυπώνει το επώνυμο και τον βαθμό του φοιτητή με τον μεγαλύτερο βαθμό.
- (4) Να υπολογίζει και να τυπώνει το επώνυμο και τον βαθμό του φοιτητή με τον μικρότερο βαθμό.
- (5) Να υπολογίζει και να τυπώνει το πλήθος των φοιτητών που πήραν απροβίβαστο βαθμό (βαθμός<50).

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 φοιτητές) Antoniou 68 Georgiou 98 Ioannou 75 Andreou 39 Charalambous 48	Παράδειγμα Εξόδου (οθόνη) (για 5 φοιτητές) Δώσε επώνυμο και βαθμό: Μέσος Όρος= 57.60 Μεγαλύτερος Βαθμός: Georgiou 98 Μικρότερος Βαθμός: Andreou 39 Πλήθος φοιτητών = 2
--	--

Άσκηση 4.22

Σε μια εταιρεία εργάζονται 100 υπάλληλοι, άντρες και γυναίκες. Στην εταιρεία παρουσιάζονται συχνές απουσίες των εργαζομένων για διάφορους λόγους και έτσι η διεύθυνση αποφάσισε να μελετήσει το θέμα. Να γράψετε το οποίο:

- (1) Να δέχεται για κάθε υπάλληλο της εταιρείας, το όνομα, το φύλο ('Μ' για τους άντρες και 'F' για τις γυναίκες), τα χρόνια υπηρεσίας στην εταιρεία και τις ημέρες απουσίας.
- (2) Να υπολογίζει και να τυπώνει το σύνολο των απουσιών των ανδρών και το σύνολο των απουσιών των γυναικών υπαλλήλων.
- (3) Να υπολογίζει και να τυπώνει το σύνολο των απουσιών των ανδρών υπαλλήλων που έχουν υπηρεσία πάνω από 10 χρόνια.
- (4) Να υπολογίζει και να τυπώνει το όνομα και τον αριθμό απουσιών της γυναίκας με τον μεγαλύτερο αριθμό απουσιών.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 υπαλλήλους) Maria F 15 2 George M 11 9 Ioannis M 2 7 Andreas M 23 0 Chara F 1 12	Παράδειγμα Εξόδου (οθόνη) (για 5 υπαλλήλους) Δώσε όνομα, φύλο, χρόνια και μέρες απουσίας: Απουσίες ανδρών= 16 Απουσίες γυναικών= 14 Απουσίες ανδρών με πάνω από 10 χρόνια υπηρεσίας= 9 Μεγαλύτερος αριθμός απουσιών: Chara 12
---	---

Άσκηση 4.23

Στην παγκόσμια Ολυμπιάδα Πληροφορικής συμμετέχουν 180 χώρες. Τα ονόματα των 180 χωρών και η βαθμολογία (1 μέχρι και το 100 συμπεριλαμβανομένων) της κάθε χώρας θα δίνονται από το χρήστη. Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται το όνομα της χώρας και τη βαθμολογία που πήρε.
- (2) Να υπολογίζει και να τυπώνει το πλήθος των χωρών που πέτυχαν βαθμολογία μεγαλύτερη από 80.
- (3) Να υπολογίζει και να τυπώνει τον μέσο όρο των χωρών που πήραν βαθμολογία 1-45.
- (4) Να υπολογίζει και να τυπώνει το όνομα της χώρας με την ψηλότερη βαθμολογία.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 χώρες) Cyprus 98 Greece 65 Italy 43 Spain 45 UK 97	Παράδειγμα Εξόδου (οθόνη) (για 5 χώρες) Δώσε όνομα και βαθμολογία: Πλήθος χωρών= 2 Μέσος Όρος= 44.00 Ψηλότερη βαθμολογία: Cyprus 98
---	---

Άσκηση 4.24

Σε μια παγκόσμια αθλητική συνάντηση συμμετέχουν αθλητές από διάφορες χώρες. Από την αντιπροσωπεία των αθλητών μιας χώρας οι 100 διακρίθηκαν εξασφαλίζοντας πρώτη μέχρι δέκατη θέση σε κάποιο αγώνισμα. Όσοι εξασφάλισαν πρώτη, δεύτερη ή τρίτη θέση πήραν μετάλλιο. Επιπρόσθετα, η αθλητική ομοσπονδία της χώρας αποφάσισε να δώσει επίδομα σε όσους αθλητές διακρίθηκαν σε οποιοδήποτε αγώνισμα κατά τη συνάντηση αυτή, σύμφωνα με τον πιο κάτω πίνακα.

Θέση	Επίδομα
1 ^η	50000
2 ^η	40000
3 ^η	30000
4 ^η – 6 ^η	2000
7 ^η – 8 ^η	1000
9 ^η – 10 ^η	500

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται για τον κάθε αθλητή την κατάταξή του και να ελέγχει ότι ο αριθμός που δόθηκε είναι 1 μέχρι 10. Σε αντίθετη περίπτωση να δίνεται το μήνυμα «Λάθος Κατάταξη» και να ξαναζητά την κατάταξη.
- (2) Να υπολογίζει και να τυπώνει τον συνολικό αριθμό των μεταλλίων που εξασφάλισε η χώρα.
- (3) Να υπολογίζει και να τυπώνει το συνολικό ποσό της επιδότησης που θα πληρώσει η ομοσπονδία.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 αθλητές) 1 5 3 4 9	Παράδειγμα Εξόδου (οθόνη) (για 5 αθλητές) Δώσε κατάταξη αθλητή: Μετάλλια= 2 Ποσό επιδότησης=84500 ευρώ
---	--

Άσκηση 4.25

Ένα συσκευαστήριο φρούτων απασχολεί 15 εργάτες. Η διεύθυνση αποφάσισε να τους δώσει κίνητρα σε μια προσπάθεια να αυξήσει την παραγωγή. Κάθε εργάτης θα πρέπει να συσκευάζει τουλάχιστον 500 κιβώτια φρούτων την εβδομάδα. Σε περίπτωση, όμως που συσκευάσει περισσότερα από 500 κιβώτια σε μια εβδομάδα, τότε θα πάρει επιπρόσθετο επίδομα σύμφωνα με τον πιο κάτω πίνακα:

Επιπρόσθετα κιβώτια (>500)	Επίδομα (ανά επιπρόσθετο κιβώτιο)
1-15	1ευρώ
16-30	2 ευρώ
>30	3 ευρώ

Επεξήγηση: Εάν ένας εργάτης συσκευάσει 520 κιβώτια φρούτων σε μια εβδομάδα, τότε θα πάρει επίδομα 25 ευρώ.

Δηλαδή, για τα πρώτα 15 επιπρόσθετα κιβώτια θα πάρει επίδομα 15 ευρώ ($15 * 1$) και για τα υπόλοιπα 5 επιπρόσθετα κιβώτια θα πάρει επίδομα 10 ευρώ ($5 * 2$).

Να γράψετε πρόγραμμα το οποίο να:

- (1) Να δέχεται το όνομα του κάθε εργάτη και τον αριθμό των κιβωτίων που συσκευάσε σε μια εβδομάδα.
- (2) Να υπολογίζει και να τυπώνει για τον κάθε εργάτη το όνομά του και το επιπρόσθετο επίδομα που θα πάρει. Σε περίπτωση που συσκευάσει ακριβώς 500 κιβώτια να τυπώνει το όνομα του και το μήνυμα «Μηδέν επίδομα».
- (3) Να υπολογίζει και να τυπώνει το σύνολο των κιβωτίων που συσκευάσαν όλοι οι εργάτες μαζί.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 εργάτες) Maria: 520 Leonidas: 560 George: 500 Elena: 580 Antonis: 515
--

Παράδειγμα Εξόδου (οθόνη) (για 5 εργάτες)

Δώσε όνομα και αριθμό κιβωτίων:
 Maria: 25 ευρώ
 Leonidas: 135 ευρώ
 George: Μηδέν Επίδομα
 Elena: 195 ευρώ
 Antonis: 15 ευρώ
 Σύνολο κιβωτίων= 2675

Άσκηση 4.26

Ο όρος παραγοντικό (factorial) ακεραίου θετικού αριθμού N είναι το γινόμενο όλων των ακεραίων αριθμών από το 1 έως το N . Για παράδειγμα το παραγοντικό του αριθμού 5 ισούται με $1*2*3*4*5=120$, ενώ το παραγοντικό του 6 ισούται με $1*2*3*4*5*6=720$, κτλ. Το παραγοντικό του 0 ισούται με 1. Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο θετικό αριθμό και να υπολογίζει και να τυπώνει το παραγοντικό του.

Παράδειγμα Εισόδου (πληκτρολόγιο)

5

Παράδειγμα Εξόδου (οθόνη)

Δώσε αριθμό:
 Παραγοντικό= 120

Άσκηση 4.27

Το νέο θέατρο του ΘΟΚ θα διαθέτει 50 καθίσματα στην πρώτη σειρά και σε κάθε επόμενη σειρά, από τις 30 συνολικά, θα υπάρχει αύξηση καθισμάτων κατά 10. Να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει τον αριθμό των καθισμάτων της τελευταίας σειράς, καθώς και τη συνολική χωρητικότητα καθισμάτων του θεάτρου.

Παράδειγμα Εξόδου (οθόνη)

Αριθμός καθισμάτων τελευταίας σειράς= 350
 Σύνολο καθισμάτων= 5850

Άσκηση 4.28

Μια ακολουθία έχει σαν πρώτο όρο το 5 και κάθε επόμενος όρος της ακολουθίας υπολογίζεται προσθέτοντας τον αριθμό 3 στον προηγούμενο όρο. Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό N και να τυπώνει τους N πρώτους όρους της ακολουθίας.

Παράδειγμα Εισόδου (πληκτρολόγιο)

5

Παράδειγμα Εξόδου (οθόνη)

Δώσε αριθμό:
 Οι 5 όροι της ακολουθίας είναι:
 5 8 11 14 17

Άσκηση 4.29

Να γράψετε πρόγραμμα το οποίο να τυπώνει το αποτέλεσμα του παραδείγματος εξόδου, χρησιμοποιώντας τη δομή επανάληψης.

Παράδειγμα Εξόδου (οθόνη)

```

1
12
123
1234
12345
123456
1234567
12345678
123456789

```

Άσκηση 4.30

Ένας θετικός ακέραιος αριθμός ονομάζεται «Τέλειος», αν το άθροισμα των θετικών αριθμών που τον διαιρούν (διαιρέτες), πλην του εαυτού του, είναι ίσο με τον αριθμό. Για παράδειγμα, το 6 είναι τέλειος γιατί έχει ως διαιρέτες τους 1, 2 και 3 και ισχύει $1+2+3=6$. Να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει όλους τους τέλειους αριθμούς που ανήκουν στο πεδίο [2..2000].

Παράδειγμα Εξόδου (οθόνη)

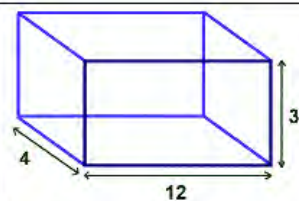
```

Τέλειοι αριθμοί [2..2000]
6
28
496
8128

```

Άσκηση 4.31

Να γράψετε πρόγραμμα το οποίο να δέχεται τον όγκο V ($1 \leq V \leq 10^7$) ενός κουτιού και να υπολογίζει και να τυπώνει το πλήθος των πιθανών συνδυασμών διαστάσεων του.



Λίγη γεωμετρία:

Όγκος κουτιού: μήκος x πλάτος x ύψος = $12 \times 4 \times 3 = 144 \text{cm}^3$

**Παράδειγμα Εισόδου
(πληκτρολόγιο)**

```

10

```

Παράδειγμα Εξόδου (οθόνη)

```

Δώσε τον όγκο:
Πιθανοί τρόποι= 2

```

Επεξήγηση: Αν ο όγκος του κουτιού είναι 10 cm^3 , τότε οι πιθανές διαστάσεις του κουτιού μπορεί να είναι: $1 \times 1 \times 10$ και $1 \times 2 \times 5$

Δηλαδή 2 πιθανοί τρόποι.

Σημείωση: Οι διαστάσεις $10 \times 1 \times 1$ και $5 \times 2 \times 1$, όπου εναλλάσσονται οι τιμές, δεν πρέπει να υπολογίζονται.

Άσκηση 4.32

Ένας αριθμός καλείται πρώτος, αν διαιρείται μόνο από τον αριθμό 1 και τον εαυτό του. Να γράψετε πρόγραμμα, το οποίο να δέχεται έναν ακέραιο αριθμό N και να υπολογίζει και να τυπώνει όλους τους πρώτους αριθμούς από το 1 έως και το N.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
25	Δώσε έναν ακέραιο αριθμό: Πρώτοι αριθμοί: 1 3 5 7 11 13 17 19 23

Άσκηση 4.33

Να μετατρέψετε το πιο κάτω πρόγραμμα, χρησιμοποιώντας τη δομή επανάληψης while.

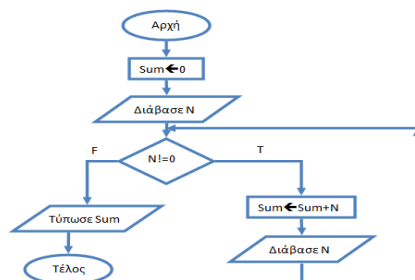
```
#include<iostream>
using namespace std;
int main() {
    for(int i=10;i>=0;i--)
        cout<<i<<endl;
    return 0;
}
```

Άσκηση 4.34

- (1) Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του πιο κάτω λογικού διαγράμματος με τιμές εισόδου:

A) $N=3, 1, 10, -2, 0$

B) $N=0$



- (2) Να μετατρέψετε το πιο πάνω λογικό διάγραμμα σε πρόγραμμα.

Άσκηση 4.35

Δίνεται το πιο κάτω πρόγραμμα. Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του πιο κάτω προγράμματος.

```
#include<iostream>
using namespace std;
int main() {
    int x=10,y=1,z=0;
    while (x%2==0 && x>0) {
        x=x-y*2;
        if(y>x)
            z=z+x+y;
        y=y+2;
    }
    cout<<"X="<<x<<endl;
    cout<<"Y="<<y<<endl;
    cout<<"Z="<<z<<endl;
}
```

Άσκηση 4.36

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται από τον χρήστη αγνώστου πλήθους ακέραιους αριθμούς. Το πρόγραμμα να σταματά να δέχεται αριθμούς όταν δοθεί ο αριθμός 999.
- (2) Να υπολογίζει και να τυπώνει τον μεγαλύτερο αριθμό από τους αριθμούς που δόθηκαν.
- (3) Να υπολογίζει και να τυπώνει τον μέσο όρο των αριθμών που δόθηκαν.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
25 33 12 -10 999	Δώσε ακέραιους αριθμούς: Μεγαλύτερος αριθμός= 33 Μέσος όρος= 15.00

**Άσκηση 4.37**

Να γράψετε πρόγραμμα το οποίο να δέχεται από τον χρήστη αριθμούς από το 1 μέχρι το 9999. Στην περίπτωση που ο αριθμός είναι εκτός ορίων, να τυπώνει το μήνυμα «Λανθασμένος αριθμός» και δέχεται άλλον αριθμό. Το πρόγραμμα σταματά να δέχεται αριθμούς όταν δοθεί ο αριθμός 0 (μηδέν). Στη συνέχεια, να υπολογίζει και να τυπώνει το πλήθος των μονοψήφιων, των διψήφιων, των τριψήφιων και των τετραψήφιων αριθμών που έχουν δοθεί.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 25 3 12 999 4356 0	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε ακέραιους αριθμούς: Πλήθος μονοψήφιων αριθμών= 1 Πλήθος διψήφιων αριθμών= 2 Πλήθος τριψήφιων αριθμών= 1 Πλήθος τετραψήφιων αριθμών= 1
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 25 3 12 12000 999 4356 0	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε ακέραιους αριθμούς: Λανθασμένος αριθμός Πλήθος μονοψήφιων αριθμών= 1 Πλήθος διψήφιων αριθμών= 2 Πλήθος τριψήφιων αριθμών= 1 Πλήθος τετραψήφιων αριθμών= 1

Άσκηση 4.38

Να γράψετε πρόγραμμα το οποίο να δέχεται μόνο τριψήφιους αριθμούς μέχρι ο χρήστης να δώσει τον αριθμό 999. Αν δοθεί αριθμός ο οποίος δεν είναι τριψήφιος, να εμφανίζεται το μήνυμα «Λανθασμένος αριθμός» και να ξαναζητά αριθμό. Στο τέλος, το πρόγραμμα να τυπώνει πόσοι από τους τριψήφιους αριθμούς που έχουν δοθεί είναι μεγαλύτεροι του 500.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 321 909 521 345 547 999	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε αριθμούς: Πλήθος αριθμών= 3
---	---

Άσκηση 4.39

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο N ($1 \leq N \leq 729$) και να υπολογίζει τον μικρότερο τριψήφιο ακέραιο του οποίου το γινόμενο των ψηφίων ισούται με N . Αν δεν υπάρχει τέτοιος αριθμός να εμφανίζεται ο αριθμός -1.

Παράδειγμα Εισόδου (πληκτρολόγιο) 5	Παράδειγμα Εξόδου (οθόνη) Δώσε αριθμό: Μικρότερος τριψήφιος= 104
---	---

Άσκηση 4.40

Ένα ντεπόζιτο νερού χωρητικότητας 2000 λίτρων περιέχει 1000 λίτρα νερού. Καθημερινά το συμβούλιο υδατοπρομήθειας προσθέτει στο ντεπόζιτο x λίτρα νερού ενώ ταυτόχρονα την ίδια μέρα υπάρχει κατανάλωση y λίτρων νερού. Να σχεδιάσετε λογικό διάγραμμα το οποίο να δέχεται για κάθε μέρα τις τιμές των x και y ($x > y$) και να τυπώνει το πλήθος των ημερών που θα χρειαστεί, για να γεμίσει εντελώς το ντεπόζιτο.

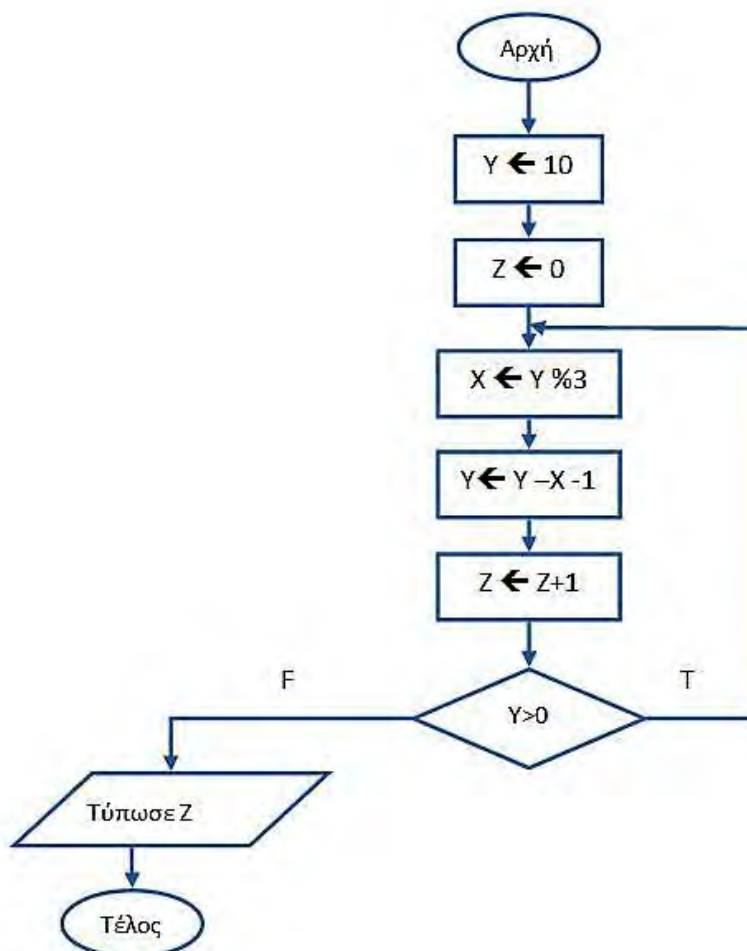
Άσκηση 4.41

Να μετατρέψετε το πιο κάτω πρόγραμμα στην επαναληπτική δομή while.

```
#include<iostream>
using namespace std;
int main() {
    for (int i = 1; i <= 8; i++){
        for (int j = 1; j <= i; j++){
            cout << "*";
        }
        cout << endl;
    }
    return 0;
}
```

Άσκηση 4.42

- (1) Χρησιμοποιώντας τη μέθοδο της προκαταρκτικής εκτέλεσης να παρουσιάσετε τα αποτελέσματα του πιο κάτω λογικού διαγράμματος.



- (2) Να μετατρέψετε το πιο πάνω λογικό διάγραμμα σε πρόγραμμα.

Άσκηση 4.43

Ένα ασανσέρ έχει ανώτατο όριο βάρους τα 600 κιλά ή τα 8 άτομα. Να γράψετε πρόγραμμα το οποίο να δέχεται το βάρος κάθε ατόμου που εισέρχεται στο ασανσέρ, μέχρι που να μην μπορεί να σηκώσει άλλο βάρος ή να φτάσει το όριο των 8 ατόμων. Στο τέλος, να τυπώνει το συνολικό βάρος και τον αριθμό των ατόμων που έχουν εισέλθει.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
58 69 88 92 53 23 72 59	Δώσε βάρος ατόμου: Πλήθος ατόμων=8 Συνολικό Βάρος=514

Άσκηση 4.44

Να μετατρέψετε το πιο κάτω πρόγραμμα, χρησιμοποιώντας τη δομή επανάληψης while και τη δομή επανάληψης for.

```
#include<iostream>
#define p 0.02
using namespace std;
int main() {
    float x,y=1;
    int z=0;
    cin>>x;
    do{
        y=y+y*p;
        z++;
    }while(y<x);
    cout<<z;
    return 0;
}
```

Άσκηση 4.45

Σε ένα διαγωνισμό λαμβάνουν μέρος 200 μαθητές. Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται από τον χρήστη το όνομα και την βαθμολογία των διαγωνιζομένων. Αν η βαθμολογία δεν είναι μεταξύ 1 και 100 συμπεριλαμβανομένων, να ζητά από τον χρήστη να την ξαναδώσει.
- (2) Να υπολογίζει και να τυπώνει τον μέσο όρο των βαθμών όλων των διαγωνιζομένων.
- (3) Να υπολογίζει και να τυπώνει το πλήθος των μαθητών με βαθμό πάνω από 50.
- (4) Να υπολογίζει τη μεγαλύτερη βαθμολογία και να τυπώνει το όνομα του μαθητή που την πέτυχε.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 μαθητές) Antonis 68 George 98 Ioanna 75 Andreas 39 Dimitra 48	Παράδειγμα Εξόδου (οθόνη) (για 5 μαθητές) Δώσε όνομα και βαθμό για κάθε μαθητή: Μέσος Όρος= 65.60 Πλήθος μαθητών = 2 Μεγαλύτερος Βαθμός: George 98
--	---

Άσκηση 4.46

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται από τον χρήστη έναν ακέραιο αριθμό. Ο αριθμός ανήκει στο διάστημα [-100...100], διαφορετικά να εμφανίζεται μήνυμα σφάλματος και να ζητά να ξαναδοθεί ο αριθμός
- (2) Να υπολογίζει και να τυπώνει το πλήθος των άρτιων αριθμών.
- (3) Να υπολογίζει και να τυπώνει τον μέσο όρο των θετικών αριθμών. Στην περίπτωση που δεν έχουν δοθεί θετικοί αριθμοί να εμφανίζεται το μήνυμα «Δεν δόθηκαν θετικοί αριθμοί»

Το πρόγραμμα σταματά να δέχεται αριθμούς όταν δοθεί ο αριθμός 0.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 αριθμούς) -99 20 58 45 36 0	Παράδειγμα Εξόδου (οθόνη) (για 5 αριθμούς) Δώσε αριθμούς: Πλήθος άρτιων = 3 Μέσος όρος θετικών=39.75
--	--

Άσκηση 4.47

Μία εταιρεία δημοσκοπήσεων θέτει ένα ερώτημα σε δείγμα 1000 πολιτών. Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται το φύλο του πολίτη (1. Άντρας, 2. Γυναίκα) και να ελέγχει κατά πόσο έχει την ορθή τιμή. Στην αντίθετη περίπτωση να τυπώνει το μήνυμα «Λάθος. Δώσε ξανά το φύλο:» και να δέχεται ξανά το φύλο του πολίτη.
- (2) Να δέχεται την απάντηση στο ερώτημα, η οποία μπορεί να είναι «1. ΝΑΙ», «2. ΟΧΙ», «3. ΔΕΝ ΞΕΡΩ» και να ελέγχει κατά πόσο έχει την ορθή τιμή. Στην αντίθετη περίπτωση να τυπώνει το μήνυμα «Λάθος. Δώσε ξανά την απάντηση:» και να δέχεται ξανά την απάντηση του πολίτη.
- (3) Να υπολογίζει και να τυπώνει το πλήθος και το ποσοστό των ατόμων που απάντησαν «ΝΑΙ».
- (4) Να υπολογίζει το ποσοστό των ανδρών και το ποσοστό των γυναικών που απάντησαν «ΝΑΙ».

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 πολίτες) 1 1 1 2 2 1 2 3 1 2	Παράδειγμα Εξόδου (οθόνη) (για 5 πολίτες) Δώσε φύλο και απάντηση: Πλήθος ατόμων = 40.00% Ποσοστό αντρών= 33.33 % Ποσοστό γυναικών= 50.00%
---	--

Άσκηση 4.48

Σε ένα διαγωνισμό υπάρχουν συνολικά 100 διαθέσιμες συμμετοχές σε τρεις κατηγορίες: K1 για άντρες, K2 για γυναίκες και K3 για παιδιά κάτω από 17 χρονών. Οι υποψήφιοι για συμμετοχή δηλώνουν την ηλικία (σε ακέραιο αριθμό), το όνομα και το φύλο τους (M άντρες, F γυναίκες).

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται τα δεδομένα για τον κάθε υποψήφιο και να τερματίζει την είσοδο δεδομένων όταν συμπληρωθεί ο μέγιστος αριθμός συμμετοχών ή όταν εισαχθεί ηλικία ίση με μηδέν.
- (2) Να υπολογίζει και να τυπώνει την κατηγορία για τον κάθε συμμετέχοντα.
- (3) Να υπολογίζει και να τυπώνει το πλήθος των συμμετεχόντων σε κάθε κατηγορία.
- (4) Να τυπώνει το όνομα και την ηλικία του νεαρότερου συμμετέχοντα στην κατηγορία K3. Σε περίπτωση που δεν υπάρχουν άτομα στην κατηγορία αυτή να τυπώνει το κατάλληλο μήνυμα.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 συμμετοχές) 16 Maria F 23 Aggelos M 17 Marilena F 28 Andreas M 45 George M 0	Παράδειγμα Εξόδου (οθόνη) (για 5 συμμετοχές) Δώσε ηλικία, όνομα και φύλο: K3 K1 K2 K1 K1 Πλήθος ατόμων στην κατηγορία 1= 3 Πλήθος ατόμων στην κατηγορία 2= 1 Πλήθος ατόμων στην κατηγορία 3= 1 Νεαρότερος στην κατηγορία 3 Maria 16
--	---

Άσκηση 4.49

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται από το χρήστη το βάρος κάθε ατόμου μιας ομάδας. Ως τέλος της εισαγωγής των δεδομένων χρησιμοποιείται ο αριθμός 0.
- (2) Να ελέγχει ότι το βάρος του κάθε ατόμου είναι αριθμός θετικός, διαφορετικά να ζητά από το χρήστη να δώσει ξανά το βάρος του.
- (3) Να υπολογίζει και να τυπώνει τον μέσο όρο βάρους όλων των ατόμων της ομάδας.
- (4) Να εμφανίζει το ανάλογο μήνυμα:
 - i. «Η ομάδα ανήκει στην ΚΑΤΗΓΟΡΙΑ Α» αν ο μέσος όρος βάρους της ομάδας είναι μικρότερος ή ίσος του 50.
 - ii. «Η ομάδα ανήκει στην ΚΑΤΗΓΟΡΙΑ Β» αν ο μέσος όρος βάρους της ομάδας είναι μεγαλύτερος του 50 και μικρότερος ή ίσος του 80.
 - iii. «Η ομάδα ανήκει στην ΚΑΤΗΓΟΡΙΑ Γ» αν ο μέσος όρος βάρους της ομάδας είναι μεγαλύτερος από το 80.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 άτομα) 82.5 68.7 75.8 81 62.35 0	Παράδειγμα Εξόδου (οθόνη) (για 5 άτομα) Δώσε βάρος: Μέσος Όρος Βάρους = 62.07 Η ομάδα ανήκει στην κατηγορία Β
--	---

Άσκηση 4.50

Σε μία διαγνωστική αξιολόγηση στο πλαίσιο του μαθήματος της Πληροφορικής, οι μαθητές θα κληθούν να απαντήσουν στην ερώτηση «Πόσο ικανοποιημένοι είστε από την επίδοσή σας στο μάθημα της Πληροφορικής;» με τις πιο κάτω επιλογές:

1. ΠΑΡΑ ΠΟΛΥ
2. ΑΡΚΕΤΑ
3. ΛΙΓΟ
4. ΚΑΘΟΛΟΥ
5. ΕΞΟΔΟΣ

Να γράψετε πρόγραμμα το οποίο να τυπώνει τις πιο πάνω επιλογές στην οθόνη και να δέχεται την επιλογή C ($1 \leq C \leq 5$) του κάθε μαθητή. Σε περίπτωση λανθασμένης εισαγωγής δεδομένων, να τυπώνει το μήνυμα «Λανθασμένη επιλογή» και να ζητά από τον μαθητή να ξαναδώσει επιλογή. Το πρόγραμμα να τερματίζει όταν δοθεί ως επιλογή ο αριθμός 5.

Το πρόγραμμα να υπολογίζει και να τυπώνει:

- (1) Το πλήθος των μαθητών που συμμετείχαν στην αξιολόγηση.
- (2) Το πλήθος των μαθητών που δεν απάντησαν «ΛΙΓΟ».
- (3) Το ποσοστό των μαθητών που απάντησαν «ΠΑΡΑ ΠΟΛΥ».
- (4) Το πλήθος των μαθητών που απάντησαν «ΑΡΚΕΤΑ» ή «ΛΙΓΟ».

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 μαθητές)

1
1
3
2
1
5

Παράδειγμα Εξόδου (οθόνη) (για 5 μαθητές)

Πόσο ικανοποιημένοι είστε από την επίδοσή σας στο μάθημα της Πληροφορικής;

1. ΠΑΡΑ ΠΟΛΥ

2. ΑΡΚΕΤΑ

3. ΛΙΓΟ

4. ΚΑΘΟΛΟΥ

5. ΕΞΟΔΟΣ

Δώσε επιλογή:

Πλήθος μαθητών= 5

Πλήθος μαθητών που δεν απάντησαν λίγο= 4

Ποσοστό μαθητών που απάντησαν ΠΑΡΑ ΠΟΛΥ = 60.00%

Πλήθος μαθητών που απάντησαν ΑΡΚΕΤΑ ή ΛΙΓΟ = 2

Άσκηση 4.51

Η Ευρωπαϊκή Ένωση στην προσπάθειά της να αντιμετωπίσει το μεγάλο πρόβλημα του προσφυγικού, καθόρισε για κάθε χώρα μέλος της ένα αριθμό προσφύγων που πρέπει να φιλοξενήσει. Μια από αυτές τις χώρες αποφάσισε να δίνει επίδομα προς τους πρόσφυγες που θα φιλοξενήσει, βάσει της κατηγορίας που τους κατατάσσει, όπως φαίνεται στον πιο κάτω πίνακα:

Κατηγορία	Επίδομα (σε ευρώ)
A	500
B	300
Γ	250

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται από τον χρήστη τον αριθμό διαβατηρίου (να υποθέσετε ότι αποτελείται από 8 ψηφία, αριθμούς ή/και χαρακτήρες), το φύλο (Μ για άντρα, F για γυναίκα), την ηλικία (πραγματικός αριθμός) και την κατηγορία στην οποία ανήκει κάθε πρόσφυγας. Να γίνεται έλεγχος της ηλικίας αν έχει δοθεί αρνητικός αριθμός ή 0. Σε αυτή την περίπτωση να παρουσιάζεται στην οθόνη το μήνυμα «Λάθος ηλικία» και να ζητείται να δοθεί ξανά η ηλικία. Τα υπόλοιπα δεδομένα να θεωρήσετε ότι δίνονται ορθά και δεν χρειάζονται οποιοδήποτε έλεγχο. Το

πρόγραμμα να σταματά να δέχεται δεδομένα όταν δοθεί ως αριθμός διαβατηρίου 'M0000000' ή όταν ο αριθμός των προσφύγων υπερβεί τις 30000.

(2) Να υπολογίζει και να τυπώνει τα πιο κάτω:

- i. το πλήθος όλων των προσφύγων που εισήλθαν στη χώρα και το συνολικό ποσό που διέθεσε η συγκεκριμένη χώρα για επιδόματα για όλους τους πρόσφυγες.
- ii. το πλήθος των γυναικών προσφύγων που ανήκουν στην κατηγορία A και είναι 16 χρονών και άνω.
- iii. το πλήθος των παιδιών κάτω των 12 χρόνων που ανήκουν στην κατηγορία A, καθώς επίσης τον αριθμό διαβατηρίου και την ηλικία του νεαρότερου από αυτούς. Να υποθέσετε ότι βρέθηκε μόνο ένα παιδί με τη μικρότερη ηλικία.
- iv. το σύνολο των χρημάτων που διέθεσε η χώρα για επιδόματα, για όλους τους πρόσφυγες που ανήκουν στην κατηγορία Γ.
- v. τον μέσο όρο ηλικίας των προσφύγων που ανήκουν στην κατηγορία B. Να υποθέσετε ότι βρέθηκε τουλάχιστον ένας πρόσφυγας.

(Παγκύπριες Εξετάσεις 2016)

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 πρόσφυγες)

M0000010 M 19 Γ
M0010101 F 54 A
M2510002 F 11 A
M3120252 F 65 B
M4558525 M 47 B
M0000000

Παράδειγμα Εξόδου (οθόνη) (για 5 πρόσφυγες)

Δώσε αρ. διαβατηρίου, φύλο, ηλικία και κατηγορία:
Πλήθος προσφύγων = 5
Συνολικό Ποσό= 1750 ευρώ
Πλήθος γυναικών προσφύγων κατηγορίας A= 1
Πλήθος παιδιών κάτω των 12 κατηγορίας A =1
Νεαρότερο παιδί κατηγορίας A:
M2510002 11
Σύνολο χρημάτων προσφύγων κατηγορίας Γ=250
Μέσος όρος ηλικίας προσφύγων κατηγορίας B= 56.00

Άσκηση 4.52

Μια επαγγελματική ομάδα καλαθοσφαίρας πρέπει να έχει στο δυναμικό της επαγγελματίες παίκτες, ντόπιους και ξένους. Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται από τον χρήστη το όνομα κάθε παίκτη, την προέλευσή του ('L' για ντόπιος, 'F' για ξένος), την ηλικία του και το ύψος του σε εκατοστόμετρα. Ο χρήστης να ερωτάται κάθε φορά κατά πόσον υπάρχει και άλλος παίκτης. Η επανάληψη να τερματίζεται όταν ο χρήστης απαντήσει «No» στην ερώτηση «Υπάρχει άλλος παίκτης; (Yes/No)». Να θεωρήσετε ότι όλα τα δεδομένα δίνονται

σωστά και δεν χρειάζεται οποιοσδήποτε έλεγχος. Να θεωρήσετε επίσης ότι δίνεται τουλάχιστον ένας ντόπιος και ένας ξένος παίκτης.

- (2) Να υπολογίζει και να τυπώνει τον συνολικό αριθμό των ξένων παικτών, καθώς και τον μέσο όρο της ηλικίας τους.
- (3) Να υπολογίζει και να τυπώνει τον συνολικό αριθμό των ντόπιων παικτών που έχουν ηλικία μικρότερη των 20 χρονών.
- (4) Να υπολογίζει και να τυπώνει το όνομα και το ύψος του ψηλότερου παίκτη της ομάδας.
- (5) Να υπολογίζει και να τυπώνει το ποσοστό (%) των παικτών με ύψος μεγαλύτερο από 205 εκατοστόμετρα.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 παίκτες)

```
Antonis L 19 187
Yes
Marios L 21 191
Yes
Nicholas F 21 189
Yes
Lenos L 24 185
Yes
Ryan F 22 206
No
```

Παράδειγμα Εξόδου (οθόνη) (για 5 παίκτες)

```
Δώσε όνομα, προέλευση, ηλικία και ύψος:
Υπάρχει άλλος παίκτης;
Υπάρχει άλλος παίκτης;
Υπάρχει άλλος παίκτης;
Υπάρχει άλλος παίκτης;
Υπάρχει άλλος παίκτης;
Ξένοι Παίκτες= 2
Μέσος όρος ηλικίας ξένων παικτών= 21.50
Πλήθος ντόπιων παικτών με ηλικία <20= 1
Ψηλότερος παίκτης: Ryan 206 cm
Ποσοστό= 20.00%
```



Άσκηση 4.53

Να γράψετε πρόγραμμα το οποίο να δέχεται δύο ακέραιους θετικούς αριθμούς και να υπολογίζει και να τυπώνει το ελάχιστο κοινό πολλαπλάσιό τους.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο)

```
4 6
```

Παράδειγμα Εξόδου 1 (οθόνη)

```
Δώσε 2 ακέραιους αριθμούς:
Ελάχιστο κοινό πολλαπλάσιο= 12
```

Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 3 4	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε 2 ακέραιους αριθμούς: Ελάχιστο κοινό πολλαπλάσιο= 12
---	--



Άσκηση 4.54

Να γράψετε πρόγραμμα το οποίο να δέχεται δύο ακέραιους αριθμούς και να υπολογίζει και να τυπώνει τον μέγιστο κοινό διαιρέτη τους.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 4 6	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε 2 ακέραιους αριθμούς: Μέγιστος Κοινός Διαιρέτης= 2
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 12 8	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε 2 ακέραιους αριθμούς: Μέγιστος Κοινός Διαιρέτης= 4

Άσκηση 4.55

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να υπολογίζει και να τυπώνει το πλήθος των ψηφίων του, καθώς επίσης και το άθροισμα των ψηφίων του.

Παράδειγμα Εισόδου (πληκτρολόγιο) 123	Παράδειγμα Εξόδου (οθόνη) Δώσε ακέραιο αριθμό: Πλήθος ψηφίων= 3 Άθροισμα ψηφίων= 6
---	--

Άσκηση 4.56

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να τυπώνει τον αντίστροφο αριθμό του.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 123	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε ακέραιο αριθμό: Αντίστροφος= 321
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 1234	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε ακέραιο αριθμό: Αντίστροφος= 4321



Άσκηση 4.57

Ο θυρωρός ενός ουρανοξύστη θέλει να γνωρίζει, στο τέλος της ημέρας, σε ποιον όροφο έχει σταματήσει ο ανελκυστήρας, καθώς επίσης και την απόσταση (σε μέτρα) που έχει διανύσει αν κάθε όροφος έχει ύψος 3 μέτρα. Ο ανελκυστήρας ξεκινά καθημερινά από το ισόγειο (όροφος 0) και μετακινείται είτε προς τα πάνω είτε προς τα κάτω. Ο ουρανοξύστης έχει 75

ορόφους και 5 υπόγειους ορόφους, που χρησιμεύουν ως χώροι στάθμευσης. Δίνονται ως δεδομένα εισόδου οι μετακινήσεις του ανελκυστήρα με τη μορφή θετικών ή αρνητικών ακεραίων, που υποδεικνύουν μετακίνηση ορόφων προς τα πάνω ή προς τα κάτω, αντίστοιχα. Πρέπει να γίνεται έλεγχος, ώστε οι όροφοι να μην υπερβαίνουν τα όρια του ουρανοξύστη. Όταν δοθεί ο αριθμός μηδέν, σημαίνει ότι ο ανελκυστήρας έχει σταματήσει τη λειτουργία του. Να γράψετε πρόγραμμα το οποίο να δέχεται τις μετακινήσεις του ανελκυστήρα και να τυπώνει τον όροφο που έχει σταματήσει και την απόσταση που έχει διανύσει σε μέτρα.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
23 -5 31 25 -40 23 5 -10 0	Δώσε μετακινήσεις ανελκυστήρα: Όροφος= 52 Απόσταση= 486 μέτρα



Άσκηση 4.58

Ο N -οστός τριγωνικός αριθμός T_N είναι το άθροισμα των πρώτων N ακεραίων, π.χ. $T_5=1+2+3+4+5=15$. Οι πρώτοι πέντε όροι της σειράς των τριγωνικών αριθμών είναι 1, 3, 6, 10, 15. Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό N και να τυπώνει τους N πρώτους τριγωνικούς αριθμούς στην οθόνη.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
10	Δώσε ακέραιο αριθμό: Οι 10 τριγωνικοί αριθμοί είναι: 1 3 6 10 15 21 28 36 45 55



Άσκηση 4.59

Η σειρά Fibonacci είναι η εξής:

0, 1, 1, 2, 3, 5, 8, 13, 21...

Εξ ορισμού, οι δύο πρώτοι αριθμοί της σειράς είναι το 0 και το 1, ενώ καθένας από τους επόμενους αριθμούς ισούται με το άθροισμα των δύο προηγούμενων. Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό N ($N \leq 45$) και να τυπώνει τους πρώτους N αριθμούς της σειράς Fibonacci στην οθόνη.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
10	Δώσε ακέραιο αριθμό: Οι 10 αριθμούς της σειράς Fibonacci είναι: 0 1 1 2 3 5 8 13 21 34

Άσκηση 4.60

Να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει τον μικρότερο ακέραιο, ο οποίος θα είναι περιττός και το γινόμενο των ψηφίων του θα είναι διπλάσιο από το άθροισμά τους.

Παράδειγμα Εξόδου (οθόνη)

Ακέραιος αριθμός: 63

**Άσκηση 4.61**

Να γράψετε πρόγραμμα το οποίο, χρησιμοποιώντας μόνο πέντε συνεχόμενα ψηφία από το σύνολο $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, να υπολογίζει και να τυπώνει το πλήθος των τετραψήφιων αριθμών που μπορούν να σχηματιστούν χωρίς επανάληψη ψηφίου. Δηλαδή, ο αριθμός 1234 είναι έγκυρος, ενώ ο 1233 άκυρος. Επίσης, να υπολογίζει και να τυπώνει πόσοι από τους αριθμούς που θα σχηματιστούν είναι ζυγοί.

Παράδειγμα Εισόδου (πληκτρολόγιο)

0 1 2 3 4

Παράδειγμα Εξόδου (οθόνη)

Δώσε 5 συνεχόμενα ψηφία:
Πλήθος αριθμών= 96
Πλήθος ζυγών αριθμών= 60

**Άσκηση 4.62**

Η ημέρα Τρίτη και 13 θεωρείται γρουσουζική μέρα. Να γράψετε πρόγραμμα το οποίο να δέχεται το έτος, καθώς και την πρώτη ημέρα του έτους (να κάνετε την αντιστοίχιση των ημερών με αριθμούς: 1-Δευτέρα, 2-Τρίτη, 3-Τετάρτη κ.λπ.) και να τυπώνει τις ημερομηνίες και το πλήθος των εμφανίσεων της ημερομηνίας Τρίτης και 13 μέσα στο έτος.

Σημείωση: Προσοχή στα δίσεκτα έτη!

Για να προσδιορίσουμε αν ένα έτος είναι δίσεκτο, εφαρμόζουμε τα εξής:

1. Ελέγχουμε το υπόλοιπο της ακεραίας διαίρεσης του έτους με το 4. Αν είναι μηδέν, ελέγχουμε το υπόλοιπο της ακεραίας διαίρεσης του έτους με το 100. Αν αυτό το υπόλοιπο είναι διαφορετικό του μηδενός, τότε το έτος είναι δίσεκτο.
2. Αν από τον έλεγχο 1 δεν προκύψει ότι το έτος είναι δίσεκτο, ελέγχουμε το υπόλοιπο της ακεραίας διαίρεσης του έτους με το 400. Αν είναι μηδέν, τότε το έτος είναι δίσεκτο, άσχετα από το αποτέλεσμα του ελέγχου 1.

Παράδειγμα Εισόδου (πληκτρολόγιο)

2015
4

Παράδειγμα Εξόδου (οθόνη)

Δώσε έτος:
Δώσε πρώτη μέρα του έτους:
Ημερομηνίες:
13/1/2015
13/10/2015
Εμφανίσεις της Τρίτης και 13ης= 2



Άσκηση 4.63

«Επιμονή» ενός αριθμού είναι ένας όρος, ο οποίος χρησιμοποιείται για να περιγράψει τον αριθμό των βημάτων που κάποιος πρέπει να εφαρμόσει σε έναν ακέραιο αριθμό, πριν φτάσει σε έναν συγκεκριμένο στόχο. Για παράδειγμα, η πολλαπλασιαστική επιμονή του αριθμού 39 είναι 3, επειδή χρειάζονται τρία βήματα για να μειωθεί το 39 σε ένα ψηφίο: $39 \rightarrow 3 \cdot 9 = 27 \rightarrow 2 \cdot 7 = 14 \rightarrow 1 \cdot 4 = 4$. Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό N και να υπολογίζει και να τυπώνει την πολλαπλασιαστική επιμονή του.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
679	Δώσε ακέραιο αριθμό: Πολλαπλασιαστική Επιμονή= 5

Άσκηση 4.64

Ο Μάιος του 2015 είχε πέντε Παρασκευές, πέντε Σάββατα και πέντε Κυριακές, κάτι ιδιαίτερα ευχάριστο για όλους. Να γράψετε πρόγραμμα το οποίο να δέχεται ένα έτος και την πρώτη ημέρα του έτους (1-Δευτέρα, 2-Τρίτη, 3-Τετάρτη κ.λπ.) και να τυπώνει όλους τους μήνες, κατά τους οποίους θα έχουμε εμφάνιση αυτού του γεγονότος μέσα στη δοθείσα χρονιά.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
2020 3	Δώσε έτος: Δώσε την πρώτη μέρα του έτους: Μήνες: 5



Άσκηση 4.65

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό N ($1 \leq N \leq 1,000,000$ και να υπολογίζει και να τυπώνει το πλήθος των ψηφίων του αριθμού τα οποία διαιρούν ακριβώς (υπόλοιπο 0) τον αριθμό.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
124	Δώσε ακέραιο αριθμό: Πλήθος ψηφίων= 3

Επεξήγηση: Το ψηφίο των εκατοντάδων διαιρεί τον αριθμό 124 ακριβώς ($124 \% 1 = 0$). Το ίδιο συμβαίνει και με το ψηφίο των δεκάδων ($124 \% 2 = 0$), όπως και με το ψηφίο των μονάδων ($124 \% 4 = 0$).



Άσκηση 4.66

Έστω ότι έχουμε τον παρακάτω αλγόριθμο:

Διάβασε n

Τύπωσε n

```

Όσο το  $n \neq 1$  επανάλαβε {
    Αν  $n$  περιττός τότε{
         $n=3n+1$ 
    }
    Αλλιώς{
         $n=n/2$ 
    }
    Τύπωσε  $n$ 
}

```

Για παράδειγμα, αν $n=22$ τότε θα τυπώσει τους αριθμούς 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 (16 επαναλήψεις). Να γράψετε πρόγραμμα το οποίο να δέχεται ένα ζευγάρι ακεραίων αριθμών X, Y ($0 < X, Y \leq 100,000$) και εφαρμόζοντας τον πιο πάνω αλγόριθμο σε όλους τους αριθμούς που βρίσκονται μέσα στο διάστημα X, Y (συμπεριλαμβανομένων), να τυπώνει το μέγιστο πλήθος επαναλήψεων, το οποίο μπορεί να προκύψει.

Παράδειγμα Εισόδου
(πληκτρολόγιο)

1 10

Παράδειγμα Εξόδου (οθόνη)

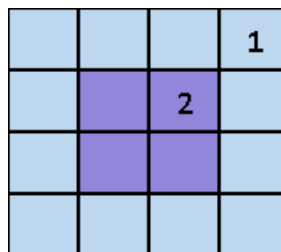
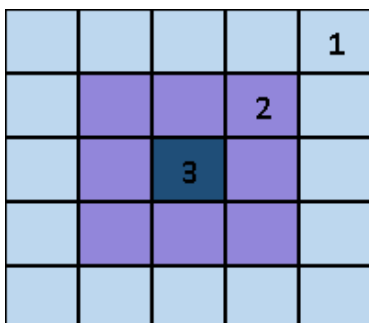
Δώσε 2 ακέραιους αριθμούς:
Πλήθος επαναλήψεων= 20

Επεξήγηση: Το μέγιστο πλήθος επαναλήψεων προκύπτει με τον αριθμό 9.



Άσκηση 4.67

Πιο κάτω, φαίνονται οι κατόψεις δύο πυραμίδων, που θα κτιστούν σε οικοπέδα διαστάσεων 5×5 (αριστερά) και 4×4 (δεξιά). Οι πυραμίδες, που αποτελούνται από τετράγωνα ογκόλιθους, είναι συμπαγείς και δεν υπάρχουν κενά μεταξύ των επιπέδων. Για την αριστερή πυραμίδα, κάθε πλευρά του πρώτου επιπέδου (αριθμός 1), θα χρειαστεί πέντε ογκόλιθους. Το δεύτερο επίπεδο της πυραμίδας (διαστάσεων 3×3), εμφανίζεται με τον αριθμό 2 και το τρίτο επίπεδο (διαστάσεων 1×1), με τον αριθμό 3. Η πυραμίδα στα δεξιά έχει μόνο δύο επίπεδα. Το πρώτο επίπεδο (αριθμός 1), είναι διαστάσεων 4×4 και το δεύτερο επίπεδο (αριθμός 2), είναι διαστάσεων 2×2 .



Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό N ($1 \leq N \leq 1000$), που υποδεικνύει τις διαστάσεις $N \times N$ του οικοπέδου και να υπολογίζει και να τυπώνει το μέγιστο ύψος που μπορεί να φτάσει η πυραμίδα και τους συνολικούς ογκόλιθους που θα χρειαστούν για να κτιστεί.

Collatz, εμφανίζοντας το βήμα που εκτελεί και το αποτέλεσμα του κάθε βήματος.

- (3) Να σταματά όταν φτάνει στο βήμα 1. Αν δεν φτάσει μετά από χίλιες επαναλήψεις στο 1, τότε να εμφανίζεται μήνυμα στον χρήστη ρωτώντας τον κατά πόσο θέλει να συνεχίσει. Το μήνυμα αυτό να εμφανίζεται κάθε χίλιες επαναλήψεις.

Άσκηση 4.69

Ένας αγρότης θέλει να φυτέψει το χωράφι του, το οποίο έχει σχήμα τετραγώνου, με σειρές από ελιές. Για να προστατέψει τις ελιές από τον υπερβολικό αέρα, αποφάσισε να φυτέψει περιμετρικά το χωράφι με κυπαρίσσια. Στο παρακάτω διάγραμμα βλέπουμε τη διάταξη των δέντρων, όπως έχει σκεφτεί να τα φυτέψει ο αγρότης. Κάθε διάγραμμα περιλαμβάνει διαφορετικές σειρές από ελιές.

(n= σειρές από ελιές)

n=1	n=2	n=3
X X X	X X X X X	X X X X X X X
X O X	X O O X	X O O O X
X X X	X X	X X
	X O O X	X O O O X
	X X X X X	X X X X X X X

X = Κυπαρίσσι

O = Ελιά

Να γράψετε πρόγραμμα το οποίο να υπολογίζει και να τυπώνει για ποιον αριθμό n θα έχουμε τον ίδιο αριθμό κυπαρισσιών και ελιών.



Άσκηση 4.70

Η Αριάδνη έχει ένα μεγάλο μπαούλο με A αρκουδάκια. Καθώς, όμως, τα αρκουδάκια πληθαίνουν, ήρθε η ώρα να μεταφέρει κάποια από αυτά στη σοφίτα. Επειδή δυσκολεύεται πολύ να διαλέξει ποια από αυτά θα πάνε στη σοφίτα, έχει βάλει τα αρκουδάκια σε σειρά και σε κάθε ένα από αυτά έβαλε μία ετικέτα με έναν ακέραιο αριθμό. Όσο μεγαλύτερος είναι ο αριθμός τόσο περισσότερο αρέσει στην Αριάδνη να παίζει με το συγκεκριμένο αρκουδάκι.

Αποφάσισε να επιλέξει τα N αρκουδάκια, που θα αποθηκεύσει στη σοφίτα με τον εξής τρόπο:

- (1) Τα αρκουδάκια δεν πρέπει να έχουν ετικέτα μεγαλύτερη από τον αριθμό X.
- (2) Πρέπει να βρίσκονται σε συνεχόμενες θέσεις.

Να γράψετε πρόγραμμα το οποίο θα υπολογίζει με πόσους πιθανούς τρόπους μπορεί να επιλέξει τα αρκουδάκια που θα αποθηκεύσει στη σοφίτα.

Δεδομένα εισόδου

1η γραμμή: Οι αριθμοί A ($1 < A \leq 10000$), X ($0 \leq X \leq 100000$), N ($1 \leq N \leq A$).

2η γραμμή: A αριθμοί, που αντιστοιχούν στην ετικέτα που έχουν τα αρκουδάκια.

Δεδομένα εξόδου

Ένας ακέραιος αριθμός, που αντιστοιχεί στους πιθανούς τρόπους που μπορεί να επιλέξει η Αριάδνη, για να αποθηκεύσει τα αρκουδάκια στη σοφίτα.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 4 3 3 2 3 1 1	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε 3 ακέραιους αριθμούς: Δώσε αριθμούς: Πλήθος πιθανών τρόπων= 2
Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 4 3 3 2 3 1 1	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε 3 ακέραιους αριθμούς: Δώσε 4 αριθμούς: Πλήθος πιθανών τρόπων= 2
Παράδειγμα Εισόδου 3 (πληκτρολόγιο) 11 4 2 2 2 0 7 3 2 2 4 9 1 4	Παράδειγμα Εξόδου 3 (οθόνη) Δώσε 3 ακέραιους αριθμούς: Δώσε αριθμούς: Πλήθος πιθανών τρόπων= 6

**Άσκηση 4.71**

Ο Παντελής ψωνίζει στο εμπορικό κέντρο σχεδόν κάθε βδομάδα. Θέλει να επισκεφτεί συγκεκριμένα μαγαζιά, για να ψωνίσει. Ο Παντελής θέλει να περπατήσει τη μικρότερη δυνατή απόσταση από το σημείο που θα σταθμεύσει το αυτοκίνητό του μέχρι να ψωνίσει και να επιστρέψει σε αυτό. Ο χώρος στάθμευσης του εμπορικού κέντρου είναι πολύ μεγάλος και ο Παντελής πρέπει να διαλέξει το κατάλληλο σημείο, για να σταθμεύσει. Μπορείτε να τον βοηθήσετε;

Δεδομένα εισόδου

- Ένας ακέραιος αριθμός N ($1 \leq N \leq 50$), το πλήθος των μαγαζιών που θα επισκεφτεί ο Παντελής.
- N ακέραιοι x ($0 \leq x \leq 99$), η θέση κάθε μαγαζιού.

Δεδομένα εξόδου

Η μικρότερη απόσταση που μπορεί να περπατήσει ο Παντελής, αν σταθμεύσει στην κατάλληλη θέση.

Παράδειγμα Εισόδου (πληκτρολόγιο) 2 20 50	Παράδειγμα Εξόδου (οθόνη) Δώσε ακέραιο αριθμό: Δώσε αριθμούς: Μικρότερη απόσταση= 60
--	--

Επεξήγηση: Ο Παντελής θέλει να επισκεφτεί δύο μαγαζιά. Το ένα βρίσκεται στη θέση 20 και το άλλο στη θέση 50. Ο Παντελής σταθμεύει στη θέση 20, όπου βρίσκεται το πρώτο μαγαζί, περπατά 30 μέτρα για να πάει στη θέση 50 όπου βρίσκεται το δεύτερο μαγαζί και ακολούθως, περπατά άλλα 30 μέτρα για να επιστρέψει στο αυτοκίνητο. Άρα ο Παντελής έχει περπατήσει συνολικά 60 μέτρα.



Άσκηση 4.72

Ένα από τα μεγαλύτερα άλυτα προβλήματα της αρχαιότητας αποτελεί η εικασία του Goldbach, η οποία δηλώνει ότι κάθε άρτιος αριθμός μεγαλύτερος του 2 μπορεί να γραφτεί ως άθροισμα δύο πρώτων αριθμών. Ως πρώτος αριθμός ορίζεται κάθε φυσικός αριθμός μεγαλύτερος του 1, ο οποίος έχει ως διαιρέτες μόνο το 1 και τον εαυτό του.

Έτσι, οι δέκα πρώτοι αριθμοί είναι:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29.

Καλείστε να γράψετε πρόγραμμα προκειμένου να επαληθεύσετε την εικασία του Goldbach.

Δεδομένα εισόδου

Η είσοδος αποτελείται από έναν άρτιο ακέραιο αριθμό N ($2 < N \leq 1,000,000$).

Δεδομένα εξόδου

Η έξοδος θα αποτελείται από μία μόνο γραμμή, με δύο πρώτους αριθμούς $P1$ και $P2$ χωρισμένους από ένα κενό. Θα πρέπει να ισχύει $P1 + P2 = N$. Σε περίπτωση που υπάρχουν πολλές λύσεις, τυπώστε αυτή με το μικρότερο $P1$.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
14	Δώσε έναν άρτιο αριθμό: Πρώτοι αριθμοί: 3 11



Άσκηση 4.73

Να γράψετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό N ($3 \leq N \leq 80$) και να τυπώνει ένα πλέγμα διαστάσεων $N \times N$ στην οθόνη, χρησιμοποιώντας τους χαρακτήρες του αστερίσκου (*) και του σταυρού (+). Αν το N είναι περιττός αριθμός, τότε ο ένας και μοναδικός χαρακτήρας που βρίσκεται στο κέντρο του τετραγώνου θα πρέπει να είναι σταυρός και οι υπόλοιποι αστερίσκοι. Αν το N είναι άρτιος, τότε οι τέσσερις χαρακτήρες στο κέντρο του τετραγώνου θα πρέπει να είναι σταυροί και οι υπόλοιποι αστερίσκοι.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο)	Παράδειγμα Εξόδου 1 (οθόνη)
3	Δώσε ακέραιο αριθμό: *** *+* ***

Παράδειγμα Εισόδου 2 (πληκτρολόγιο) 4	Παράδειγμα Εξόδου 2 (οθόνη) Δώσε ακέραιο αριθμό: **** *++* *++* ****
Παράδειγμα Εισόδου 3 (πληκτρολόγιο) 7	Παράδειγμα Εξόδου 3 (οθόνη) Δώσε ακέραιο αριθμό: ***** ***** ***** ***+*** ***** ***** *****



Άσκηση 4.74

Δίνονται πιο κάτω μαθηματικές εξισώσεις:

- $x + y + z = A$
- $xyz = B$
- $x^2 + y^2 + z^2 = C$

Αν σας δοθούν τρεις ακέραιοι A, B και C ($1 \leq A, B, C \leq 10000$), να βρείτε τρεις μοναδικούς ακέραιους x, y, z, που να επιλύουν όλες τις πιο πάνω εξισώσεις.

Παράδειγμα Εισόδου (πληκτρολόγιο) 6 6 14	Παράδειγμα Εξόδου (οθόνη) Δώσε 3 ακέραιους αριθμό: Οι αριθμοί που επιλύουν τις εξισώσεις: 1 2 3
--	--

Επεξήγηση:

Για x=1, y=2 και z=3 επιλύουμε τις τρεις εξισώσεις ως εξής:

$$1+2+3 = 6$$

$$1*2*3 = 6$$

$$1+4+9 = 14$$

B7.1.5 Πίνακες

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ❖ Να αναγνωρίζουμε προβλήματα που δεν μπορούν να επιλυθούν με απλές μεταβλητές, αλλά με μονοδιάστατο πίνακα
- ❖ Να εντοπίζουμε ποιοι πίνακες (arrays) χρειάζονται με βάση τις ανάγκες του αλγορίθμου/προγράμματος
- ❖ Να δίνουμε κατάλληλο όνομα (αναγνωριστικό) σε έναν πίνακα και να καθορίζουμε το μέγεθός του (τον αριθμό στοιχείων που μπορεί να περιέχει)
- ❖ Να επιλέγουμε κατάλληλο τύπο δεδομένων (ακέραιος/πραγματικός αριθμός, Boolean, χαρακτήρας/κείμενο) για έναν πίνακα
- ❖ Να δηλώνουμε πίνακες (arrays) με βάση τις ανάγκες του προγράμματος
- ❖ Να αναφερόμαστε σε στοιχεία του πίνακα, χρησιμοποιώντας κατάλληλο δείκτη
- ❖ Να χρησιμοποιούμε τον βρόχο for... για είσοδο/εισαγωγή (εκχώρηση τιμής) στα στοιχεία ενός πίνακα
- ❖ Να χρησιμοποιούμε τον βρόχο for... για έξοδο (εμφάνιση) των τιμών των στοιχείων ενός πίνακα
- ❖ Να χρησιμοποιούμε άλλους βρόχους (π.χ. while, do .. while) για είσοδο, έξοδο και επεξεργασία στοιχείων ενός πίνακα, σύμφωνα με την περιγραφή συγκεκριμένου προβλήματος
- ❖ Να ορίζουμε και να χρησιμοποιούμε συμβολοσειρές, σύμφωνα με τις ανάγκες του προγράμματος.
- ❖ Να ορίζουμε και να χρησιμοποιούμε παράλληλους πίνακες σε ένα πρόγραμμα
- ❖ Να καθορίζουμε το πρόβλημα με ακρίβεια, συγκεκριμένα: να εντοπίζουμε/ διακρίνουμε τα Δεδομένα, τις Πληροφορίες και την Επεξεργασία
- ❖ Να αποφασίζουμε εάν χρειάζονται πίνακες για την επίλυση του προβλήματος και να τους καθορίζουμε
- ❖ Να σχεδιάζουμε τον τρόπο επίλυσης του προβλήματος
- ❖ Να επιλέγουμε κατάλληλες δομές (επανάληψης ή και διακλάδωσης) ανάλογα με τις δυνατότητες, τους περιορισμούς και τα χαρακτηριστικά της δομής για επίλυση του προβλήματος
- ❖ Να υλοποιούμε τον σχεδιασμό τους σε πρόγραμμα, με τη χρήση του προγραμματιστικού περιβάλλοντος, ώστε να επιλυθεί το πρόβλημα
- ❖ Να επιλέγουμε κατάλληλα δεδομένα και στρατηγική για έλεγχο του προγράμματος
- ❖ Να ελέγχουμε την ορθότητα της λύσης του προγράμματος με τη χρήση του debugger για επαλήθευση
- ❖ Να μελετούμε έτοιμο πρόγραμμα, το οποίο περιλαμβάνει μονοδιάστατους πίνακες και να εντοπίζουμε βασικά μέρη του, τα οποία συνδέονται με πτυχές του προβλήματος που επιλύει
- ❖ Να συμπληρώνουμε ένα έτοιμο πρόγραμμα με πίνακες και δομή/δομές επανάληψης, ώστε να αποτελεί λύση ενός διαφοροποιημένου προβλήματος.

1. Εισαγωγή

Στη C++, όπως και σε όλες τις γλώσσες προγραμματισμού, υπάρχει η δυνατότητα ομαδοποίησης δεδομένων ίδιου τύπου. Ο τρόπος με τον οποίο επιτυγχάνεται αυτό είναι με τη δημιουργία πινάκων (arrays). Αναφορά στον πίνακα γίνεται μέσω ενός ονόματος (αναγνωριστικού), όπως στις μεταβλητές, ενώ για πρόσβαση στα δεδομένα ενός πίνακα χρησιμοποιείται το όνομα και ένας αριθμός (δείκτης - index), ο οποίος καθορίζει τη θέση του στοιχείου μέσα στον πίνακα.

Αν χρειαστεί, για παράδειγμα, να γράψουμε ένα πρόγραμμα, το οποίο να δέχεται τις βαθμολογίες πέντε κριτών σε έναν αγώνα καταδύσεων, χωρίς τη χρήση πίνακα, θα πρέπει να δηλώσουμε πέντε μεταβλητές:

```
double scores1, scores2, scores3, scores4, scores5;
```

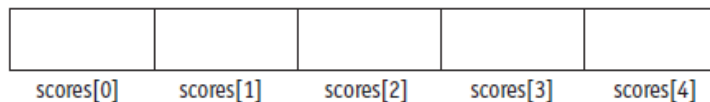
Με τη χρήση ενός πίνακα, μπορούμε να έχουμε και τις πέντε τιμές μέσα σε μία μόνο δομή:

```
double scores[5];
```


Με την πιο πάνω δήλωση, δημιουργούμε πέντε στοιχεία τύπου double (πραγματικός αριθμός), τα οποία αποθηκεύονται σε σειρά στη μνήμη. Τα πέντε στοιχεία του πιο πάνω πίνακα αναφέρονται ως:

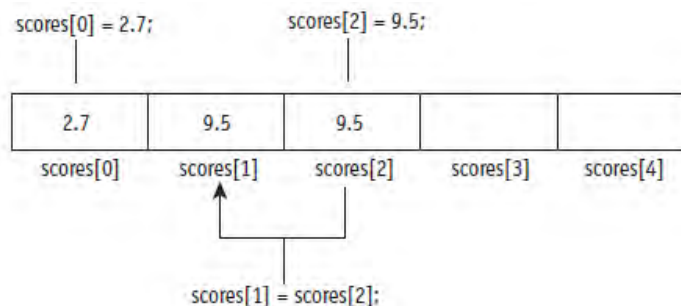
`scores[0]`, `scores[1]`, `scores[2]`, `scores[3]`, `scores[4]`

Οι αριθμοί μέσα στις αγκύλες καλούνται δείκτες (indexes). Στην C++ η αρίθμηση των στοιχείων ενός πίνακα ξεκινά πάντα από το μηδέν και μπορείτε να τον φανταστείτε όπως το πιο κάτω σχήμα:



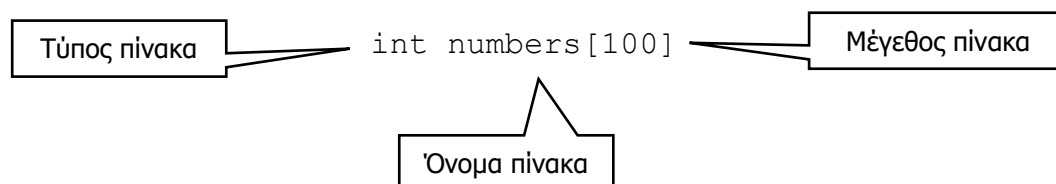
Μέσα στο πρόγραμμα μπορούμε να χρησιμοποιήσουμε τα στοιχεία του πίνακα σαν κανονικές μεταβλητές.

```
scores[0] = 2.7;           // Ο κριτής 1 δίνει βαθμολογία 2.7
scores[2] = 9.5;           // Ο κριτής 3 δίνει βαθμολογία 9.5
scores[1] = scores[2];     // Ο κριτής 2 δίνει την ίδια βαθμολογία με τον 3
```



2. Δήλωση πινάκων

Ένας πίνακας θα πρέπει να δηλωθεί πριν χρησιμοποιηθεί μέσα σε ένα πρόγραμμα. Για τον ορισμό ενός πίνακα είναι απαραίτητα τρία στοιχεία. Το όνομα του πίνακα, τον τύπο δεδομένων των στοιχείων του και τον αριθμό των θέσεων του πίνακα.



Αρχικά γράφουμε τον τύπο των δεδομένων των στοιχείων που περιέχονται στον πίνακα, μετά το όνομα του πίνακα και στη συνέχεια το πλήθος των στοιχείων του πίνακα ανάμεσα σε αγκύλες. Το πλήθος των στοιχείων καλείται και μέγεθος του πίνακα, επειδή καθορίζει πόσες θέσεις δεδομένων του τύπου που ορίσαμε θα δεσμευτούν στη μνήμη.

3. Απόδοση αρχικών τιμών σε πίνακες

Αν γνωρίζουμε εκ των προτέρων τις τιμές που θα πάρουν τα στοιχεία του πίνακα, η απόδοση τιμών μπορεί να γίνει ως εξής:

```
int grades[4] = {16, 18, 15, 20};
```

Η δήλωση του μεγέθους μέσα στις αγκύλες δεν είναι απαραίτητη όταν αποδίδουμε αρχικές τιμές, αφού ο μεταγλωττιστής μπορεί να καταμετρήσει τις τιμές μέσα στις παρενθέσεις. Αν δώσουμε το μέγεθος του πίνακα και το πλήθος των αρχικών τιμών μέσα στα άγκιστρα είναι μικρότερο, δεν θα υπάρχει πρόβλημα, καθώς στα στοιχεία του πίνακα που απομένουν θα αποδοθεί ως αρχική τιμή το 0. Αν όμως το πλήθος των αρχικών τιμών μέσα στα άγκιστρα είναι μεγαλύτερο από το μέγεθος που δηλώθηκε, τότε υπάρχει σφάλμα. Ας δούμε κάποια παραδείγματα:

```
int primes[] = {1,2,3,5,7};    //Δεν χρειάζεται να δηλώνουμε το μέγεθος.
char characters[3] = {'a', 'b', 'c', 'd'}; //Σφάλμα! Περισσότερα στοιχεία.
int a[5] = {12,5,4};           // a → {12, 5, 4, 0, 0}
```

4. Εισαγωγή στοιχείων σε πίνακα

Για την εισαγωγή στοιχείων σε πίνακα μπορούμε να χρησιμοποιήσουμε τη δομή επανάληψης for μαζί με μία μεταβλητή-δείκτη. Η διαδικασία έχει ως ακολούθως:

```
char arr[10];
for (int i=0; i<10; i++)
    cin >> arr[i];    //το στοιχείο που διαβάζεται θα
                      //αποθηκευτεί στην i-οστή θέση του πίνακα
```

5. Εμφάνιση στοιχείων του πίνακα

Για την εμφάνιση των στοιχείων του πίνακα στην οθόνη χρησιμοποιούμε παρόμοια δομή επανάληψης με την εισαγωγή στοιχείων με εντολή εξόδου.

```
char arr[10];
// Εισαγωγή στοιχείων σε πίνακα
for (int i=0; i<10; i++)
    cin >> arr[i];
// Εμφάνιση στοιχείων του πίνακα
for (int i=0; i<10; i++)
    cout << arr[i] << " ";
```

Παράδειγμα 5.1

Να γράψετε πρόγραμμα το οποίο να αποθηκεύει τους αριθμούς 8,0,69,167,349,-428,29 σε έναν πίνακα ακεραίων και να τους τυπώνει στην οθόνη.

```
#include<iostream>
using namespace std;
int main(){
    int arr[] = {8,0,69,167,349,-428,29};    // μέγεθος 7
    for (int i = 0; i < 7; i++)              // χρήση δομής επανάληψης
        cout << arr[i] << " ";
    return 0;
}
```

```
arr = [8,0,69,167,349,-428,29]
for i in range(0,7):
    print (arr[i], end=" ")
```



Παράδειγμα 5.2

Να γράψετε πρόγραμμα το οποίο να αποθηκεύει τους χαρακτήρες 'Ο', 'L', 'L', 'E', 'H' σε έναν πίνακα χαρακτήρων και να τους τυπώνει στην οθόνη αντίστροφα, δηλαδή ο πρώτος να εμφανίζεται τελευταίος, ο δεύτερος να εμφανίζεται προτελευταίος.

```
#include<iostream>
using namespace std;
int main(){
    char arr[] = {'O', 'L', 'L', 'E', 'H'};    // μέγεθος 5

    for (int i = 4; i >= 0; i--)                // χρήση δομής επανάληψης
        cout << arr[i] << " ";
    return 0;
}
```

```
package par5_2;
public class par5_2 {
    public static void main(String arg[]){
        char[] arr = {'O', 'L', 'L', 'E', 'H'};
        for(int i=4;i>=0;i--)
            System.out.print(arr[i]);
    }
}
```



6. Υπολογισμός του μέσου όρου των στοιχείων του πίνακα

```
int arr[10],sum=0;
float mo;
// Υπολογισμός του αθροίσματος των στοιχείων του πίνακα
for (int i=0; i<10; i++)
    sum+= arr[i];
// Υπολογισμός και εμφάνιση του μέσου όρου
mo=sum/10.0
cout << mo;
```

Παράδειγμα 5.3

Να γράψετε πρόγραμμα το οποίο να δέχεται δέκα ακέραιους αριθμούς, να τους αποθηκεύει σε πίνακα και να τυπώνει το πλήθος των στοιχείων του πίνακα, τα οποία είναι μεγαλύτερα από τον μέσο όρο όλων των στοιχείων του πίνακα.

```
#include<iostream>
using namespace std;
int main() {
    int arr[10], sum = 0, ans = 0, i;
    float avg;
    for (i = 0; i < 10; i++) {
        cin >> arr[i];
        sum += arr[i];           // αντιστοιχεί με sum = sum + arr[i]
    }
    avg = float(sum) / 10;
    for (i = 0; i < 10; i++) {
        if (arr[i] > avg)
            ans++;
    }
    cout << ans;
    return 0;
}
```

Παράδειγμα 5.4

Να γράψετε πρόγραμμα το οποίο να γεμίζει έναν πίνακα ακεραιών 30 θέσεων με τους 30 πρώτους αριθμούς της ακολουθίας: 1, 3, 7, 15, 31, 63, $2_{N-1}+1$. Ακολουθώς, να ζητά από τον χρήστη έναν αριθμό N ($1 \leq N \leq 30$) και να τυπώνει τον N-οστό αριθμό της πιο πάνω σειράς.

```
#include<iostream>
using namespace std;
int main() {
    int arr[30], N;
    arr[0]=1;           // Γνωρίζουμε το πρώτο στοιχείο

    for (int i = 1; i < 30; i++)
        arr[i] = 2 * arr[i-1] + 1;    // Από το προηγούμενο στοιχείο
                                     // βρίσκουμε το επόμενο
    cin >> N;                       // Εισαγωγή θέσης στοιχείου
    cout << arr[N-1];    // ο δείκτης μας πρέπει να μειωθεί κατά ένα
    return 0;
}
```

```
program par5_4;
type
    tb = array[1..30] of integer;
var
    arr:tb;
    N,i:integer;
```



```

begin
    arr[1]:=1;
    for i:=2 to 30 do
        arr[i]:=2*arr[i-1]+1;
    readln(N);
    writeln(arr[N])
end.

```

7. Παράλληλοι πίνακες

Πολλές φορές έχουμε περισσότερα δεδομένα, διαφορετικών τύπων, για να χειριστούμε. Για παράδειγμα, έχουμε τον πιο κάτω πίνακα, ο οποίος παρουσιάζει τα συνολικά κέρδη μία εταιρείας ανά έτος:

YEAR	PROFIT
2010	97.77
2011	98.34
2012	103.56
2013	102.68
2014	104.54

Για να καταφέρουμε να συσχετίσουμε τα πιο πάνω στοιχεία, μπορούμε να χρησιμοποιήσουμε δύο παράλληλους πίνακες (parallel arrays). Αρχικά, καταχωρούμε τα στοιχεία σε δύο πίνακες:

```

int year[] = {2010, 2011, 2012, 2013, 2014};
double profit[] = {97.77, 98.34, 103.56, 102.68, 104.54};

```

Αν η τιμή του δείκτη ενός από τους πίνακες είναι η ίδια με την τιμή του δείκτη του άλλου πίνακα, τότε αναφερόμαστε στα δύο στοιχεία της ίδιας χρονιάς. Δηλαδή, τα στοιχεία year[3] και profit[3] θα αναφέρονται στα στοιχεία 2013 και 102.68, αντίστοιχα. Με αυτόν τον τρόπο μπορούμε να επιτύχουμε τον συσχετισμό ανόμοιων στοιχείων. Αν θέλουμε να εμφανίσουμε το πιο πάνω αποτέλεσμα στην οθόνη, τότε θα γράψουμε τον πιο κάτω κώδικα:

```

cout << "YEAR" << "\t" << "PROFIT" << endl; // Το \t αντιστοιχεί
                                                // με το πλήκτρο TAB
for (int i = 0; i < 5; i++)
    cout << year[i] << "\t" << profit[i] << endl;

```

Παράδειγμα 5.5

Να γράψετε πρόγραμμα το οποίο να δέχεται αρχικά τα στοιχεία μίας παραγγελίας μίας υπεραγοράς. Συγκεκριμένα, το πρόγραμμα να δέχεται για 50 διαφορετικά προϊόντα: τον κωδικό του προϊόντος, την ποσότητα και την τιμή και θα τα αποθηκεύει σε κατάλληλους παράλληλους πίνακες. Για κάθε ένα από τα προϊόντα, το πρόγραμμα να υπολογίζει το κόστος (κόστος = ποσότητα x τιμή) και να το αποθηκεύει σε πίνακα. Τέλος, να υπολογίζει και να τυπώνει τη συνολική χρέωση της παραγγελίας. Ενδεικτικά, βλέπετε ένα μέρος της παραγγελίας στον πιο κάτω πίνακα. Οι πρώτες τρεις στήλες παρουσιάζουν τον κωδικό, την ποσότητα και την τιμή. Η τέταρτη στήλη υπολογίζει το κόστος για κάθε προϊόν. Τέλος, εμφανίζεται η συνολική χρέωση της παραγγελίας.

Κωδικός	Ποσότητα	Τιμή	Ποσό
1001	30	3.99	119.70
1002	45	2.49	112.05
1003	12	0.99	11.88
1004	18	5.99	107.82
.	.	.	.
.	.	.	.
.	.	.	.
1050	10	8.49	84.90
Συνολική χρέωση παραγγελίας:			9987.46

Υλοποίηση στη γλώσσα προγραμματισμού C++

```
#include<iostream>
using namespace std;
#define N 50
int main() {
    int code[N], quantity[N];
    double price[N], total[N], ans = 0;
    for (int i = 0; i < N; i++){
        cin >> code[i] >> quantity[i] >> price[i];
        total[i] = quantity[i] * price[i];
        ans += total[i];
    }
    cout << ans;
    return 0;
}
```



Υλοποίηση στη γλώσσα προγραμματισμού Pascal

```
program par5_5;
const
    N=50;
type
    tb1=Array[1..N] of integer;
    tb2=Array[1..N] of real;
var
    code,quantity:tb1;
    price,total:tb2;
    i:integer;
    ans:real;
Begin
    ans:=0;
    for i:=1 to N do
        Begin
```



```

        readln(code[i],quantity[i]);
        readln(price[i]);
        total[i]:=quantity[i]*price[i];
        ans:=ans+total[i]

    end;

    writeln(ans)

end.

```

Παράδειγμα 5.6

Να γράψετε πρόγραμμα το οποίο να δέχεται 20 ακέραιους αριθμούς. Οι πρώτοι 10 αριθμοί καταχωρίζονται στον πίνακα **Num1** και οι επόμενοι στον πίνακα **Num2**. Στη συνέχεια υπολογίζει το άθροισμα μεταξύ των αντίστοιχων στοιχείων του κάθε πίνακα και να το αποθηκεύει στον πίνακα **Sum**. Τέλος, να τυπώνει τους τρεις πίνακες τον ένα δίπλα στον άλλο, με τις κατάλληλες κεφαλίδες.

```

#include<iostream>
#include<iomanip>
using namespace std;
int main() {
    int Num1[10], Num2[10], Sum[10];
    int i;
    for(i=0;i<10;i++)
        cin>>Num1[i];
    for(i=0;i<10;i++){
        cin>>Num2[i];
        Sum[i]=Num1[i]+Num2[i];
    }
    cout<<setw(6)<<"Num1"<<setw(6)<<"Num2"<<setw(6)<<"Sum"<<endl;
    for(i=0;i<10;i++)
        cout<<setw(6)<<Num1[i]<<setw(6)<<Num2[i]<<setw(6)<<Sum[i]<<endl;
    return 0;
}

```



Ασκήσεις Κεφαλαίου

Άσκηση 5.1

Δίνονται οι πιο κάτω πίνακες:

A	B	X
20	16	0
18	22	0
25	50	0
33	11	0

(α) Να γράψετε τις εντολές για να δηλωθούν οι πίνακες A, B, X αρχικοποιώντας τους με τις τιμές που περιέχονται σε αυτούς.

(β) Να βρείτε τα αποτελέσματα των πιο κάτω εντολών:

- (1) $X[0] = A[0] + B[0];$
- (2) $X[1] = A[3] + B[1];$
- (3) $X[2] = B[2] / A[2];$
- (4) $X[3] = A[1] - B[0] + B[3];$

Άσκηση 5.2

Τι θα εμφανίσει στην οθόνη ο πιο κάτω κώδικας;

```
int numbers[5];
numbers[0] = 8;
numbers[1] = 40;
numbers[2] = numbers[1] - 10;
numbers[3] = numbers[1] / 10;
numbers[4] = numbers[0] / numbers[3];
for (int i=0; i<5; i++)
    cout << numbers[i] << " ";
```

Άσκηση 5.3

Δίνεται το παρακάτω τμήμα κώδικα:

```
for (int i=0; i<10; i++)
    if (A[i]< 100)
        B[i]= A[i] + A[i] / 2;
    else
        B[i]= A[i];
```

Αν ο πίνακας A έχει τα περιεχόμενα:

120	200	500	430	80	40	100	110	600	4
-----	-----	-----	-----	----	----	-----	-----	-----	---

ποια θα είναι τα περιεχόμενα του πίνακα B, μετά την εκτέλεση του παραπάνω τμήματος κώδικα;

--	--	--	--	--	--	--	--	--	--

Άσκηση 5.4

Να γράψετε πρόγραμμα το οποίο να δέχεται δέκα ακέραιους αριθμούς, να τους αποθηκεύει σε πίνακα και να τυπώνει στην οθόνη αυτόν που δόθηκε πρώτος και αυτόν που δόθηκε τελευταίος.

Παράδειγμα Εισόδου (πληκτρολόγιο) 4 4 6 2 5 -8 45 12 9 0	Παράδειγμα Εξόδου (οθόνη) Δώσε 10 ακέραιους αριθμούς: Πρώτος αριθμός=4 Τελευταίος αριθμός=0
--	---

Άσκηση 5.5

Να γράψετε πρόγραμμα το οποίο να εισάγει τιμές σε έναν μονοδιάστατο πίνακα ακεραίων 100 θέσεων, τοποθετώντας την τιμή 0 στις περιττές θέσεις και 1 στις ζυγές θέσεις. Το πρόγραμμα να τυπώνει τον πίνακα.

Παράδειγμα Εξόδου (οθόνη) (για 10 αριθμούς) Οι αριθμοί του πίνακα είναι: 0 1 0 1 0 1 0 1 0 1

Άσκηση 5.6

Να γράψετε πρόγραμμα το οποίο να δέχεται 15 χαρακτήρες και να τους τοποθετεί σε έναν πίνακα. Στη συνέχεια, να τυπώνει τους χαρακτήρες με την αντίστροφη σειρά εισόδου.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 χαρακτήρες) A C D ! e	Παράδειγμα Εξόδου (οθόνη) (για 5 χαρακτήρες) Δώσε χαρακτήρες: Αντίστροφα οι χαρακτήρες είναι: e ! D C A
--	---

Άσκηση 5.7

Να γράψετε πρόγραμμα το οποίο να δέχεται τα 15 ονόματα των μαθητών μιας τάξης και να τα αποθηκεύει σε έναν πίνακα. Στη συνέχεια, να ζητά από τον χρήστη να δώσει έναν αριθμό από το 0 μέχρι το 14 που να παριστάνει τη θέση και να τυπώνει το όνομα του μαθητή που είναι καταχωρημένο στη θέση αυτή.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 μαθητές) Maria Anna Andreas George Marios 3	Παράδειγμα Εξόδου (οθόνη) (για 5 μαθητές) Δώσε τα ονόματα: Δώσε τον αριθμό της θέσης: George
--	--

Άσκηση 5.8

Να γράψετε πρόγραμμα το οποίο να δέχεται τους τελικούς βαθμούς 20 μαθητών και να τους αποθηκεύει σε πίνακα. Στη συνέχεια, να υπολογίζει και να τυπώνει το πλήθος των μαθητών που έμειναν ανεξεταστέοι (βαθμός < 10) και το πλήθος των μαθητών που άριστευσαν (βαθμός 18-20).

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 βαθμούς) 12 19 20 8 14	Παράδειγμα Εξόδου (οθόνη) (για 5 βαθμούς) Δώσε τους βαθμούς των μαθητών: Πλήθος ανεξεταστέων=1 Πλήθος άριστευσάντων= 2
---	--

Άσκηση 5.9

Να γράψετε πρόγραμμα το οποίο να δέχεται 20 χαρακτήρες και να τους αποθηκεύει σε έναν πίνακα. Στη συνέχεια, να δέχεται έναν χαρακτήρα X και να τυπώνει το πλήθος των εμφανίσεων του χαρακτήρα X μέσα στον πίνακα.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 χαρακτήρες) C G U R R R	Παράδειγμα Εξόδου (οθόνη) (για 5 χαρακτήρες) Δώσε τους χαρακτήρες: Δώσε χαρακτήρα προς αναζήτηση: Πλήθος εμφανίσεων= 2
---	--

Άσκηση 5.10

Να γράψετε πρόγραμμα το οποίο να δέχεται 10 ακέραιους αριθμούς και να τους αποθηκεύει στον πίνακα A. Στη συνέχεια, να δέχεται έναν ακέραιο αριθμό B και να υπολογίζει και να τυπώνει το άθροισμα των αριθμών του πίνακα A που είναι μεγαλύτεροι από τον αριθμό B.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 αριθμούς) 23 12 45 6 9 20	Παράδειγμα Εξόδου (οθόνη) (για 5 αριθμούς) Δώσε τους αριθμούς: Δώσε τον αριθμό: Άθροισμα= 68
---	--

**Άσκηση 5.11**

Στο Παγκύπριο Πρωτάθλημα Ρυθμικής Γυμναστικής λαμβάνουν μέρος 15 αθλήτριες. Η βαθμολογία της κάθε αθλήτριας είναι ένας δεκαδικός αριθμός μεταξύ των αριθμών 1 και 10 συμπεριλαμβανομένων. Να γράψετε πρόγραμμα το οποίο να δέχεται τις βαθμολογίες των αθλητριών, να τις αποθηκεύει σε πίνακα και να τυπώνει στην οθόνη τις βαθμολογίες που είναι μεγαλύτερες από τον μέσο όρο όλων των βαθμολογιών.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 αθλήτριες) 9.7 4.3 6.8 9.5 7.2	Παράδειγμα Εξόδου (οθόνη) (για 5 αθλήτριες) Δώσε τις βαθμολογίες: Οι βαθμολογίες είναι: 9.7 9
---	--

Άσκηση 5.12

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται 20 ακέραιους αριθμούς και τους καταχωρίζει στον πίνακα A.
- (2) Να υπολογίζει και τυπώνει τον μέσο όρο των αριθμών.
- (3) Να υπολογίζει και τυπώνει τον μεγαλύτερο και τον μικρότερο αριθμό του πίνακα.
- (4) Να τυπώνει τους αριθμούς του πίνακα σε αντίστροφη σειρά εισόδου.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 αριθμούς) 23 12 15 27 -10	Παράδειγμα Εξόδου (οθόνη) (για 5 αριθμούς) Δώσε τους αριθμούς: Μέσος Όρος=13.40 Μεγαλύτερος αριθμός=27 Μικρότερος αριθμός=-10 Οι αριθμοί σε αντίστροφη σειρά είναι: -10 27 15 12 23
--	--

Άσκηση 5.13

Να γράψετε πρόγραμμα το οποίο να δέχεται 100 πραγματικούς αριθμούς και να τους αποθηκεύει σε έναν πίνακα. Στη συνέχεια, να υπολογίζει και να τυπώνει τον μεγαλύτερο αριθμό, καθώς επίσης και τη θέση της πρώτης εμφάνισής του στον πίνακα.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 αριθμούς) 100.23 12.35 61.80 92.5 172.25	Παράδειγμα Εξόδου (οθόνη) (για 5 αριθμούς) Δώσε τους αριθμούς: Ο μεγαλύτερος αριθμός είναι ο 172.25 Η θέση της πρώτης εμφάνισης είναι η 4
--	---

Άσκηση 5.14

Να γράψετε πρόγραμμα το οποίο να δέχεται τους 20 βαθμούς των μαθητών μιας τάξης στο μάθημα της Πληροφορικής και να υπολογίζει και να τυπώνει τον δεύτερο μεγαλύτερο βαθμό της τάξης.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 βαθμούς) 19 17 16 20 15	Παράδειγμα Εξόδου (οθόνη) (για 5 βαθμούς) Δώσε τους βαθμούς: Δεύτερος μεγαλύτερος βαθμός= 19
--	---

Άσκηση 5.15

Να γράψετε πρόγραμμα το οποίο να δέχεται 12 ακέραιους αριθμούς και να τους τοποθετεί σε έναν πίνακα. Στη συνέχεια, να υπολογίζει και να τυπώνει το πλήθος των εμφανίσεων του μέγιστου αριθμού.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 αριθμούς) 1 5 5 4 2	Παράδειγμα Εξόδου (οθόνη) (για 5 αριθμούς) Δώσε τους αριθμούς: Πλήθος εμφανίσεων= 2
--	--

Άσκηση 5.16

Να γράψετε πρόγραμμα το οποίο να δέχεται τη νικήτρια στήλη του ΠΡΟΠΟ (13 σημεία όπου σημειώνουμε 1 για νίκη γηπεδούχου, 2 για νίκη φιλοξενούμενου και X για ισοπαλία) και να υπολογίζει και να τυπώνει το πλήθος για κάθε ένα από τα σημεία αυτά.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 σημεία) 1 X X 2 1	Παράδειγμα Εξόδου (οθόνη) (για 5 σημεία) Δώσε τις τιμές: Πλήθος νικών γηπεδούχου= 2 Πλήθος νικών φιλοξενούμενου= 1 Πλήθος ισοπαλιών= 2
--	---

Άσκηση 5.17

Η υπηρεσία περιβάλλοντος καταγράφει την πυκνότητα της σκόνης στην ατμόσφαιρα καθημερινά. Η τιμή της πυκνότητας της σκόνης χαρακτηρίζεται από έναν δεκαδικό αριθμό, από το 0.0 μέχρι το 1.0 συμπεριλαμβανομένων. Να γράψετε πρόγραμμα το οποίο:

- (1) Να καταχωρίζει τα στοιχεία για έναν χρόνο στον πίνακα Μ.
- (2) Να υπολογίζει και να τυπώνει τον μέσο όρο.
- (3) Να υπολογίζει και να τυπώνει τις μέρες με τη μεγαλύτερη πυκνότητα σκόνης. Σε περίπτωση που υπάρχουν περισσότερες από μία μέρες με μέγιστο επίπεδο σκόνης, να τις τυπώνει όλες.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 τιμές) 4.3 5.3 4.9 5.3 5.1	Παράδειγμα Εξόδου (οθόνη) (για 5 τιμές) Δώσε τις τιμές της πυκνότητας σκόνης: Μέσος Όρος= 4.98 Οι μέρες με τη μέγιστη πυκνότητα είναι: 2 4
---	---

Άσκηση 5.18

Σε έναν πίνακα 30 θέσεων αποθηκεύεται σε κάθε θέση ένας από τους χαρακτήρες A, B και C. Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται 30 χαρακτήρες και τους αποθηκεύει στον πίνακα χαρ.
- (2) Να υπολογίζει και να τυπώνει το πλήθος των εμφανίσεων του χαρακτήρα B μέσα στον πίνακα.
- (3) Να αντικαθιστά τον χαρακτήρα C με τον χαρακτήρα D και να τυπώνει τον πίνακα (το ένα στοιχείο κάτω από το άλλο).

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 χαρακτήρες)	Παράδειγμα Εξόδου (οθόνη) (για 5 χαρακτήρες)
B C A B X	Δώσε τους χαρακτήρες: Πλήθος εμφανίσεων του B= 2 Τα στοιχεία του πίνακα είναι: B D A B X

Άσκηση 5.19

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται 20 ακέραιους αριθμούς σε αύξουσα σειρά και τους αποθηκεύει σε πίνακα.
- (2) Να υπολογίζει και να τυπώνει στην οθόνη τη μικρότερη διαφορά μεταξύ δύο συνεχόμενων αριθμών του πίνακα.
- (3) Να υπολογίζει και να τυπώνει στην οθόνη τη μεγαλύτερη διαφορά μεταξύ δύο συνεχόμενων αριθμών του πίνακα.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 αριθμούς)	Παράδειγμα Εξόδου (οθόνη) (για 5 αριθμούς)
10 45 122 130 215	Δώσε τους αριθμούς του πίνακα: Μικρότερη διαφορά=8 Μέγιστη διαφορά=85

**Άσκηση 5.20**

Να γράψετε πρόγραμμα το οποίο να δέχεται 1000 χαρακτήρες και να υπολογίζει και να τυπώνει το πλήθος των εμφανίσεων της συμβολοσειράς «GTA», μέσα σε αυτούς τους χαρακτήρες.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 10 χαρακτήρες)	Παράδειγμα Εξόδου (οθόνη) (για 10 χαρακτήρες)
A G T A G T B G T A	Δώσε τους χαρακτήρες: Πλήθος εμφανίσεων=2



Άσκηση 5.21

Να γράψετε πρόγραμμα το οποίο να διαβάσει μέχρι πενήντα τυχαίους ακέραιους αριθμούς. Η εισαγωγή των αριθμών να τερματίζεται, μόλις δοθεί αριθμός που έχει ξαναδοθεί προηγουμένως. Το πρόγραμμα να τυπώνει το άθροισμα των αριθμών, οι οποίοι έχουν καταχωρισθεί χωρίς να λαμβάνει υπόψη τον τελευταίο αριθμό (π.χ. αν καταχωρίζονται οι αριθμοί 2, 3, 6, 5, 4, 7, 6 θα εμφανίζεται άθροισμα 27). (Παγκύπριες εξετάσεις 2011)

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 7 αριθμούς) 2 3 6 5 4 7 6	Παράδειγμα Εξόδου (οθόνη) (για 7 αριθμούς) Δώσε τους αριθμούς: Άθροισμα=27
--	---

Άσκηση 5.22

Σε ένα διαγωνισμό λαμβάνουν μέρος 10 μαθητές. Να γράψετε το πρόγραμμα το οποίο να δέχεται το όνομα και το αποτέλεσμα για κάθε έναν από τους διαγωνιζόμενους και να τυπώνει το όνομα και το αποτέλεσμα των μαθητών που έχουν εξασφαλίσει τη βάση (≥ 50).

Σημείωση: Να χρησιμοποιήσετε δύο παράλληλους πίνακες.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 μαθητές) Johnson 21 Ptak 66 Lane 50 Thomson 89 Soto 48	Παράδειγμα Εξόδου (οθόνη) (για 5 μαθητές) Δώσε τα ονόματα και τα αποτελέσματα: Εξασφάλισαν τη βάση οι μαθητές: Ptak 66 Lane 50 Thomson 89
--	--

Άσκηση 5.23

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται τα ονόματα και τις βαθμολογίες των 20 μαθητών μιας τάξης και τα καταχωρίζει στους πίνακες Names και Grades, αντίστοιχα.
- (2) Να υπολογίζει και να τυπώνει τον μέσο όρο των βαθμολογιών.
- (3) Να υπολογίζει και να τυπώνει το πλήθος των μαθητών που έχουν βαθμολογία μεγαλύτερη από τον μέσο όρο.
- (4) Να υπολογίζει και να τυπώνει το όνομα του μαθητή με τη μεγαλύτερη βαθμολογία.
- (5) Να τυπώνει τα ονόματα και τις βαθμολογίες χρησιμοποιώντας τα κατάλληλα μηνύματα.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 μαθητές)	Παράδειγμα Εξόδου (οθόνη) (για 5 μαθητές)
Andreas 14 Georgia 17 Maria 20 Nicholas 19 Eleni 16	Δώσε το όνομα και τη βαθμολογία: Μέσος Όρος= 17.20 Πλήθος μαθητών= 2 Μαθητής με μεγαλύτερη βαθμολογία= Maria Όνομα Βαθμολογία Andreas 14 Georgia 17 Maria 20 Nicholas 19 Eleni 16

Άσκηση 5.24

Ο πιο κάτω πίνακας παρουσιάζει το όνομα, τον αριθμό των νικών, τον αριθμό των ηττών και τον αριθμό των ισοπαλιών για κάθε μια από τις δέκα ποδοσφαιρικές ομάδες.

Team	Wins	Ties	Losses
Packers	14	1	0
Patriots	13	2	0
49ers	12	2	1
Saints	11	3	1
Giants	10	2	3
Broncos	9	4	2
Ravens	8	4	3
Vikings	4	7	4
Eagles	2	8	5
Colts	1	1	13

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται το όνομα, τον αριθμό των νικών, τον αριθμό των ηττών και τον αριθμό των ισοπαλιών κάθε ομάδας και να αποθηκεύει τα στοιχεία αυτά σε παράλληλους πίνακες.
- (2) Να υπολογίζει το σύνολο των βαθμών κάθε ομάδας και να το αποθηκεύει σε έναν ξεχωριστό πίνακα.

Σημείωση: Κάθε ομάδα παίρνει 3 βαθμούς για κάθε νίκη, 1 για κάθε ισοπαλία και 0 για κάθε ήττα.

- (3) Να τυπώνει το όνομα της κάθε ομάδας και το σύνολο των βαθμών της.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 ομάδες)	Παράδειγμα Εξόδου (οθόνη) (για 5 ομάδες)
Packers 14 1 0 Patriots 13 2 0 49ers 12 2 1 Saints 11 3 1 Giants 10 2 3	Δώσε τα στοιχεία της ομάδας: Ομάδα Βαθμοί Packers 43 Patriots 41 49ers 38 Saints 36 Giants 32

Άσκηση 5.25

Να γράψετε πρόγραμμα το οποίο να αποθηκεύει σε δύο μονοδιάστατους πίνακες τα ονόματα και τις ηλικίες μέχρι και 100 ατόμων. Ανήλικος θεωρείται κάποιος ο οποίος είναι κάτω των δεκαοκτώ (18) ετών.

Το πρόγραμμα πρέπει:

- (1) Δέχεται το πλήθος των ατόμων $N(1 \leq N \leq 100)$ και να αποθηκεύει στους πίνακες τα ονόματα και τις ηλικίες των ατόμων.
- (2) Να τυπώνει το περιεχόμενο των δύο πινάκων σε δύο στήλες.
- (3) Να υπολογίζει και να τυπώνει το ποσοστό επί τοις εκατό των ανηλίκων.
- (4) Να τυπώνει το όνομα και την ηλικία του μεγαλύτερου σε ηλικία ατόμου.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 άτομα)	Παράδειγμα Εξόδου (οθόνη) (για 5 άτομα)
Andreas 14 George 23 Maria 22 Nicholas 21 Eleni 7	Δώσε το όνομα και την ηλικία: Όνομα Ηλικία Andreas 14 George 23 Maria 22 Nicholas 21 Eleni 7 Ποσοστό=40.00% Όνομα μεγαλύτερου ατόμου= George Ηλικία=23

Άσκηση 5.26

Σε μια έρευνα του τμήματος Μετεωρολογίας Κύπρου μετρήθηκε το επίπεδο βροχόπτωσης για το μήνα Δεκέμβριο για 5 έτη (2014 μέχρι 2018) . Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται τις 5 βροχοπτώσεις και να τις αποθηκεύει στον μονοδιάστατο πίνακα vroxopt.

- (2) Να υπολογίζει και να τυπώνει τον μέσο όρο των επιπέδων βροχόπτωσης και για τα πέντε (5) έτη όπως και το πλήθος των μετρήσεων που είναι μικρότερες από τον μέσο όρο.
- (3) Να υπολογίζει και να τυπώνει το μέγιστο επίπεδο βροχόπτωσης αλλά και το τελευταίο έτος στο οποίο έχει καταγραφεί το επίπεδο αυτό.
- (4) Να τυπώνει σε πόσες περιπτώσεις το επίπεδο βροχόπτωσης ήταν ψηλότερο από το προηγούμενο και το επόμενο έτος.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
261.1 423.3 628.0 295.8 235.7	Δώσε βροχοπτώσεις: Μέσος Όρος= 368.78 Πλήθος μετρήσεων= 3 Μέγιστη βροχόπτωση= 628.00 Έτη μέγιστης βροχόπτωσης= 3ο έτος Περιπτώσεις= 1

Άσκηση 5.27

Ο πιο κάτω πίνακας παρουσιάζει το όνομα, την ποσότητα και την τιμή για μια νέα παραλαβή 10 φαρμάκων.

Όνομα	Ποσότητα	Τιμή
Antibiotic	75	€6.00
Aspirin	50	€3.50
Vitamin_C	60	€3.00
Vitamin_A	20	€4.00
Folic_Acid	15	€2.50
Multivitamin	50	€7.00
Omega_3	100	€6.00
Aminoacid	30	€5.00
Vitamin_E	10	€4.50
Magnesium	30	€5.00

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται το όνομα, την ποσότητα και την τιμή για κάθε ένα από τα 10 φάρμακα και να αποθηκεύει τα στοιχεία αυτά σε παράλληλους πίνακες.
- (2) Να τυπώνει τα ονόματα των φαρμάκων, που στοιχίζουν πάνω από €5.00 ανά τεμάχιο.
- (3) Να υπολογίζει και να τυπώνει τον συνολικό αριθμό των τεμαχίων που έχει παραλάβει.
- (4) Να υπολογίζει την συνολική τιμή για κάθε ένα από τα δέκα φάρμακα και να αποθηκεύει την τιμή αυτή σε ένα μονοδιάστατο πίνακα ο οποίος να είναι παράλληλος με τους άλλους πίνακες.

Παράδειγμα: Για τη Βιταμίνη E ο φαρμακοποιός πρέπει να πληρώσει δηλαδή $10 \times 4.50 = €45$.

- (5) Να τυπώνει το όνομα και τη συνολική του τιμή για κάθε φάρμακο.
- (6) Να υπολογίζει και να τυπώνει το συνολικό ποσό που πρέπει να πληρώσει ο φαρμακοποιός, για όλα τα φάρμακα, αν υπάρχει και επιπλέον χρέωση 20% για το ΦΠΑ πάνω στη συνολική τιμή.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 φάρμακα)	Παράδειγμα Εξόδου (οθόνη) (για 5 φάρμακα)
Antibiotic 75 6.00 Aspirin 50 3.50 Vitamin_C 60 3.00 Vitamin_A 20 4.00 Folic_Acid 15 2.50	Δώσε τα στοιχεία των φαρμάκων: Φάρμακα που κοστίζουν πάνω από 5 ευρώ Antibiotic Multivitamin Omega_3 Αριθμός τεμαχίων=410 Όνομα Συνολική Τιμή Antibiotic 487.50 Aspirin 175.00 Vitamin_C 180.00 Vitamin_A 80.00 Folic_Acid 37.50 Συνολικό Ποσό=1151.40 ευρώ

Άσκηση 5.28

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται 30 ακέραιους αριθμούς (>99) και τους αποθηκεύει μέσα στον πίνακα N.
- (2) Για κάθε αριθμό να ελέγχει αν είναι αριθμός ariadne. Αριθμός ariadne ορίζεται ο αριθμός που το άθροισμα του πιο σημαντικού (ψηφίο στα αριστερά) και του λιγότερου σημαντικού (ψηφίο στα δεξιά) ψηφίου ισούται με το άθροισμα των υπόλοιπων ψηφίων (π.χ. 121, 1335). Αν είναι αριθμός ariadne, τότε να αποθηκεύεται στον πίνακα P η τιμή true διαφορετικά αποθηκεύεται η τιμή false.
- (3) Να τυπώνει τους 2 πίνακες, τον έναν δίπλα στον άλλο.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 αριθμούς)	Παράδειγμα Εξόδου (οθόνη) (για 5 αριθμούς)
131 121 1302 4567	Δώσε τους αριθμούς: Αριθμός Ariadne 131 false 121 true 1302 true 4567 false

Άσκηση 5.29

Ο κανονισμός για την ελλιπή φοίτηση στα λύκεια είναι σαφής. Ειδικά για το Β' Τετράμηνο, ο κανονισμός αναφέρει ότι παραπέμπονται σε ανεξετάσεις τον Σεπτέμβριο όσοι έχουν αριθμό αδικαιολόγητων απουσιών μεταξύ 21-25 συμπεριλαμβανομένων, ή στο σύνολο (δικαιολογημένες, αδικαιολόγητες) 76-80 συμπεριλαμβανομένων. Επίσης, παραπέμπονται σε όσα μαθήματα συγκεντρώσουν επταπλάσιο αριθμό απουσιών από τις ώρες διδασκαλίας του μαθήματος.

Η γραμματεία έχει δημιουργήσει την πιο κάτω αναφορά για 300 μαθητές, η οποία παρουσιάζει αναλυτικά τα εξής: Αριθμός μητρώου μαθητή, αδικαιολόγητες απουσίες Β' τετράμηνου, συνολικές απουσίες Β' τετράμηνου και επταπλάσιο. Ενδεικτικά, η αναφορά εμφανίζεται όπως πιο κάτω. Συγκεκριμένα για το επταπλάσιο, η αναφορά τυπώνει 1, αν ο μαθητής έχει συμπληρώσει το επταπλάσιο σε κάποιο μάθημα, αλλιώς τυπώνει 0:

Αρ. Μητρώου	Αδικαιολόγητες	Σύνολο	Επταπλάσιο
1001	30	90	1
1002	20	56	0
1003	19	93	0
1004	24	80	1
1005	12	30	0
.	.	.	.
1300	9	45	0

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται τον αριθμό μητρώου, τον αριθμό των αδικαιολόγητων και το σύνολο των απουσιών, καθώς επίσης και το επταπλάσιο για τον κάθε μαθητή και να αποθηκεύει τα στοιχεία αυτά σε παράλληλους μονοδιάστατους πίνακες.
- (2) Να υπολογίζει και να τυπώνει τον αριθμό μητρώου του μαθητή με το μεγαλύτερο σύνολο απουσιών.
- (3) Να υπολογίζει και να τυπώνει στην οθόνη τους Αριθμούς Μητρώου όλων των μαθητών που παραπέμπονται, κατά το Β' τετράμηνο, καθώς επίσης, να τυπώνει και το πλήθος αυτών των μαθητών.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 5 μαθητές)				Παράδειγμα Εξόδου (οθόνη) (για 5 μαθητές)
1001	3	90	1	Δώσε τα στοιχεία των μαθητών: Μαθητής με μεγαλύτερο σύνολο απουσιών 1003
1002	20	56	0	Μαθητές που παραπέμπονται: 1001
1003	19	93	0	1003
1004	24	80	1	1004
1005	12	30	0	Πλήθος μαθητών=3

Άσκηση 5.30

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται 100 ακέραιους αριθμούς και τους αποθηκεύει στον πίνακα nums.

- (2) Να υπολογίζει και να τυπώνει το πλήθος για: θετικούς, αρνητικούς και μηδενικά.
- (3) Να δημιουργεί τον πίνακα filter ως εξής: Αν το nums[i] είναι θετικός αριθμός που δεν διαιρείται με το 3, τότε filter[i]=0 διαφορετικά filter[i]=1.
- (4) Να τυπώνει όλους τους πίνακες χρησιμοποιώντας τις κατάλληλες κεφαλίδες.

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) (για 5 αριθμούς) 5 6 0 56 23	Παράδειγμα Εξόδου 1 (οθόνη) (για 5 αριθμούς) Δώσε τους αριθμούς: Πλήθος θετικών= 4 Πλήθος αρνητικών= 0 Πλήθος μηδενικών= 1 Αριθμός Πολλαπλάσιο του 3 5 0 6 1 0 0 56 0 23 0
--	---

Άσκηση 5.31

Να γράψετε πρόγραμμα το οποίο να δέχεται 20 τυχαίους ακέραιους αριθμούς και να τους αποθηκεύει σε ένα μονοδιάστατο πίνακα με το όνομα **Num**. Στη συνέχεια, να αναζητά και να τυπώνει πόσες φορές (πλήθος) υπάρχει αριθμός στον πίνακα, όπου το διπλάσιό του είναι ίσο με το άθροισμα των δύο αριθμών που βρίσκονται πριν και μετά από αυτόν (προηγούμενος και επόμενος αριθμός).

Επεξήγηση:

Σύμφωνα με τον πιο κάτω πίνακα, οι αριθμοί 6 ($2*6=12$ και $3+9=12$) και 7 ($2*7=14$ και $5+9=14$) πληρούν τις πιο πάνω προϋποθέσεις. Άρα το πρόγραμμα θα πρέπει να τυπώνει: Πλήθος = 2.

Πίνακας Num	5	8	9	15	3	6	9	11	2	1	8	9	12	0	21	5	7	9	5	4
-----------------------	---	---	---	----	---	---	---	----	---	---	---	---	----	---	----	---	---	---	---	---

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) 5 8 9 15 3 6 9 11 2 1 8 9 12 0 21 5 7 9 5 4	Παράδειγμα Εξόδου 1 (οθόνη) Δώσε τους αριθμούς: Πλήθος=2
--	---

Άσκηση 5.32

Να γράψετε πρόγραμμα το οποίο να καταχωρεί σε ένα μονοδιάστατο πίνακα 20 θέσεων με το όνομα **voles** τα αποτελέσματα των 20 βολών που έκανε ένας αθλητής στο αγώνισμα της σκοποβολής. Στις βολές που δεν βρήκε τον στόχο καταχωρείται η τιμή 0 και σε αυτές που βρήκε τον στόχο η τιμή 1. Στη συνέχεια, το πρόγραμμα να υπολογίζει και να τυπώνει τον μεγαλύτερο αριθμό συνεχόμενων εύστοχων βολών.

Επεξήγηση:

Πίνακας voles

1	1	0	1	0	0	1	1	1	1	1	0	1	1	1	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Στο παράδειγμα αυτό η συνάρτηση θα επιστρέψει την τιμή 5 (πέντε συνεχόμενες εύστοχες βολές).

Παράδειγμα Εισόδου 1 (πληκτρολόγιο)	Παράδειγμα Εξόδου 1 (οθόνη)
1 1 0 1 0 0 1 1 1 1 1 0 1 1 1 1 0 1 1 0	Δώσε τους αριθμούς: Πλήθος=5

Άσκηση 5.33

Να γράψετε πρόγραμμα το οποίο:

- (1) Να δέχεται 10 ακέραιους αριθμούς και να τους αποθηκεύει στον πίνακα A.
- (2) Να υπολογίζει και να τυπώνει τον μεγαλύτερο και τον μικρότερο αριθμό του πίνακα A.
- (3) Να δέχεται έναν ακέραιο αριθμό S ($0 < S \leq 10$) και να εκτελεί S κυκλικές μετακινήσεις. Σε μία κυκλική μετακίνηση ο αριθμός που βρίσκεται στη θέση 0 θα μετακινηθεί στη θέση 1, ο αριθμός που βρίσκεται στη θέση 1 θα μετακινηθεί στη θέση 2 κλπ. Ο αριθμός που βρίσκεται στην τελευταία θέση του πίνακα θα μετακινηθεί στην θέση 0 (1η θέση) του πίνακα. Αφού εκτελεστούν οι κυκλικές μετακινήσεις να εμφανίζεται ο πίνακας στην οθόνη.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
9 2 3 4 1 -7 1 1 6 5 3	Δώσε 10 ακέραιους αριθμούς Μεγαλύτερος=9 Μικρότερος=-7 Δώσε αριθμό κυκλικών μετακινήσεων Πίνακας A 1 6 5 9 2 3 4 1 -7 1

Άσκηση 5.34

Να γράψετε πρόγραμμα το οποίο να καταχωρίζει σε πίνακα 20 ακέραιους αριθμούς και να τυπώνει το μήνυμα «SORTED», αν ο πίνακας είναι ταξινομημένος σε αύξουσα σειρά, αλλιώς να τυπώνει το μήνυμα «NOT SORTED».

Παράδειγμα Εισόδου 1 (πληκτρολόγιο) (για 5 αριθμούς)	Παράδειγμα Εξόδου 1 (οθόνη) (για 5 αριθμούς)
4 131 234 456 1234	Δώσε τους αριθμούς: SORTED

Παράδειγμα Εισόδου 2 (πληκτρολόγιο) (για 5 αριθμούς) 131 4 34 56 34	Παράδειγμα Εξόδου 2 (οθόνη) (για 5 αριθμούς) Δώσε τους αριθμούς: NOT SORTED
---	--



Άσκηση 5.35

Ο αλγόριθμος RLE (Run Length Encoding) αποτελεί μία από τις πιο απλές τεχνικές συμπίεσης δεδομένων. Η τεχνική αυτή ελέγχει όλη την ακολουθία των χαρακτήρων, που αποτελούν τα δεδομένα προς συμπίεση και εντοπίζει τις διαδοχικές επαναλήψεις του ίδιου χαρακτήρα. Στη συνέχεια, αντικαθιστά τις συνεχόμενες επαναλήψεις χαρακτήρων με το πλήθος τους, ακολουθούμενο από τον χαρακτήρα. Να θεωρήσετε ότι τα δεδομένα που δίνονται δημιουργούν πάντοτε αποτέλεσμα με μικρότερο αριθμό χαρακτήρων.

Παράδειγμα: Αν δοθεί η ακολουθία χαρακτήρων AAAADCCCBEEEEAA για συμπίεση με τη μέθοδο RLE, τότε το αποτέλεσμα της συμπίεσης θα είναι η ακολουθία 4A1D3C2B3E2A, που έχει μήκος 12 χαρακτήρες αντί 15 που είχε η αρχική.

Να γράψετε πρόγραμμα το οποίο να δέχεται από τον χρήστη μία ακολουθία 15 χαρακτήρων και στη συνέχεια να τυπώνει την ακολουθία συμπιεσμένη, σύμφωνα με τον αλγόριθμο RLE.

(Παγκύπριες εξετάσεις 2015)

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 7 χαρακτήρες) aaaabbbbccccccdd	Παράδειγμα Εξόδου (οθόνη) (για 7 χαρακτήρες) Δώσε τους χαρακτήρες: Συμπιεσμένη συμβολοσειρά: 4a3b6c2d
---	--



Άσκηση 5.36

Ένας φίλος σας έχει στείλει τον κωδικό του Wi-Fi του στο ηλεκτρονικό ταχυδρομείο. Για σκοπούς ασφαλείας έχει κρύψει τον κωδικό μέσα σε μία ακολουθία μέχρι 1000 χαρακτήρων. Η ακολουθία αποτελείται από κεφαλαία και πεζά γράμματα, αριθμούς και σύμβολα. Να γράψετε πρόγραμμα το οποίο να δέχεται το πλήθος των χαρακτήρων και τους χαρακτήρες της ακολουθίας, να καταχωρίζει τα περιεχόμενά της σε πίνακα και να τυπώνει τον κωδικό.

Σημείωση: Ο κωδικός αποτελείται μόνο από ψηφία και για να ανήκει ένα ψηφίο στον κωδικό, πρέπει να βρίσκεται μεταξύ των συμβόλων '>' και '<'.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 7 χαρακτήρες) 26 aC2*&3>1<42/>2<0oP>3<>4<89	Παράδειγμα Εξόδου (οθόνη) (για 7 χαρακτήρες) Δώσε το πλήθος των χαρακτήρων: Δώσε τους χαρακτήρες της ακολουθίας: Κωδικός:1234
---	--

Άσκηση 5.37

Να γράψετε πρόγραμμα το οποίο να δέχεται το πλήθος των χαρακτήρων N ($1 < N \leq 100$) και τους χαρακτήρες μιας ακολουθίας και να τυπώνει τον χαρακτήρα που εμφανίζεται τις περισσότερες φορές μέσα σε αυτή. Οι χαρακτήρες της ακολουθίας είναι τα πεζά γράμματα του λατινικού αλφαβήτου (από το a μέχρι το z συμπεριλαμβανομένων).

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 11 χαρακτήρες) 11 abracatabra	Παράδειγμα Εξόδου (οθόνη) (για 11 χαρακτήρες) Δώσε το πλήθος των χαρακτήρων: Δώσε τους χαρακτήρες της ακολουθίας: Χαρακτήρας με μέγιστο αριθμό εμφανίσεων: a
---	--

Άσκηση 5.38

Να γράψετε πρόγραμμα το οποίο να δέχεται το πλήθος N ($0 < N < 25$) και στη συνέχεια N ακέραιους αριθμούς (από το 0 μέχρι το 9) και να τυπώνει στην οθόνη το οριζόντιο ιστόγραμμα, το οποίο αναπαριστά το πλήθος εμφανίσεων του κάθε ψηφίου. Το οριζόντιο ιστόγραμμα παρουσιάζει ένα αστερίσκο για κάθε εμφάνιση του ψηφίου.

Παράδειγμα Εισόδου (πληκτρολόγιο) (για 8 αριθμούς) 8 1 1 2 7 6 1 6 1	Παράδειγμα Εξόδου (οθόνη) (για 8 αριθμούς) Δώσε το πλήθος των αριθμών: Δώσε τους αριθμούς: Οριζόντιο Ιστόγραμμα 0 1***** 2* 3 4 5 6** 7* 8 9
---	--

Άσκηση 5.39

Τα παιδιά της γειτονιάς έχουν μαζευτεί στο πάρκο για τον καθιερωμένο, απογευματινό αγώνα ποδοσφαίρου. Οι δύο αρχηγοί έχουν να επιλέξουν μεταξύ δέκα παιδιών και κάθε ομάδα θα έχει στη σύνθεσή της πέντε παίκτες. Όλα τα παιδιά έχουν ένα επίπεδο ικανότητας L ($1 \leq L \leq 100$). Οι δέκα παίκτες στέκονται σε σειρά, με πρώτον αυτόν που έχει το χαμηλότερο επίπεδο ικανότητας και τελευταίο αυτόν με το μεγαλύτερο. Οι αρχηγοί ξεκινούν εκ περιτροπής να επιλέγουν παίκτες, προσπαθώντας να στήσουν την πιο καλή ομάδα. Ο πρώτος αρχηγός θα επιλέξει τον παίκτη με το ψηλότερο επίπεδο ικανότητας, ο δεύτερος

αρχηγός τον παίκτη με το δεύτερο ψηλότερο επίπεδο ικανότητας, ο πρώτος αυτόν με το τρίτο, ο δεύτερος αυτόν με το τέταρτο κ.ο.κ. Να γράψετε πρόγραμμα το οποίο να δέχεται το επίπεδο ικανότητας και για τους δέκα παίκτες και να τυπώνει το συνολικό επίπεδο ικανότητας για την καλύτερη ομάδα που θα σχηματιστεί.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
5 10 20 30 40 50 60 70 80 90	Δώσε τους αριθμούς: Συνολικό επίπεδο ικανότητας καλύτερης ομάδας=250



Άσκηση 5.40

Ο Παντελής προγραμματίζει τις καλοκαιρινές του διακοπές από νωρίς. Διακαής του πόθος είναι να επισκεφτεί το Ντουμπάι. Στο Ντουμπάι τα παραθαλάσσια ξενοδοχεία ξεφυτρώνουν, σχεδόν καθημερινά, από το πουθενά. Ο Παντελής επιθυμεί να κλείσει ένα δωμάτιο με θέα τη θάλασσα. Θέλει να ξέρει πόσες θα είναι οι επιλογές του.

Ο Παντελής έχει βρει ένα διαφημιστικό φυλλάδιο, όπου διακρίνει τον αριθμό των παραλιακών ξενοδοχείων, αλλά και το ύψος σε ορόφους του κάθε ξενοδοχείου. Η θάλασσα βρίσκεται στα δεξιά της σειράς των ξενοδοχείων, άρα ένα ξενοδοχείο με 10 ορόφους, που βρίσκεται στα αριστερά ενός ξενοδοχείου με 20 ορόφους, δεν μπορεί να έχει θέα. Αντίθετα, ένα ξενοδοχείο 50 ορόφων, που βρίσκεται στα αριστερά του ξενοδοχείου με τους 10 ορόφους, θα έχει δωμάτια με θέα τη θάλασσα. Μπορείτε να βοηθήσετε τον Παντελή να βρει πόσα ξενοδοχεία έχουν δωμάτια με θέα τη θάλασσα, για να μπορέσει να κλείσει έγκαιρα τις διακοπές του;

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
5 20 45 25 30 15	Δώσε τον αριθμό των ξενοδοχείων: Δώσε το ύψος σε ορόφους: Αριθμός ξενοδοχείων=3

+ Ασκήσεις Εμπλουτισμού



Άσκηση 5.41

Κατά την διάρκεια της θητείας του, ένας πεζοναύτης θα κληθεί να ανταπεξέλθει με επιτυχία σε μία σειρά από επίπονες και απαιτητικές δραστηριότητες, που σκοπό έχουν να δοκιμάσουν τα όρια της ψυχικής και φυσικής του αντοχής. Από τις βασικότερες ασκήσεις είναι η άσκηση «ΔΡΟΜΕΑΣ». Η άσκηση αυτή περιλαμβάνεται στο πρόγραμμα εκπαίδευσης και των υπόλοιπων Ειδικών Δυνάμεων (Καταδρομείς, Αλεξιπτωτιστές κ.λπ.) και είναι, ίσως, η πιο δύσκολη άσκηση από ψυχοσωματικής άποψης.

Η επανομασία της άσκησης βασίζεται στην εξής διαδικασία, που λαμβάνει χώρα κατά τη διάρκεια της πορείας. Καθώς προχωρούν, οι πεζοναύτες δέχονται από τον εκπαιδευτή τους (Κ φορέας) τη διαταγή «ΔΡΟΜΕΑΣ». Μόλις δοθεί η διαταγή, ο πεζοναύτης που βρίσκεται στην

τελευταία θέση της φάλαγγας τρέχει και προσπερνά τον πεζοναύτη που βρίσκεται στην πρώτη θέση και ηγείται τώρα αυτός της φάλαγγας. Όταν ολοκληρώσει τις K διαταγές, ο εκπαιδευτής καλεί (Q φορές) κάποιους από τους πεζοναύτες να αναφερθούν, με βάση τη θέση τους στη φάλαγγα, για να διαπιστώσει αν όλοι γνωρίζουν επακριβώς σε ποια θέση βρίσκονται. Να γράψετε πρόγραμμα το οποίο να τυπώνει τους στρατιώτες της φάλαγγας, οι οποίοι πρέπει να αναφερθούν στον εκπαιδευτή.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
3 2 3 21 12 33 0 1 2	Δώσε αριθμό πεζοναυτών: Δώσε αριθμό διαταγών: Δώσε αριθμό κλήσεων: Δώσε αριθμούς πεζοναυτών: Δώσε σειρά αναφοράς: Οι πεζοναύτες θα δώσουν αναφορά ως εξής: 12 33 21

Επεξήγηση παραδείγματος

Έχουμε 3 πεζοναύτες, στους οποίους θα δοθεί η διαταγή «ΔΡΟΜΕΑΣ» 2 φορές και ακολούθως, θα δεχτούν 3 κλήσεις για αναφορά. Οι πεζοναύτες ξεκινούν αρχικά την πορεία με τον πεζοναύτη 21 στην πρώτη θέση, τον 12 στη δεύτερη και τον 33 στην τρίτη. Μετά την πρώτη διαταγή «ΔΡΟΜΕΑΣ», οι θέσεις των πεζοναυτών θα είναι 33, 21, 12, ενώ μετά και τη δεύτερη διαταγή οι τελικές τους θέσεις θα είναι 12, 33, 21. Ο πεζοναύτης που θα κληθεί να δώσει αναφορά πρώτος, είναι αυτός που βρίσκεται στη θέση 0, δηλαδή ο πεζοναύτης 12. Ο πεζοναύτης που θα κληθεί να δώσει αναφορά δεύτερος είναι αυτός που βρίσκεται στη θέση 1, δηλαδή ο 33. Τέλος, ο πεζοναύτης που θα κληθεί να δώσει αναφορά τρίτος είναι αυτός που βρίσκεται στη θέση 2, δηλαδή ο 21.



Άσκηση 5.42

Η Αριάδνη θέλει να επισκεφθεί τον Κυριάκο, τον παλιό συμμαθητή της, για πρώτη φορά μετά από πολλά χρόνια. Θέλοντας να του κάνει έκπληξη, δεν έχει επικοινωνήσει προηγουμένως μαζί του τηλεφωνικά και αντιμετωπίζει τώρα ένα πρόβλημα. Ενώ θυμάται την οδό που μένει ο Κυριάκος, δεν θυμάται τον αριθμό του σπιτιού του. Θυμάται, όμως, ότι ο αριθμός του σπιτιού του Κυριάκου είναι διπλάσιος από κάποιο άλλο σπίτι της ίδιας οδού και, ταυτόχρονα, ο μισός κάποιου άλλου σπιτιού της ίδιας οδού. Αν σας δίνεται ο συνολικός αριθμός των σπιτιών μαζί με τον αριθμό του κάθε σπιτιού, να βοηθήσετε την Αριάδνη να βρει το σπίτι του Κυριάκου.

Δεδομένα εισόδου

- Ένας ακέραιος αριθμός N ($3 \leq N \leq 50$), ο συνολικός αριθμός των σπιτιών στην οδό του Κυριάκου.
- N ακέραιοι αριθμοί K ($1 \leq K \leq 250$), ο αριθμός του κάθε σπιτιού. Δύο σπίτια δεν μπορούν να έχουν τον ίδιο αριθμό.

Δεδομένα εξόδου

Ο αριθμός του σπιτιού του Κυριάκου.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
5 1 26 13 52 32	Δώσε αριθμό σπιτιών: Δώσε αριθμούς σπιτιών: Ο αριθμός του σπιτιού είναι το:26

**Άσκηση 5.43**

Ο Μήτσος θέλει να κερδίσει το Τζόκερ. Εδώ και μερικά χρόνια, κάθε εβδομάδα που γίνεται η κλήρωση, κρατά ένα αρχείο με τους αριθμούς που έχουν κληρωθεί. Σκοπός του είναι να βρει ποιοι αριθμοί εμφανίζονται συχνότερα στις κληρώσεις, για να μπορέσει να τους επιλέξει στα επόμενα δελτία. Αν σας δοθούν N ακέραιοι αριθμοί, να βρείτε ποιος από αυτούς τους αριθμούς εμφανίζεται τις περισσότερες φορές, καθώς και το πλήθος των εμφανίσεών του. Αν κάποιοι αριθμοί έχουν το ίδιο πλήθος εμφανίσεων, να εμφανίζεται ο μικρότερος.

Δεδομένα εισόδου

- Ένας ακέραιος N ($1 \leq N \leq 10000$), το πλήθος των αριθμών.
- N ακέραιοι αριθμοί x ($1 \leq x \leq 49$).

Δεδομένα εξόδου

Ο αριθμός με τις περισσότερες εμφανίσεις και το πλήθος των εμφανίσεών του.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
7 36 43 36 28 36 12 4	Δώσε πλήθος αριθμών: Δώσε αριθμούς: Αριθμός= 36 Πλήθος εμφανίσεων= 3

Άσκηση 5.44

Δίνονται N ($0 < N \leq 10000$) αριθμοί, όλοι διαφορετικοί μεταξύ τους. Να υπολογίσετε τα συνολικά ζεύγη αριθμών που έχουν διαφορά K ($0 < K < 1000$).

Δεδομένα εισόδου

- Στην πρώτη γραμμή θα εμφανίζονται οι αριθμοί N και K .
- N ακέραιοι αριθμοί, όλοι διαφορετικοί μεταξύ τους.

Δεδομένα εξόδου

- Ένας ακέραιος, το πλήθος των ζευγαριών από αριθμούς με διαφορά K .

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
5 2 1 5 3 4 2	Δώσε πλήθος αριθμών: Δώσε διαφορά: Δώσε αριθμούς: Αριθμός ζευγαριών= 3



Άσκηση 5.45

Στο νηπιαγωγείο «Μέλι – Κουκουμέλι», τα παιδιά ετοιμάζονται να πάνε βόλτα στο γειτονικό πάρκο. Η δασκάλα έχει δώσει σαφείς οδηγίες στα παιδιά ότι στον δρόμο πρέπει να κρατούν το χέρι κάποιου άλλου παιδιού και να μην το αφήσουν κατ' ουδέν λόγο. Το πρόβλημα είναι ότι το σύνολο των παιδιών του νηπιαγωγείου είναι περιττός αριθμός, οπότε κάποιος παιδάκι θα πρέπει να κρατά το χέρι της δασκάλας. Για αποφυγή παρεξηγήσεων, αφού κανένα παιδί δεν θέλει να κρατά το χέρι της, η δασκάλα έχει βάλει αριθμημένα χαρτάκια σε ένα κουτί και κάθε παιδί, με τη σειρά, τραβά ένα χαρτάκι. Μέσα στο κουτί, η δασκάλα έχει βάλει N χαρτάκια. Κάθε χαρτάκι με αριθμό έχει το αντίστοιχο ζευγάρι του, εκτός από ένα χαρτάκι που ο αριθμός που αναγράφεται σε αυτό εμφανίζεται μόνο μία φορά. Το παιδί που θα τραβήξει αυτό το χαρτάκι, με τον μοναδικό αριθμό, θα κρατά το χέρι της δασκάλας. Βρείτε ποιος αριθμός είναι αυτός.

Δεδομένα εισόδου

- Ένας ακέραιος, περιττός αριθμός N ($0 <= N <= 100$), το πλήθος των χαρτιών που βρίσκονται στο κουτί.
- Ακολουθούν N ακέραιοι αριθμοί στο διάστημα $[0..100]$, οι αριθμοί που αναγράφονται στο κάθε χαρτάκι. Όλοι οι αριθμοί εμφανίζονται δύο φορές εκτός από έναν, ο οποίος εμφανίζεται μόνο μία.

Παράδειγμα Εισόδου (πληκτρολόγιο)	Παράδειγμα Εξόδου (οθόνη)
5 17 0 5 0 17	Δώσε πλήθος χαρτιών: Δώσε αριθμούς: Αριθμός= 5

B7.2 Δημιουργία Παιγνιδιών για Ηλεκτρονικούς Υπολογιστές

B7.2.1 Δημιουργία παιγνιδιών για Ηλεκτρονικούς Υπολογιστές

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ◈ Να περιγράφουμε την χρησιμότητα και την επίδραση των παιγνιδιών στη ζωή μας
- ◈ Να αναφέρουμε/αναγνωρίζουμε τα είδη των παιγνιδιών
- ◈ Να περιγράφουμε τις επισημάνσεις PEGI
- ◈ Να αναφέρουμε την ιστορία των παιγνιδιών
- ◈ Να περιγράφουμε τον κύκλο ζωής ενός βιντεοπαιγνιδιού.

1. Εισαγωγή

Σήμερα τα βιντεοπαιχνίδια αποτελούν μέσο ψυχαγωγίας για κάθε ηλικία και φύλο, με σχεδόν ολοκληρωτική διείσδυση στο νεανικό κοινό και τεράστια απήχηση στους ενήλικες. Είναι διαθέσιμα σε κάθε μορφής ψηφιακή συσκευή – από τις κονσόλες μέχρι τα κινητά τηλέφωνα – και αποτελούν μία ισχυρή βιομηχανία, που συναγωνίζεται σε κέρδη ακόμα τη βιομηχανία του κινηματογράφου. Επιπλέον, αλλάζουν σταδιακά τον τρόπο με τον οποίο επικοινωνούμε, αλληλεπιδρούμε, μαθαίνουμε και εργαζόμαστε – και, για αυτόν το λόγο, αποτελούν αντικείμενο ακαδημαϊκής έρευνας αλλά και κριτικής και κοινωνικής ανησυχίας.

2. Παιχνίδια και εκπαίδευση

2.1 Εκπαίδευση

Σε όλο τον κόσμο οι εκπαιδευτικοί ενδιαφέρονται όλο και περισσότερο για τον δυνητικό ρόλο των παιγνιδιών υπολογιστή και βίντεο όσον αφορά στην υποστήριξη της μάθησης των νεαρών ατόμων. Πρόσφατες μελέτες δείχνουν ότι, όταν παίζουν παιχνίδια, τα παιδιά επιδίδονται σταθερά σε δραστηριότητες μάθησης, οι οποίες είναι πολύ πιο περίπλοκες και συναρπαστικές από το μεγαλύτερο μέρος των κανονικών σχολικών εργασιών τους.

2.2 Ικανότητες

Ακόμα και παιχνίδια χωρίς κανέναν προφανή εκπαιδευτικό σκοπό απαιτούν από τους παίκτες να καλλιεργήσουν πολλαπλές δεξιότητες. Τα παιχνίδια είναι σύνθετα, απαιτούν προσαρμοστικότητα και αναγκάζουν τους παίκτες να αναπτύξουν ένα ευρύ φάσμα ικανοτήτων. Να σημειωθεί ότι οι ερευνητές έχουν καταγράψει πολυάριθμα οφέλη, μεταξύ των οποίων η αύξηση της κοινωνικό-αισθηματικής, γνωστικής και φυσικής ανάπτυξης του παιδιού, καθώς και το ακόνισμα των, λεπτών και μη, κινητικών ικανοτήτων, του συντονισμού χεριού-ματιού, της δημιουργικότητας, της συγκέντρωσης, καθώς και των μηχανισμών επίλυσης προβλημάτων.

2.3 Κοινωνική αλληλεπίδραση

Πολλοί άνθρωποι πιστεύουν ότι τα ηλεκτρονικά παιχνίδια αποτελούν μία μοναχική δραστηριότητα. Η πραγματικότητα, όμως, είναι πολύ διαφορετική. Τα παιδιά σπάνια παίζουν μόνο τους, καθώς οι περισσότερες κονσόλες σήμερα διαθέτουν δυνατότητες παιγνιδιού πολλαπλών παικτών. Σε μία κονσόλα μπορούν να παίξουν μαζί δύο έως τέσσερις παίκτες, όμως όλο και περισσότερα παιχνίδια επιτρέπουν στους παίκτες να αλληλεπιδρούν με άλλους μέσω Διαδικτύου και, συνεπώς, ο αριθμός των διαδικτυακών κοινοτήτων παιγνιδιού αυξάνεται ταχύτατα. Η κοινωνική αλληλεπίδραση αυτών των παιγνιδιών μπορεί να βοηθήσει τους παίκτες να έρθουν πιο κοντά.

Πηγή: Παιδαγωγικό Ινστιτούτο Κύπρου (<http://www.pi.ac.cy>)

Γνωρίζετε ότι τα παιχνίδια...

- Καλλιεργούν τη δημιουργικότητα και την αλληλεπίδραση και μπορούν να διαδραματίσουν σημαντικό ρόλο στην κοινωνική και πνευματική ανάπτυξη.
- Βοηθούν τους νεοεισερχόμενους στην τεχνολογία και καλλιεργούν το ενδιαφέρον για τις ΤΠΕ (Τεχνολογίες της Πληροφορίας και της Επικοινωνίας).
- Απαιτούν από τα παιδιά να υπακούουν σε κανόνες και να ακολουθούν οδηγίες, αυξάνοντας τις ικανότητές τους για αυτοπειθαρχία και αυτονομία.
- Προσφέρουν ευκαιρίες στους παίκτες, για να αναπτύξουν τη στρατηγική τους σκέψη και τις ικανότητές τους στην επίλυση προβλημάτων, ειδικά τα παιχνίδια γρίφων, τα επιτραπέζια, τα παιχνίδια περιπέτειας και αναζήτησης.
- Αυξάνουν τις ικανότητες λεπτών κινήσεων και αίσθησης του χώρου στα μικρότερα παιδιά και παρουσιάζουν πλεονεκτήματα για τα παιδιά με σωματικές αναπηρίες.

Πηγή: Pan-European Games Information (www.pegi.info)

**3. Είδη παιχνιδιών**

Υπάρχουν πολλά είδη παιχνιδιών, τα οποία κατηγοριοποιούνται με βάση τα κύρια χαρακτηριστικά τους.

Παιγνίδι Περιπέτειας (Adventure): Οι παίκτες λαμβάνουν ρόλους και το παιχνίδι βασίζεται στην επίλυση γρίφων, π.χ. Monkey Island, Broken Sword.

Παιγνίδι Δράσης και Περιπέτειας (Action Adventure): Ο παίκτης ελέγχει έναν χαρακτήρα σε δραστηριότητες επίλυσης γρίφων και/ή μάχης, π.χ. Tomb Raider.

Παιγνίδι Πλατφόρμας (Platform/Platformer): Διαδραματίζεται γύρω από παίκτες που πηδάνε από πλατφόρμα σε πλατφόρμα, π.χ. Donkey Kong.

Παιγνίδι Γρίφου (Puzzle): Εύκολο στη χρήση, συνήθως διατίθεται σε κονσόλες χειρός ή σε ηλεκτρονικούς υπολογιστές μέσω του Διαδικτύου, π.χ. Tetris, Zoo Keeper.

Παιγνίδι Ρόλων (Role Playing Game, RPG): Παιγνίδι όπου ο παίκτης ελέγχει έναν χαρακτήρα/ρόλο, ο οποίος έχει συγκεκριμένες ικανότητες και ρόλο στην πλοκή του παιχνιδιού, η οποία διαδραματίζεται σε γύρους, π.χ. Final Fantasy.

Παιγνίδι Αγώνων Ταχύτητας (Racer): Ένα παιγνίδι οδήγησης, το οποίο περιλαμβάνει την ολοκλήρωση διαδρομών σε δεδομένο χρόνο ή ενάντια σε άλλους διαγωνιζομένους, π.χ. Burnout, Gran Turismo.

Παιγνίδι Ρυθμού-Χορού (Rhythm-Dance): Παιγνίδι όπου ο παίκτης κινείται παράλληλα με το παιγνίδι/τη μουσική, π.χ. Singstar.

Shoot 'em Up: Παιγνίδι όπου ο παίκτης γυρνάει πυροβολώντας αντικείμενα, με στόχο να κερδίσει βαθμούς ή να ολοκληρώσει ένα στάδιο του παιχνιδιού και να προχωρήσει στο επόμενο, π.χ. Halo.

Προσομοίωση (Simulation, Sim): Παιγνίδι το οποίο προσομοιώνει κάποια πραγματική δραστηριότητα (χτίσιμο πόλης, οικοσυστήματα, κ.λπ.), τη ζωή σε κάποιον (φανταστικό ή πραγματικό περιβάλλον), τη λειτουργία και χρήση κάποιου οχήματος (οδήγηση συγκεκριμένου μοντέλου αεροπλάνου, ελικοπτέρα, αυτοκινήτου, κ.λπ.), π.χ. Microsoft Flight Simulator.

Αθλητικό παιγνίδι (Sports): Προσομοίωση κάποιου αθλήματος, π.χ. Pro Evolution Soccer ή Tiger Woods Golf.

Παιγνίδι Στρατηγικής (Strategy): Σε μία εξέλιξη των αρχών που διέπουν τα επιτραπέζια παιγνίδια στρατηγικής, οι παίκτες, καθένας με τη σειρά του, πραγματοποιούν ελιγμούς, π.χ. Command & Conquer, Total War.

Online παιγνίδι: Τα online παιγνίδια έχουν το χαρακτηριστικό ότι παίζονται μέσω Διαδικτύου. Έχουν πολλές μορφές, οι οποίες ποικίλουν από απλά παιγνίδια arcade μέχρι και παιγνίδια ρόλων. Τα online παιγνίδια συχνά αγοράζονται σε δίσκο και μπορούν να παιχτούν μαζί/ενάντια σε άλλους παίκτες, μέσω ηλεκτρονικού υπολογιστή ή κονσόλας παιγνιδιών με σύνδεση στο Διαδίκτυο. Εναλλακτικά διατίθενται για να τα «κατεβάσει» και να τα εγκαταστήσει ο παίκτης στον υπολογιστή ή στην κονσόλα του. Τα πιο απλά παιγνίδια μπορούν, επίσης, να παιχτούν σε δικτυακούς τόπους μέσω ενός προγράμματος περιήγησης στο Διαδίκτυο.

Online παιγνίδι μεγάλου αριθμού πολλαπλών παικτών (Massively Multiplayer Online, MMO): Παιγνίδι μέσω Διαδικτύου το οποίο συχνά περιλαμβάνει χιλιάδες παίκτες που παίζουν ταυτόχρονα και μπορούν να αλληλεπιδράσουν.

Σε αρκετές περιπτώσεις ένα παιγνίδι μπορεί να ανήκει σε περισσότερες από μία κατηγορίες. Για παράδειγμα ένα παιγνίδι ρόλων μπορεί να περιέχει στοιχεία δράσης και περιπέτειας, ενώ παράλληλα να παίζεται Online.





4. Σχετικά με το σύστημα ταξινόμησης PEGI


Το Πανερωπαϊκό Σύστημα Πληροφόρησης για τα Ηλεκτρονικά Παιγνίδια (PEGI) είναι ένα ευρωπαϊκό σύστημα ταξινόμησης των βιντεοπαιγνιδιών που δημιουργήθηκε για να βοηθήσει τους Ευρωπαίους καταναλωτές να κάνουν συνειδητές αποφάσεις για την αγορά ηλεκτρονικών παιγνιδιών. Τέθηκε σε χρήση τον Απρίλιο του 2003.

Οι επισημάνσεις PEGI εμφανίζονται στο μπροστινό και στο πίσω μέρος της συσκευασίας, προσδιορίζοντας ένα από τα ακόλουθα ηλικιακά επίπεδα: 3, 7, 12, 16 και 18. Επίσης, εμφανίζουν περιγραφικές ενδείξεις για τον λόγο που ένα παιχνίδι έχει ενταχθεί σε συγκεκριμένο ηλικιακό επίπεδο.




Προσφέρουν μία αξιόπιστη ένδειξη για την καταλληλότητα του περιεχομένου του παιχνιδιού όσον αφορά στην προστασία των ανηλίκων. Η ηλικιακή διαβάθμιση δεν λαμβάνει υπόψη το επίπεδο δυσκολίας ή τις ικανότητες που απαιτούνται για ένα παιχνίδι.

Οι επισημάνσεις οι οποίες σχετίζονται με τα ηλικιακά επίπεδα φαίνονται στον πιο κάτω πίνακα:

Επισήμανση	Περιγραφή
 PEGI 3	<p>Το περιεχόμενο των παιχνιδιών στα οποία δίνεται αυτή η διαβάθμιση κρίνεται κατάλληλο για όλες τις ηλικιακές ομάδες. Γίνεται αποδεκτή κάποιου βαθμού βία σε κωμικό πλαίσιο (συνήθως, μορφές βίας του τύπου που απαντάται στα κινούμενα σχέδια Μπαγκς Μπάνι ή Τομ & Τζέρι).</p> <p>Το παιδί δεν πρέπει να μπορεί να συνδέσει τον χαρακτήρα στην οθόνη με χαρακτήρες της πραγματικής ζωής. Πρέπει να ανήκουν πλήρως στον χώρο της φαντασίας. Το παιχνίδι δεν πρέπει να περιέχει ήχους ή εικόνες, που μπορούν να τρομάξουν ή να φοβίσουν τα μικρά παιδιά.</p>
 PEGI 7	<p>Οποιαδήποτε παιχνίδια που κανονικά θα είχαν διαβάθμιση 3, αλλά περιέχουν κάποιες, πιθανώς τρομακτικές σκηνές ή ήχους και μπορεί να θεωρηθούν κατάλληλα για αυτή την κατηγορία.</p>
 PEGI 12	<p>Σε αυτή την ηλικιακή κατηγορία κατατάσσονται βιντεοπαιχνίδια που δείχνουν βία ελαφρώς πιο παραστατικής φύσης προς φανταστικούς χαρακτήρες και/ή μη παραστατική βία προς χαρακτήρες με ανθρώπινη εμφάνιση ή αναγνωρίσιμα ζώα, καθώς και βιντεοπαιχνίδια που δείχνουν ελαφρώς πιο παραστατικές σκηνές γυμνού.</p> <p>Τυχόν χυδαία γλώσσα σε αυτή την κατηγορία πρέπει να είναι ήπια και να μην περιέχει επιφωνήματα σεξουαλικού περιεχομένου.</p>
 PEGI 16	<p>Αυτή η διαβάθμιση εφαρμόζεται όταν η απεικόνιση βίας (ή σεξουαλικής δραστηριότητας) φτάνει σε ένα επίπεδο, το οποίο φαίνεται ίδιο με αυτό που θα αναμενόταν στην πραγματική ζωή. Στα παιχνίδια με διαβάθμιση 16 μπορεί να περιέχεται πιο ακραίου βαθμού χυδαία γλώσσα, η ιδέα της χρήσης καπνού και ναρκωτικών και η απεικόνιση εγκληματικών δραστηριοτήτων.</p>

 <p>PEGI 18</p>	<p>Η ταξινόμηση στην κατηγορία για ενήλικες εφαρμόζεται όταν το επίπεδο της βίας είναι τέτοιο, ώστε να αφορά στην απεικόνιση ακραίας βίας και/ή να περιλαμβάνει στοιχεία συγκεκριμένων τύπων βίας. Ο προσδιορισμός της ακραίας βίας είναι πολύ δύσκολος, καθώς σε πολλές περιπτώσεις μπορεί να είναι κάτι αρκετά υποκειμενικό, αλλά γενικά μπορεί να προσδιοριστεί ως οι απεικονίσεις βίας, που δημιουργούν ένα αίσθημα αποστροφής στον θεατή.</p>
---	--

Οι περιγραφικές ενδείξεις στο πίσω μέρος της συσκευασίας αναφέρουν τους κύριους λόγους για τους οποίους ένα παιχνίδι έλαβε μία ορισμένη ηλικιακή διαβάθμιση. Υπάρχουν οκτώ τέτοιες περιγραφικές ενδείξεις: βία, χυδαία γλώσσα, φόβος, ναρκωτικά, σεξ, διακρίσεις, τζόγος και online παιχνίδι με άλλα άτομα.

Επισήμανση		Περιγραφή
	Χυδαία γλώσσα	Το παιχνίδι εμπεριέχει χυδαία γλώσσα ή εκφράσεις ακατάλληλες για το συγκεκριμένο ηλικιακό επίπεδο.
	Διακρίσεις	Το παιχνίδι απεικονίζει διακρίσεις ή εμπεριέχει υλικό που μπορεί να τις ενθαρρύνει.
	Ναρκωτικά	Το παιχνίδι απεικονίζει τη χρήση ναρκωτικών ή εμπεριέχει αναφορές σε αυτήν.
	Φόβος	Το παιχνίδι μπορεί να είναι τρομακτικό για το συγκεκριμένο ηλικιακό επίπεδο.
	Τζόγος	Το παιχνίδι παροτρύνει σε τζόγο ή τον διδάσκει.
	Σεξ	Το παιχνίδι απεικονίζει γυμνότητα και/ή σεξουαλική συμπεριφορά ή σεξουαλικές αναφορές.
	Βία	Το παιχνίδι εμπεριέχει απεικονίσεις βίας.
	Online	Το παιχνίδι μπορεί να παίζεται μέσω Διαδικτύου.

Πηγή: Pan-European Games Information (www.pegi.info)

5. Η ιστορία των ψηφιακών παιγνιδιών

5.1 Από τα ερευνητικά κέντρα στις κονσόλες κερμάτων (1960 – 1970)

Η ιστορία των ψηφιακών παιγνιδιών ξεκινά τη δεκαετία του 1950 σε ερευνητικά κέντρα μεγάλων πανεπιστημίων. Το πρώτο «επίσημο» παιγνίδι ήταν το Spacewar! και δημιουργήθηκε από μία ομάδα φοιτητών του MIT το 1961. Το παιγνίδι αυτό έβαζε δύο παίκτες να πολεμήσουν μεταξύ τους. Ο κάθε παίκτης είχε τον έλεγχο ενός διαστημοπλοίου, ενώ μία μαύρη τρύπα βρισκόταν στο κέντρο της οθόνης. Στη συνέχεια, το παιγνίδι κυκλοφόρησε μαζί με τους υπολογιστές της DEC, οι οποίοι και πάλι απευθύνονταν σε ερευνητικά κέντρα.

Τον Σεπτέμβριο του 1971, το παιγνίδι Galaxy Game εγκαταστάθηκε στο πανεπιστήμιο του Στάνφορντ. Βασίζονταν στο Spacewar! και ήταν το πρώτο, το οποίο χρησιμοποιούσε κέρματα, για να λειτουργήσει. Την ίδια χρονιά, ο Nolan Bushnell, που θα διαδραμάτιζε τεράστιο ρόλο στον χώρο των παιγνιδιών αργότερα, δημιούργησε το πρώτο εμπορικό παιγνίδι με τίτλο Computer Space. Η εταιρεία Nutting Associates αγόρασε το παιγνίδι αυτό και κατασκεύασε 1500 μηχανές, που λειτουργούσαν με κέρματα. Το 1972, ο Nolan Bushnell και ο Ted Dabney ίδρυσαν μία εταιρεία που την ονόμασαν «Atari» και στη συνέχεια δημιούργησαν το Pong, ένα παιγνίδι σταθμό στον χώρο της ψηφιακής ψυχαγωγίας. Το Pong ήταν το πρώτο παιγνίδι που είχε τεράστια επιτυχία στις συσκευές με κέρματα (arcades ή coin-ops, όπως ονομάστηκαν). Το Pong θύμιζε αρκετά το τένις, με 2 γραμμές να απεικονίζουν τους παίκτες και ένα τετράγωνο την μπάλα. Πωλήθηκαν περισσότερα από 19.000 μηχανήματα με το Pong, γεγονός που αποτέλεσε όχι μόνο τεράστια επιτυχία για την Atari, αλλά και την αρχή μίας νέας εποχής.

Τα video games μπήκαν στη χρυσή τους εποχή το 1978, με την κυκλοφορία του παιγνιδιού – σταθμού Space Invaders από την Taito. Τόσο τεράστια ήταν η επιτυχία αυτού του παιγνιδιού, που πολύ σύντομα έσπρωξε πολλές άλλες εταιρείες να αρχίσουν να κατασκευάζουν παιγνίδια και κονσόλες. Το 1979 η Atari κυκλοφόρησε το Asteroids, μία ακόμη επιτυχία, ενώ την επόμενη χρονιά κυκλοφόρησε και το έγχρωμο Pac-Man. Μέχρι το τέλος της δεκαετίας του 1980, λειτουργούσαν με κέρματα μεγάλες κονσόλες με παιγνίδια σε πάρα πολλά κέντρα ψυχαγωγίας, σε ολόκληρο σχεδόν τον κόσμο.

5.2 Οι πρώτες οικιακές κονσόλες (1970 – 1980)

Η εξάπλωση και η μεγάλη επιτυχία των κονσόλων κερμάτων οδήγησαν στη δημιουργία και των πρώτων συσκευών για το σπίτι. Ο Ralph Baer και οι συνεργάτες του δημιούργησαν ένα πρωτότυπο το 1968, το οποίο στη συνέχεια το βελτίωσαν και το κυκλοφόρησε η αμερικανική Magnavox το 1972. Η κονσόλα αυτή ονομάστηκε Magnavox Odyssey και μπορούσε να δεχτεί κάρτες (cartridges), μέσω των οποίων ο παίκτης μπορούσε να χρησιμοποιήσει διαφορετικά παιγνίδια (κυρίως παραλλαγές του ίδιου τίτλου). Αργότερα η Philips αγόρασε τη Magnavox και κυκλοφόρησε την κονσόλα της και στην Ευρώπη, με τεράστια επιτυχία και πωλήσεις πέραν των 2 εκατομμυρίων συσκευών.

Με την κυκλοφορία των πρώτων μικροεπεξεργαστών και τη συνεχή πτώση της τιμής τους, όλο και περισσότερες εταιρείες ξεκίνησαν να ενδιαφέρονται για την αγορά των παιγνιδιών. Η ανάπτυξη καρτών (cartridges), στις οποίες μπορούσε να υπάρχει αποθηκευμένο ένα παιγνίδι, οδήγησε στη δημιουργία των πρώτων εταιρειών ανάπτυξης παιγνιδιών. Πλέον, μία συσκευή μπορούσε να δεχτεί διαφορετικές κάρτες και ως εκ τούτου ο χρήστης είχε στη διάθεσή του μία ποικιλία από διαφορετικά παιγνίδια. Την περίοδο αυτή, η αγορά των βιντεοπαιγνιδιών είχε αρχίσει να κατακτά τις ΗΠΑ. Το 1977, η Atari κυκλοφόρησε την κονσόλα της με το όνομα

Video Computer System, που αργότερα μετονομάστηκε σε Atari 2600. Η Atari κυκλοφόρησε σχεδόν ταυτόχρονα και 9 διαφορετικά παιχνίδια, με αποτέλεσμα η κονσόλα της να γνωρίσει μεγάλες πωλήσεις. Άλλες σημαντικές κονσόλες ήταν η Intellivision της Mattel και η ColecoVision.

5.3 Η χρυσή περίοδος της δεκαετίας του 1980

Προς το τέλος της δεκαετίας του 1970 και αφού είχαν εξαπλωθεί οι κονσόλες παιχνιδιών, έκαναν την εμφάνισή τους οι πρώτες εταιρείες δημιουργίας και έκδοσης παιχνιδιών. Σε αντίθεση με τα άτομα που το είχαν ως ερασιτεχνική απασχόληση (χόμπι) ή τους κατασκευαστές κονσόλων, οι εταιρείες αυτές ασχολούνταν αποκλειστικά με τη συγγραφή και πώληση βιντεοπαιγνιδιών. Εξαιτίας μίας διαφωνίας με τη διοίκηση της Atari, μία ομάδα προγραμματιστών αποχώρησαν από την εταιρεία και ίδρυσαν την Activision, με στόχο την ανάπτυξη παιχνιδιών.

Την ίδια περίοδο δημιουργήθηκε και η Electronic Arts, η οποία ακόμη και σήμερα είναι μία από τις μεγαλύτερες εταιρείες στον χώρο των βιντεοπαιγνιδιών. Τη δημιουργία των εταιρειών αυτών ακολούθησε μία άνευ προηγουμένου κυκλοφορία τίτλων, οι οποίοι έγραψαν τη δική τους ιστορία. Η δεκαετία του 1980 ονομάστηκε «χρυσή εποχή των βιντεοπαιγνιδιών» και καθόρισε τα είδη (genres) στα οποία χωρίζουμε τα παιχνίδια. Ορισμένα παιχνίδια - σταθμοί ήταν το Zork (1980), ένα παιχνίδι περιπέτειας όπου ο παίκτης έδινε εντολές μέσω κειμένου στο παιχνίδι, ώστε να προχωρήσει η πλοκή. Η Roberta Williams δημιούργησε το Mystery House, την ίδια χρονιά, το πρώτο παιχνίδι περιπέτειας με γραφικά. Αργότερα, το 1984, η Roberta Williams με τον σύζυγό της θα ιδρύσουν τη Sierra και θα κυκλοφορήσουν την πετυχημένη σειρά παιχνιδιών περιπέτειας «King's Quest». Η LucasArts, η εταιρεία του George Lucas που σκηνοθέτησε τις ταινίες Star Wars (μεταξύ άλλων), κυκλοφόρησε το παιχνίδι περιπέτειας Maniac Mansion (1987), ένα από τα πρώτα παιχνίδια του είδους, που λειτουργούσε με τη μέθοδο 'point and click' (όλες οι επιλογές γίνονταν με τη βοήθεια ενός βέλους, που εμφανιζόταν στην οθόνη). Το παιχνίδι Kung-Fu Master (1984), καθιέρωσε τα παιχνίδια πολεμικών τεχνών, ενώ δύο χρόνια αργότερα το Renegade έφερε τον κόσμο των συμμοριών στις οθόνες μας.

Το Dragon's Lair (1983) ήταν το πρώτο παιχνίδι που κυκλοφόρησε σε Laserdisc (τον προκάτοχο, σε αρκετά σημεία, του CD / DVD). Τα γραφικά του ήταν εντυπωσιακά, όμως υστερούσε αρκετά σε χειρισμό. Το Pac-Man (1980) ήταν μία από τις μεγαλύτερες επιτυχίες όλων των εποχών και καθιέρωσε τα παιχνίδια με λαβυρίνθους, ενώ τη μεγάλη ανατροπή έφερε το Donkey Kong (1981), του Shigeru Miyamoto της Nintendo. Το Donkey Kong ήταν το πρώτο παιχνίδι, που επέτρεπε στον χαρακτήρα του να πηδήξει πάνω από αντικείμενα σε μία σειρά από πλατφόρμες, για αυτό και όλα τα παιχνίδια που ακολούθησαν τα βήματά του ονομάστηκαν παιχνίδια 'πλατφόρμας'. Το παιχνίδι αυτό παρουσίασε στον κόσμο για πρώτη φορά και ένα χοντρούλη υδραυλικό, τον Μάριο, που θα γίνει αργότερα σύμβολο των παιχνιδιών τύπου 'πλατφόρμας'. Με το τέλος της δεκαετίας του '80, κυκλοφόρησε ένα παιχνίδι πλατφόρμα / περιπέτειας, το 'Prince of Persia' (1989), που έγινε γνωστό για τα εντυπωσιακά κινηματογραφικά γραφικά και την κίνησή του.

5.4 Η εξέλιξη των οικιακών υπολογιστών (1980 - 1990)

Μέχρι τις αρχές της δεκαετίας του 1980, οι κυριότερες κυκλοφορίες, αλλά και οι μεγαλύτερες πωλήσεις παιχνιδιών, αφορούσαν στις κονσόλες. Από τις αρχές της δεκαετίας, αρκετοί υπολογιστές, όπως οι Commodore 64, Apple II, Sinclair ZX81 και Sinclair ZX Spectrum (ανάμεσα σε άλλους), προσπαθούσαν να κυριαρχήσουν τόσο στην αγορά οικιακών

υπολογιστών όσο και στην αγορά των παιγνιδιών. Το Commodore 64, που κυκλοφόρησε το 1982, χάρη στη χαμηλή τιμή του και στις δυνατότητές του στον ήχο και στα γραφικά, κατάφερε να κυριαρχήσει σε ένα μεγάλο μερίδιο της αγοράς και να τραβήξει την προσοχή των εταιρειών ανάπτυξης παιγνιδιών. Την ίδια περίοδο, στο Ηνωμένο Βασίλειο κυκλοφόρησε το Sinclair ZX Spectrum και γνώρισε σημαντική επιτυχία, κυρίως, εξαιτίας της πολύ χαμηλής του τιμής. Και τα δύο μηχανήματα μπορούσαν να συνδεθούν με μία κοινή τηλεόραση, γεγονός που περιόριζε ακόμη περισσότερο το κόστος τους, ενώ χρησιμοποιούσαν κοινές κασέτες ως αποθηκευτικό μέσο! Το 1984, η IBM κυκλοφόρησε τον δικό της υπολογιστή, ο οποίος πολύ σύντομα έγινε μεγάλη επιτυχία και οδήγησε στην αλλαγή του ονόματος των υπολογιστών από 'οικιακούς' σε 'προσωπικούς'. Αν και αρχικά τα πρώτα IBM ήταν πολύ ακριβά και δεν είχαν ιδιαίτερα καλές δυνατότητες στην αναπαραγωγή ήχου και γραφικών, η ανοικτή αρχιτεκτονική τους επέτρεψε την κυκλοφορία 'συμβατών' μηχανημάτων από άλλους κατασκευαστές, που οδήγησε σε σταδιακή πτώση των τιμών. Επίσης, η δυνατότητα επέκτασης των δυνατοτήτων των μηχανημάτων αυτών οδήγησε πολλές εταιρείες στην κατασκευή 'καρτών επέκτασης', που έδιναν εντυπωσιακές δυνατότητες στον τομέα, τόσο του ήχου όσο και της εικόνας.

5.5 Τα πρώτα Δικτυακά παιγνίδια

Πολύ πριν την εξάπλωση του Διαδικτύου, είχαν διαδοθεί δικτυακά παιγνίδια. Αυτά, συνήθως, προσφέρονταν ως υπηρεσία μέσω εταιρειών που παρείχαν δικτυακές υπηρεσίες τύπου Dialup bulletin. Οι υπηρεσίες αυτές ήταν πολύ διαδεδομένες στο εξωτερικό. Τα πρώτα τέτοια συστήματα ήταν διαθέσιμα από τα τέλη της δεκαετίας του '70, ενώ γνώρισαν μεγάλη επιτυχία στις αρχές και μέσα του '80. Έδιναν στον χρήστη ένα περιβάλλον επικοινωνίας που βασιζόταν σε κείμενο και πολύ γρήγορα εμφανίστηκαν τα παιγνίδια, που υποστήριζαν πολλούς παίκτες και μετέφεραν τους συνδρομητές σε φανταστικούς κόσμους γεμάτους περιπέτειες. Τα παιγνίδια αυτά ονομάστηκαν Multi user Dungeons και ήταν ο πρόδρομος των σημερινών Massively Multiuser Online Role Playing Games (MMORG), όπως το πολύ γνωστό World of Warcraft ή και το Eve Online. Ανάμεσα στους προδρόμους των δικτυακών παιγνιδιών πρέπει να αναφέρουμε το Maze War (1974), διαδικτυακό παιγνίδι, που λειτουργούσε σε ένα μεγάλο δίκτυο ερευνητικών υπολογιστών, καθώς και το Spasim, ένα τρισδιάστατο παιγνίδι προσομοίωσης διαστημοπλοίου, που έτρεχε κάτω από υπερυπολογιστές της εποχής του.

5.6 Η δεκαετία του '90

Στις αρχές της δεκαετίας του '90 άρχισε να γίνεται η εξάπλωση των πολυμέσων (multimedia), με την εμφάνιση σχετικά φθηνών συσκευών ανάγνωσης ψηφιακών δίσκων (CD-ROM drives), ενώ τέθηκαν και οι προδιαγραφές για τα multimedia PC (σύστημα Windows, κάρτα ήχου, ελάχιστη μνήμη). Οι ταχύτεροι επεξεργαστές, καθώς και οι κάρτες γραφικών, που υποστήριζαν τρισδιάστατες εικόνες, βοήθησαν τους δημιουργούς παιγνιδιών να αναπτύξουν πιο εντυπωσιακούς τίτλους, που οδήγησαν με τη σειρά τους περισσότερα άτομα στην αξιοποίηση των υπολογιστών ως μέσου ψυχαγωγίας.

Επίσης, η δεκαετία του '90 ήταν η πλέον σημαντική στην αγορά των παιγνιδιών. Μετά από μία ατυχή συνεργασία με τη Nintendo για δημιουργία CD-ROM για την κονσόλα της, η Sony αποφάσισε να κυκλοφορήσει δική της παιγνιδομηχανή. Έτσι, το PSX από μία εξωτερική συσκευή CD-ROM μετατράπηκε στην κονσόλα Playstation και σε πολύ μικρό χρονικό διάστημα κατάφερε να κατακτήσει την αγορά. Σε αυτό βοήθησαν και τίτλοι, όπως το Tomb

Raider με τη θρυλική Lara Croft, καθώς και το Final Fantasy VII, που απαιτούσαν μεγάλο αποθηκευτικό χώρο.

Τα τρισδιάστατα παιχνίδια άρχισαν να εξελίσσονται ταχύτατα με αποτέλεσμα οι εταιρείες ανάπτυξης παιχνιδιών να στραφούν σε αυτά, για όλες σχεδόν τις κυκλοφορίες τους. Παιχνίδια, όπως το Wolfenstein 3D, έκαναν την εμφάνισή τους και οδήγησαν αρκετούς παίκτες στο να αγοράσουν κάρτες γραφικών και ήχου, για να μπορούν να τα παίξουν στον υπολογιστή τους. Καθώς η ποιότητα (στον τομέα των γραφικών και ήχου) των παιχνιδιών βελτιωνόταν, οι ανάγκες σε χωρητικότητα αυξάνονταν με γεωμετρικό ρυθμό. Πολύ σύντομα οι δισκέτες σταμάτησαν να αποτελούν μέσο αποθήκευσης των παιχνιδιών, ενώ έγινε επιτακτική ανάγκη οι υπολογιστές να έχουν σκληρούς δίσκους. Οι ψηφιακοί δίσκοι (CD-ROM) έγιναν απαραίτητοι, ενώ τα πρώτα παιχνίδια που τους αξιοποιούσαν έκαναν την εμφάνισή τους. Χαρακτηριστικά είναι τα παιχνίδια 7th Guest (μυστήριο/περιπέτεια με στοιχεία ταινίας τρόμου) και Myst (περιπέτειας με εξαιρετικά ακόμη και με τα σημερινά δεδομένα γραφικά).

5.7 Διαδικτυακά παιχνίδια

Αυτή την περίοδο κάνουν την εμφάνισή τους τα πρώτα παιχνίδια, που υποστήριζαν πολλούς παίκτες μέσω Διαδικτύου (MUD ή Multi User Dungeons). Πλέον, όλα σχεδόν τα παιχνίδια δράσης πρώτου προσώπου, καθώς και αρκετά παιχνίδια στρατηγικής, αποκτούν δυνατότητα παιχνιδιού με άλλους παίκτες μέσω διαδικτύου. Με την ανάπτυξη των παιχνιδιών και την εξάπλωση του διαδικτύου, όλο και περισσότεροι παίκτες επιλέγουν ομαδικά παιχνίδια. Έτσι, τα MUD εξελίσσονται στα MMORG (Massively Multiplayer Online Role Playing Games), με τίτλους όπως το Ultima Online, Everquest κ.ά. Ανάμεσα στα παιχνίδια που κυριαρχούν σήμερα και ανήκουν στην κατηγορία αυτή είναι τα Final Fantasy XI, World of Warcraft, Eve Online κ.ά.

Παράλληλα με τα παιχνίδια που εγκαθιστά ο χρήστης στον υπολογιστή του, αναπτύχθηκαν και εκατοντάδες παιχνίδια, τα οποία έτρεχαν (και τρέχουν) αποκλειστικά μέσω του φυλλομετρητή ιστού. Τα παιχνίδια αυτά χρησιμοποιούν είτε την επέκταση (plugin) flash, είτε τον shockwave της Adobe (πρώην Macromedia). Αρκετά διαδικτυακά παιχνίδια αξιοποιούν και την Java με εντυπωσιακά πολλές φορές αποτελέσματα. Πέρα από τις σελίδες με τα διαδικτυακά παιχνίδια (π.χ. www.miniclip.com, www.orisinal.com), υπάρχουν και αρκετές σελίδες με παιχνίδια στρατηγικής ή και ρόλων πολλών παικτών (MMORG). Αρκετά από αυτά, όπως το Rune Scape, γνωρίζουν τεράστια επιτυχία και είναι δωρεάν (<http://www.runescape.com>).

5.8 Νέα γενιά κονσόλων (2004 – σήμερα)

Η Nintendo και η Sony κυκλοφορούν τις δικές τους φορητές κονσόλες – η Nintendo το DS, και η Sony το PSP. Τα δύο αυτά συστήματα είναι διαμετρικά αντίθετα μεταξύ τους. Το DS είναι μία φορητή κονσόλα με δύο οθόνες εκ των οποίων η μία είναι αφής. Ο χειρισμός της οθόνης αφής γίνεται με το χέρι ή (συνήθως) με τη γραφίδα. Πέρα από τον μοναδικό χειρισμό του DS, η Nintendo φρόντισε να εφοδιάσει την κονσόλα της με δυνατότητες ασύρματης δικτύωσης, όταν βρίσκονται και άλλα DS κοντά, για ασύρματο παιχνίδι πολλών παικτών. Το DS με τον μοναδικό χειρισμό του έδωσε την ευκαιρία για τη δημιουργία μίας σειράς παιχνιδιών που απευθύνονταν σε εντελώς διαφορετικό ακροατήριο από ό,τι μέχρι εκείνη την περίοδο. Παιχνίδια όπως το Nintendogs, Brain Age και άλλα παρόμοια κατάφεραν να τραβήξουν την προσοχή, ακόμη και ατόμων που δεν ασχολούνταν με παιχνίδια. Με την προσθήκη του Opera Browser (από την έκδοση DSi και μετά είναι ενσωματωμένο στην κονσόλα), οι χρήστες του Nintendo DS μπορούν να συνδεθούν στον παγκόσμιο ιστό μέσω

WiFi. Η κονσόλα DSi ενσωμάτωσε δύο κάμερες και δυνατότητα αποθήκευσης αρχείων από το Διαδίκτυο απευθείας στην κονσόλα.

Το PSP ακολούθησε εντελώς διαφορετική πορεία. Με γραφικές δυνατότητες που άγγιζαν αυτές του Playstation2, το PSP ξεπερνούσε στα γραφικά ακόμη και το GameCube. Η συσκευή αυτή προσφέρει ασύρματη σύνδεση στο Διαδίκτυο, αναπαραγωγή ταινιών από δίσκους UMD, καθώς και διασύνδεση με εξωτερικές οθόνες/τηλεοράσεις. Με το PSP Go, το οποίο κυκλοφορεί ταυτόχρονα με το PSP, η Sony αφαίρεσε τις δυνατότητες αναπαραγωγής περιεχομένου από δίσκους UMD και πλέον η συγκεκριμένη κονσόλα βασίζεται αποκλειστικά στα παιχνίδια, που ο χρήστης μπορεί να αγοράσει από το Διαδίκτυο. Είναι η πρώτη φορά που κονσόλα βασίζεται αποκλειστικά στο δίκτυο για την απόκτηση παιχνιδιών.

Το 2005 η Microsoft κυκλοφόρησε την επόμενη γενιά της κονσόλας της, το Xbox 360 με πολύ μεγαλύτερες δυνατότητες ως προς τα τεχνικά χαρακτηριστικά, τον αποθηκευτικό χώρο, αλλά και τις δυνατότητες σύνδεσης με το Διαδίκτυο. Η Microsoft φρόντισε να ενισχύσει την υπηρεσία Xbox Live και να προσφέρει παιχνίδια, ταινίες, πληροφοριακό υλικό, διασύνδεση με Facebook, MSN Messenger και άλλα εργαλεία, ώστε να κάνει την κονσόλα της κάτι περισσότερο από μία παιχνιδιομηχανή. Το 2013 η Microsoft κυκλοφόρησε το τελευταίο μοντέλο Xbox One με τα εξής χαρακτηριστικά:

- Μονάδα Blu-ray /DVD (μόνο για ανάγνωση): Ναι.
- CPU: 8 Core Microsoft custom CPU.
- GPU: 1,31 TFLOPS, κάρτα γραφικών 1853 MHz AMD Radeon GPU με 768 shaders.
- Είσοδος / Έξοδος: Super-Speed USBx (USB 3.0).
- Μονάδα σκληρού δίσκου: 1TB(μη αφαιρούμενο).
- Σύνδεση: Gigabit Ethernet, IEEE 802.11 a/b/g/n dual-band Wi-Fi (2.4 & 5Ghz),IR out, IR Blaster (Kinect).
- Μνήμη: 8GB DDR3.

Η Sony ακολούθησε το 2006 με την κυκλοφορία του Playstation3, μίας πανίσχυρης από πλευράς επεξεργαστικής ισχύος κονσόλα, που ενσωμάτωνε εντυπωσιακές τεχνολογίες, όπως συσκευή ανάγνωσης Blu-ray, διαδικτυακές υπηρεσίες κοινωνικής δικτύωσης (Playstation Home), φυλλομετρητή ιστού κ.ά. Το 2013 κυκλοφόρησε το Playstation 4 με τα εξής χαρακτηριστικά:

- Μονάδα Blu-ray /DVD (μόνο για ανάγνωση): BD x 6 CAV, DVD x 8 CAV.
- Κύριος επεξεργαστής: Προσαρμοσμένος επεξεργαστής μονού τσιπ.
- CPU: χαμηλής κατανάλωσης x86-64 AMD «Jaguar», 8 πυρήνες.
- GPU: 1,84 TFLOPS, κάρτα γραφικών AMD επόμενης γενιάς βασισμένη σε Radeon.
- Είσοδος / Έξοδος: Super-Speed USBx (USB 3.0) x 2, AUX x 1.
- Μονάδα σκληρού δίσκου: 500 GB.
- Σύνδεση: Ethernet (10BASE-T, 100BASE-TX, 1000BASE-T), IEEE 802.11 b/g/n, Bluetooth 2.1 (EDR).
- Μνήμη: GDDR5 8 GB.



Πηγή: Παιδαγωγικό Ινστιτούτο Κύπρου (<http://www.pi.ac.cy>)

6. Ο κύκλος ζωής ενός βιντεοπαιγνιδιού

6.1 Σύλληψη της ιδέας

Τα άτομα που ασχολούνται με την ανάπτυξη, επεξεργάζονται συνεχώς τις νέες ιδέες για παιχνίδια ως εξής:

- Σε μία μικρή ομάδα θα ανατεθεί η διερεύνηση μίας «καλής» ιδέας.
- Θα γίνει η πρώτη υλοποίηση της ιδέας και η αξιολόγηση του κατά πόσον είναι εφικτή.
- Θα παραχθεί ένα πρώτο έγγραφο σχεδίου του παιχνιδιού - πρόταση (game proposal).

6.2 Πρόταση παιχνιδιού

Η πρόταση παιχνιδιού δίνει μία εποπτική εικόνα του παιχνιδιού

- Περιγράφει την κεντρική ιδέα του παιχνιδιού.
- Το είδος στο οποίο θα ανήκει.
- Τι καινούργιο εισάγει στον χώρο και πώς θα σταθεί απέναντι στον ανταγωνισμό.
- Το κοινό στο οποίο απευθύνεται και την πλατφόρμα για την οποία θα αναπτυχθεί.
- Εκτιμήσεις για το κόστος ανάπτυξης και το ρίσκο του όλου έργου καταγράφονται.
- Περιέχει τα πρώτα καλλιτεχνικά σχέδια, που δίνουν το οπτικό στυλ και ύφος του παιχνιδιού.

Marketing

Το υλικό που παράγεται στο στάδιο αυτό θα χρησιμοποιηθεί για την διεκδίκηση και εξασφάλιση των απαραίτητων πόρων.

6.3 Προ-παραγωγή

Αν η πρόταση είναι επιτυχής και αποδεκτή, η ομάδα θα λάβει χρήματα για την επόμενη φάση, του preproduction.

- Αναπτύσσεται ένα συνολικό και λεπτομερές σχέδιο ανάπτυξης του παιγνιδιού, το οποίο θα αποτελέσει οδηγό και αναφορά για την μετέπειτα υλοποίησή του (game και technical design documents).
- Επιπλέον, αναπτύσσεται ένα πρωτότυπο (prototype) του παιγνιδιού.

Αυτή η φάση, συνήθως, χρησιμοποιεί υπάρχουσα τεχνολογία για την ανάπτυξη και αποτελεί σημαντικό στοιχείο του κύκλου ζωής ενός παιγνιδιού.

6.4 Παραγωγή του παιγνιδιού

Ένα παιγνίδι που εξασφαλίζει χρηματοδότηση περνά στη φάση της ανάπτυξης. Η συγκεκριμένη φάση είναι η πιο μακροσκελής και διαρκεί από 6 μήνες έως και 2 χρόνια.

- Η ομάδα ανάπτυξης μεγαλώνει δραματικά.
- Ορίζονται οι υποομάδες σχεδιασμού, περιεχομένου και προγραμματισμού.
- Δίνεται έμφαση στην πολύ καλή οργάνωση και διοίκηση.
- Υπάρχει αποδοτική παρακολούθηση ενεργειών των μελών (task-tracking) και αποδοτική αντιμετώπιση διενέξεων.

Για να μην ξεφύγει η ομάδα από τον συμφωνημένο χρόνο παράδοσης ο εκδότης επιβάλλει τη χρήση «παραδοτέων» (milestones).

- Συμφωνία μεταξύ ομάδας ανάπτυξης και εκδότη για ολοκλήρωση διάφορων χαρακτηριστικών του παιγνιδιού σε συγκεκριμένες χρονικές περιόδους.

Για παράδειγμα, στους 6 πρώτους μήνες θα είναι έτοιμα όλα τα επίπεδα (backgrounds), στους επόμενους 6 θα είναι έτοιμοι όλοι οι χαρακτήρες κ.λπ.

6.5 Φάση Alpha

Στο τέλος της φάσης παραγωγής του παιγνιδιού, η ομάδα είναι έτοιμη να περάσει στη φάση Alpha.

- Αρχή του τέλους της ανάπτυξης.
- Το παιγνίδι μπορεί να παιχτεί από αρχή μέχρι τέλους.
- Το περιεχόμενο μπορεί να μην είναι πλήρες.
- Αρχίζει το testing για σφάλματα (bug testing) από εξειδικευμένη ομάδα.
- Απορρίπτονται στοιχεία και λειτουργίες, που δεν μπορούν να υλοποιηθούν εμπρόθεσμα.
- Οι κατασκευαστές κονσόλας εξετάζουν το παιγνίδι και προτείνουν βελτιώσεις.

6.6 Φάση Beta

Στη φάση Beta το παιγνίδι είναι πλήρες.

- Ο κώδικας και το περιεχόμενο κλειδώνει, η ανάπτυξη σταματά.
- Όλη η προσπάθεια επικεντρώνεται στο να εξαλειφθούν τα σφάλματα από τον κώδικα.
- Έντονος κύκλος ανακάλυψης και διόρθωσης σφαλμάτων, που μπορεί να οδηγήσει σε μεγάλες υπερωρίες (crunch time).
- Τα σφάλματα (bugs) κατηγοριοποιούνται σε σχέση με την σημαντικότητά τους.

- Η ομάδα ασχολείται πρώτα με τα σφάλματα, που επηρεάζουν την εμπειρία του παιγνιδιού.
- Μερικά σφάλματα μπορεί να μην επιλυθούν ποτέ.

6.7 Πιστοποίηση και συσκευασία

Όταν ολοκληρωθεί η φάση Beta, το παιγνίδι πρέπει να πάρει πιστοποίηση καλής λειτουργίας.

- Η πιστοποίηση εκχωρείται από τον κατασκευαστή κονσόλας, αφότου περάσει το παιγνίδι από μία σειρά ελέγχων ποιότητας.
- Λάθη και παραλείψεις που τυχόν αποκαλύπτονται σε αυτό το βήμα απαιτούν ταχεία επέμβαση της ομάδας ανάπτυξης για την επίλυσή τους.
- Τέλος, ετοιμάζεται το πρωτότυπο (master disc) για μαζική αναπαραγωγή και συσκευασία (release to manufacture) και διάθεση στην αγορά.

6.8 Μετά την έκδοση (post release)

Η εμπλοκή της ομάδας ανάπτυξης στο παιγνίδι, συνήθως, δεν σταματά με τη διάθεση του στην αγορά.

- Συχνές είναι οι διορθώσεις σφαλμάτων στο παιγνίδι, ακόμα και μετά την έκδοσή του (patching) – όχι μόνο από αμέλεια.
- Σε multiplayer παιγνίδια η ομάδα ανάπτυξης θα πρέπει να παρέχει υποστήριξη καλής λειτουργίας στο παιγνίδι για αρκετό, ίσως, καιρό.
- Πρέπει να τροφοδοτούν το παιγνίδι με νέο περιεχόμενο, νέες αποστολές ακόμα και νέες λειτουργίες.

Με αυτές τις δραστηριότητες η ομάδα παραμένει παραγωγική μέχρι το ξεκίνημα ενός νέου παιγνιδιού.

6.9 Marketing - Προώθηση

Η προώθηση ενός παιγνιδιού περιλαμβάνει όλες εκείνες τις διεργασίες που αποσκοπούν πρώτον στο να αναγνωριστεί το κοινό ενός παιγνιδιού και δεύτερον να πειστεί το συγκεκριμένο κοινό να αγοράσει το παιγνίδι.

- Ξεκινά από τη φάση της σύλληψης της ιδέας, κατά την οποία πρέπει καταρχάς να καθοριστεί το προφίλ του δυνητικού κοινού του παιγνιδιού.
- Το μάρκετινγκ ενός παιγνιδιού γίνεται από τον εκδότη, που αναλαμβάνει να καλύψει και τα έξοδα.
- Μία ειδική ομάδα είναι υπεύθυνη για το μάρκετινγκ ενός παιγνιδιού σε κάθε εκδότη.
- Η ομάδα ανάπτυξης πρέπει να τροφοδοτεί την ομάδα μάρκετινγκ με πολύ υλικό.

Όλα αυτά απαιτούν χρόνο και από την ομάδα ανάπτυξης, με κίνδυνο να αποσπάται η προσοχή της σε κάτι άλλο εκτός από την ανάπτυξη του παιγνιδιού.

Να σημειωθεί ακόμη ότι οι διεργασίες της προώθησης και ο κύκλος ζωής ενός παιγνιδιού, είναι στοιχεία που πρέπει να είναι απόλυτα συγχρονισμένα.

Πηγή: Τμήμα Πληροφορικής – Ιόνιο Πανεπιστήμιο

**Ασκήσεις Κεφαλαίου****Άσκηση 1.1**

Να αναφέρετε τρία βασικά χαρακτηριστικά ενός ηλεκτρονικού παιχνιδιού.

Άσκηση 1.2

Να προσδιορίσετε σε ποιο ή σε ποια είδη (κατηγορίες) παιχνιδιών ανήκει το καθένα από τα πιο κάτω:

- (1) Ο παίκτης βρίσκεται σε ένα νησί πειρατών και προσπαθεί να βρει τον θησαυρό λύνοντας γρίφους και ακολουθώντας κομμάτια του χάρτη που βρίσκει, αποφεύγοντας τους πειρατές. Παίζεται από πολλούς παίκτες ανταγωνιστικά μέσω Διαδικτύου.
- (2) Παιγνίδι όπου ο κύριος χαρακτήρας είναι ένας ηλεκτρολόγος, ο οποίος πρέπει να περάσει από διάφορες πλατφόρμες, ασανσέρ και άλλα εμπόδια, για να φτάσει στον τελευταίο όροφο ενός ουρανοξύστη, αποφεύγοντας διάφορους κακοποιούς.
- (3) Παιγνίδι όπου ο κύριος χαρακτήρας κυκλοφορεί σε ένα στρατόπεδο και συλλέγει όπλα και άλλα αντικείμενα πυροβολώντας τους εχθρούς και τις εγκαταστάσεις για να κερδίσει βαθμούς.
- (4) Παιγνίδι όπου ο παίκτης μαθαίνει να πετά ένα επιβατικό αεροσκάφος συγκεκριμένου τύπου.
- (5) Διαδικτυακό παιγνίδι, με πολλούς παίκτες, όπου ο καθένας επιλέγει τον δικό του χαρακτήρα με ιδιαίτερες ικανότητες. Το παιγνίδι διαδραματίζεται σε έναν φανταστικό κόσμο, όπου οι διάφοροι χαρακτήρες συνομιλούν, ανταγωνίζονται, δημιουργούν συμμαχίες και προσπαθούν ο καθένας να ολοκληρώσει τη δική του αποστολή.
- (6) Διαδικτυακό παιγνίδι ποδοσφαίρου, όπου ο παίκτης ελέγχει τη συμπεριφορά των παιχτών μίας ομάδας, ενώ την αντίπαλη ομάδα ελέγχει είτε άλλος παίκτης, είτε ο υπολογιστής.

Άσκηση 1.3

Ποιός είναι ο σκοπός του Πανευρωπαϊκού Συστήματος Πληροφόρησης για τα Ηλεκτρονικά Παιχνίδια (PEGI);

- (1) Να βοηθά τους καταναλωτές στην επιλογή παιχνιδιών των οποίων οι νοητικές απαιτήσεις να ανταποκρίνονται σε συγκεκριμένη ηλικία.
- (2) Να βοηθά τους καταναλωτές στην επιλογή παιχνιδιών ώστε να προστατεύονται οι ανήλικοι από ακατάλληλο για την ηλικία τους περιεχόμενο.
- (3) Να εμποδίζει τους δημιουργούς παιχνιδιών από την προώθηση παιχνιδιών με βίαιο, σεξουαλικό ή προκλητικό περιεχόμενο.
- (4) Να βοηθά τους δημιουργούς παιχνιδιών στη δημιουργία παιχνιδιών κατάλληλων για τις νοητικές απαιτήσεις και δεξιότητες της κάθε ηλικίας.

Άσκηση 1.4

Να προσδιορίσετε την επισήμανση PEGI καθώς και τις περιγραφικές ενδείξεις (όπου υπάρχουν) που αντιστοιχούν στο καθένα από τα πιο κάτω παραδείγματα παιγνιδιών:

- (1) Παιγνίδι όπου ένας χαρούμενος σκαντζόχοιρος κυκλοφορεί σε ένα λαβύρινθο και τρώει σταφύλια για να κερδίσει βαθμούς, χάνει όμως μία ζωή εάν φάει φραγκόσυκο.
- (2) Τρισδιάστατο παιγνίδι όπου ο κύριος χαρακτήρας είναι ένας δραπέτης από τη φυλακή, ο οποίος περιφέρεται σε μία πόλη, καπνίζει, κλέβει αντικείμενα, ποντάρει σε καζίνο και με διάφορα όπλα σκοτώνει με βίαιο και παραστατικό τρόπο όσους τον απειλούν.
- (3) Παιγνίδι που απεικονίζει τον σκύλο Scooby ο οποίος προσπαθεί να λύσει γρίφους και να βγει από διάφορους τρομακτικούς χώρους (π.χ. το στοιχειωμένο κάστρο, το μαύρο δάσος, κ.λπ.), αποφεύγοντας φαντάσματα, μάγισσες και τρομακτικά τέρατα.
- (4) Διαδικτυακό παιγνίδι με τράπουλα, με έξι παίκτες, τρεις άντρες και τρεις γυναίκες γύρο από ένα τραπέζι, όπου ο κάθε παίκτης ποντάρει και κερδίζει ή χάνει χρήματα. Στο τέλος κάθε γύρου του παιγνιδιού ο χαρακτήρας του παίκτη που χάνει πρέπει να βγάλει κάποιο από τα ρούχα που φοράει. Ο παίκτης χάνει όταν δεν έχει πλέον άλλα χρήματα ή άλλα ρούχα.

Άσκηση 1.5

Έχετε δημιουργήσει ένα παιχνίδι με πρωταγωνιστή τον Μπομπ Σφουγγαράκη. Ο Μπομπ είναι ένα χαμογελαστό και άκακο σφουγγάρι που ζει στο βυθό της θάλασσας. Η μεγάλη αδυναμία του Μπομπ είναι οι καραμέλες. Σκοπός σας είναι να μαζέψετε όσες περισσότερες καραμέλες μπορείτε, μέσα σε ένα καθορισμένο χρονικό πλαίσιο. Στιγμιότυπο από το παιχνίδι φαίνεται πιο κάτω.



Σε ποια ηλικιακή κατηγορία της PEGI (Pan European Game Information) θα κατατάσσατε το πιο πάνω παιχνίδι και γιατί;





Άσκηση 1.6

- (1) Τι δηλώνει η πιο κάτω εικόνα στη συσκευασία ενός παιχνιδιού;



- (2) Να αναφέρετε τέσσερα (4) είδη παιχνιδιών με βάση τα κύρια χαρακτηριστικά τους.
- (3) Οι περιγραφικές ενδείξεις στο πίσω μέρος της συσκευασίας ενός παιχνιδιού αναφέρουν τους κύριους λόγους για τους οποίους ένα παιχνίδι έλαβε μια ορισμένη ηλικιακή διαβάθμιση. Υπάρχουν οχτώ τέτοιες περιγραφικές ενδείξεις: βία, χυδαία γλώσσα, φόβος, ναρκωτικά, σεξ, διακρίσεις, τζόγος και online παιχνίδι με άλλα άτομα.

Να συμπληρώσετε το παρακάτω πίνακα με την Περιγραφική Ένδειξη και με κατάλληλη περιγραφή για κάθε επισήμανση.

Επισήμανση		Περιγραφή
		
		
		
		

B7.2.2 Εισαγωγή στο GameMaker – Τα Δομικά Στοιχεία ενός Παιγνιδιού

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ◆ Να ενεργοποιούμε το περιβάλλον ανάπτυξης παιγνιδιού GameMaker
- ◆ Να ανοίγουμε ένα έτοιμο παιχνίδι, που έχει δημιουργηθεί με το GameMaker
- ◆ Να αναγνωρίζουμε τις οντότητες του GameMaker
- ◆ Να εκτελούμε ένα παιχνίδι
- ◆ Να σταματάμε την εκτέλεση ενός παιγνιδιού
- ◆ Να εισάγουμε ένα έτοιμο sprite
- ◆ Να δημιουργούμε ένα object
- ◆ Να δημιουργούμε ένα room
- ◆ Να ταξινομούμε τη σειρά εμφάνισης πίστας (room)
- ◆ Να εισάγουμε αντικείμενα ήχου
- ◆ Να αλλάζουμε τη μουσική και τους ήχους ενός έτοιμου παιγνιδιού
- ◆ Να αποθηκεύουμε ένα παιχνίδι.

1. Εισαγωγή

Το **Game Maker** είναι ένα περιβάλλον ανάπτυξης παιγνιδιών, δηλαδή ένα πρόγραμμα που μας βοηθά στη δημιουργία παιγνιδιών ηλεκτρονικού υπολογιστή. Το πρόγραμμα αυτό δημιουργήθηκε από τον Mark Overmars, καθηγητή στο ίδρυμα πληροφοριών και επιστημών πληροφορικής του πανεπιστημίου της Ουτρέχτης.

Ο σκοπός του **Game Maker** είναι να παράσχει ένα πρόγραμμα κατασκευής δισδιάστατων και τρισδιάστατων παιγνιδιών σε ένα συνηθισμένο προσωπικό υπολογιστή. Το πιο σημαντικό με το πρόγραμμα του Game Maker είναι το γεγονός ότι παρέχει ένα περιβάλλον που μπορεί εύκολα να χρησιμοποιηθεί από αρχάριους χρήστες και την ίδια στιγμή, παρέχει τεράστιες δυνατότητες για ευπαρουσίαστα και εξελιγμένα παιχνίδια.

2. Ενεργοποίηση του GameMaker, άνοιγμα έτοιμου παιγνιδιού και αποθήκευση παιγνιδιού

Για να ενεργοποιήσουμε το GameMaker και να ανοίξουμε ένα έτοιμο παιχνίδι που δημιουργήθηκε με το συγκεκριμένο περιβάλλον ακολουθούμε τα πιο κάτω βήματα. Το παιχνίδι που θα ανοίξουμε ονομάζεται *Treasure Island* και βρίσκεται στο αρχείο *Treasure.gmk*:


(1) Ανοίγουμε το GameMaker και πατούμε το κουμπί με την ένδειξη *Continue Using the Light Edition*.

(2) Από το μενού επιλέγουμε File→Open και ανοίγουμε το παιχνίδι *Treasure.gmk*.

Για να αποθηκεύσουμε το παιχνίδι με το ίδιο όνομα, από το μενού επιλέγουμε File→Save.

Για να αποθηκεύσουμε το παιχνίδι με άλλο όνομα (π.χ. *Treasure2.gmk*) , από το μενού επιλέγουμε File→Save As→ *Treasure2.gmk*.

3. Διαχείριση της εκτέλεσης ενός παιγνιδιού

Για να εκτελέσετε το παιχνίδι, να πατήσετε το πράσινο δεξί βέλος  (Run the game).

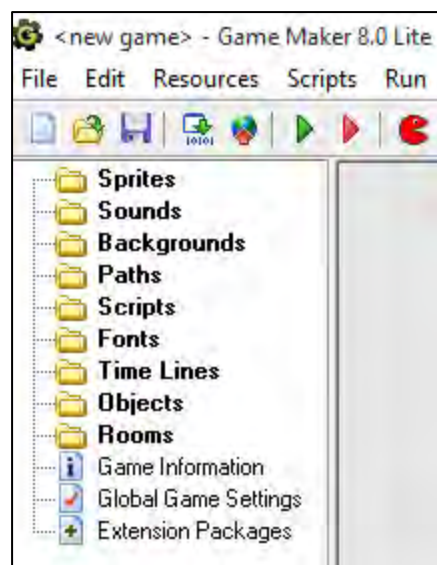


Για να σταματήσετε την εκτέλεση του παιχνιδιού, να πατήσετε το πλήκτρο **Esc**.

4. Τα βασικά δομικά στοιχεία (οντότητες) ενός παιχνιδιού στο GameMaker

Όταν ξεκινήσετε το GameMaker, θα δείτε έναν κατάλογο όλων των διαφορετικών τύπων οντοτήτων στην αριστερή πλευρά του παραθύρου. Οι οντότητες τοποθετούνται σε μία λίστα με μορφή δέντρου, παρόμοια με το παράθυρο εξερεύνησης αρχείων των Windows, που μπορούν να ανοίξουν και να κλείσουν. Αυτό το μέρος της οθόνης καλείται εξερευνητής των οντοτήτων. Οι διαφορετικές οντότητες είναι:

- A. Sprites,
- B. Sounds,
- C. Backgrounds,
- D. Paths,
- E. Scripts,
- F. Fonts,
- G. Time Lines,
- H. Objects,
- I. Rooms.




Σε περίπτωση που δεν εμφανίζονται όλες οι οντότητες, να κάνετε κλικ στο μενού File και επιλέξετε το Advanced Mode.

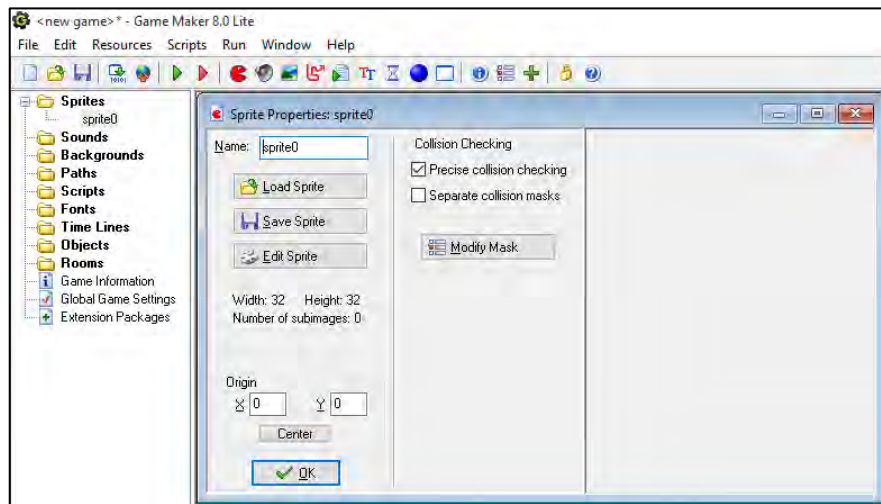
5. Sprites

Η οντότητα αυτή είναι βασικά μία εικόνα ή μία σειρά εικόνων. Καλείται sprite, επειδή, εκτός από τα δεδομένα για μία εικόνα, περιέχει επιπρόσθετες ιδιότητες για το πλάτος, το ύψος, τη διαφάνεια κ.ά. Κάθε Sprite απεικονίζει ψηφιακά κάποιο στοιχείο ή αντικείμενο στο παιχνίδι μας. Τα sprites μπορούν να προστεθούν στο παιχνίδι είτε με τη φόρτωση αρχείων εικόνας, που είναι ήδη αποθηκευμένες στο πρόγραμμα, είτε με τη δημιουργία των εικόνων. Θα δούμε παρακάτω βήμα προς βήμα την εισαγωγή ενός sprite.

Μπορούμε να προσθέσουμε μία νέα οντότητα sprite στο παιχνίδι με έναν από τους δύο πιο κάτω τρόπους:

- Κάνουμε κλικ στο κουμπί  **Create Sprite**.
- Κάνουμε δεξί κλικ **στο Sprite** στο μενού αριστερά και επιλέγουμε **Create Sprite**.

Το παράθυρο με τα χαρακτηριστικά του νέου sprite εμφανίζεται:



Name: Στο πεδίο αυτό εισάγεται το όνομα του sprite. Το όνομα θα αποτελείται από χαρακτήρες μικρούς ή κεφαλαίους (a, b,c,d,e ...A,B,C,D...κ.λπ.), από την κάτω παύλα (_) και από αριθμούς (1,2,3,4,...κ.λπ.).

Προσοχή! Το όνομα ενός sprite δεν ξεκινά με αριθμό ούτε με άλλα σύμβολα, εκτός από την κάτω παύλα.

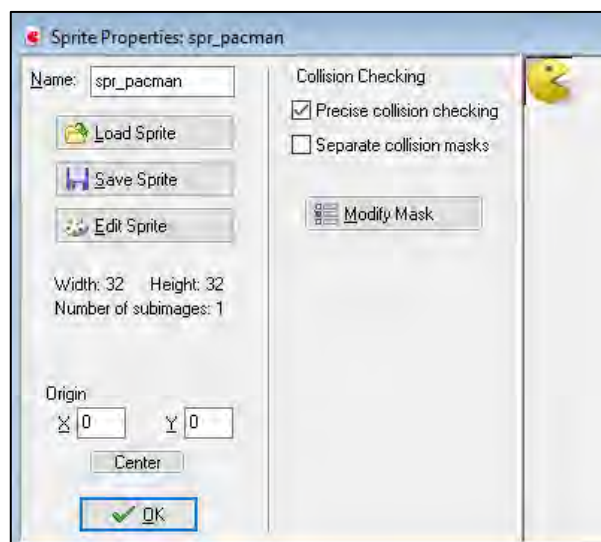
Συμβουλή! Να αρχίζετε την ονομασία των sprite συνήθως με **spr_**, προκειμένου να αντιλαμβάνεστε στο μέλλον καθώς χτίζετε το παιχνίδι σας ότι δεν πρόκειται για κάποια άλλη από τις οντότητες.

Εφόσον πλέον η οντότητά σας έχει κάποιο όνομα, θα πρέπει να της δώσετε μορφή μέσω μιας εικόνας. Αυτό γίνεται με δύο τρόπους:

- Με τη φόρτωση αρχείων εικόνας, που είναι ήδη αποθηκευμένες στο πρόγραμμα μέσω του κουμπιού **Load Sprite**.
- Δημιουργώντας τις δικές σας εικόνες, με τη βοήθεια του κουμπιού **Edit Sprite**.

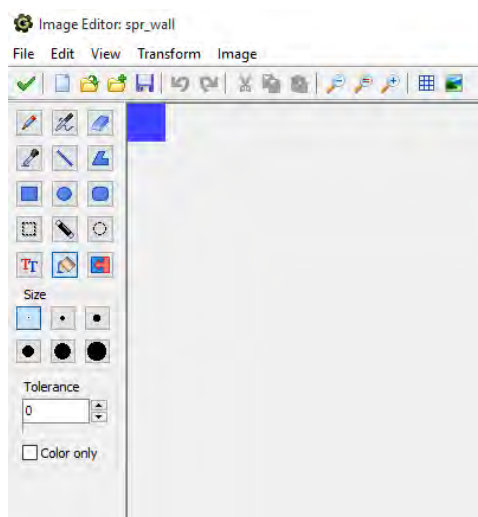
5.1 Φόρτωση έτοιμου Sprite από αρχείο (Load Sprite)

Για παράδειγμα, μπορούμε να φορτώσουμε την εικόνα του pacman.png, επιλέγοντας Load Sprite και στο πεδίο Name δίνουμε ένα κατάλληλο όνομα, π.χ. spr_pacman. Τέλος επιλέγουμε το OK.



5.2 Δημιουργία νέου Sprite (Edit Sprite)

Μπορούμε επίσης να δημιουργήσουμε ένα νέο sprite, μέσα από το GameMaker. Αυτό μπορεί να γίνει επιλέγοντας **Resources→Add Sprite** (ή δεξί κλικ στο sprite) → **Edit Sprite** → **File** → **New**. Στο νέο Sprite δίνουμε ένα κατάλληλο όνομα. Π.χ. εάν θέλουμε να έχουμε ένα Sprite που να αναπαριστά τον τοίχο, θα το ονομάσουμε spr_Wall. Κάνοντας διπλό κλικ στο image0 μπαίνουμε στη διαδικασία σχεδιασμού, όπου μπορούμε να σχεδιάσουμε την απεικόνιση του Sprite. Συνήθως η μέθοδος αυτή είναι κατάλληλη για απλά σχέδια, όπως γεωμετρικά σχήματα. Στο παράδειγμά μας θα σχεδιάσουμε ένα μπλε τετράγωνο.



5.3 Τροποποίηση Sprite (Edit Sprite)

Πολλές φορές χρειάζεται να τροποποιήσουμε ένα έτοιμο sprite, ή ένα sprite που δημιουργήσαμε. Αυτό μπορεί να γίνει μέσα από τη διαδικασία σχεδιασμού που είδαμε στην προηγούμενη παράγραφο. Ιδιαίτερης σημασίας είναι τα εργαλεία κάτω από την επιλογή **Transform**, τα οποία μας επιτρέπουν την αλλαγή του μεγέθους (Scale) και την περιστροφή (Rotate, Flip κ. ά), τα οποία συχνά χρειαζόμαστε σε ένα παιχνίδι. Για παράδειγμα να σμικρύνουμε μία εικόνα που βρήκαμε σε συγκεκριμένες διαστάσεις για να ταιριάζει με το παιχνίδι μας, ή για να δημιουργούμε διαφορετικές απεικονίσεις ενός αντικειμένου ανάλογα με την κατεύθυνση κίνησής του.

6. Εικόνες Φόντου (Backgrounds)

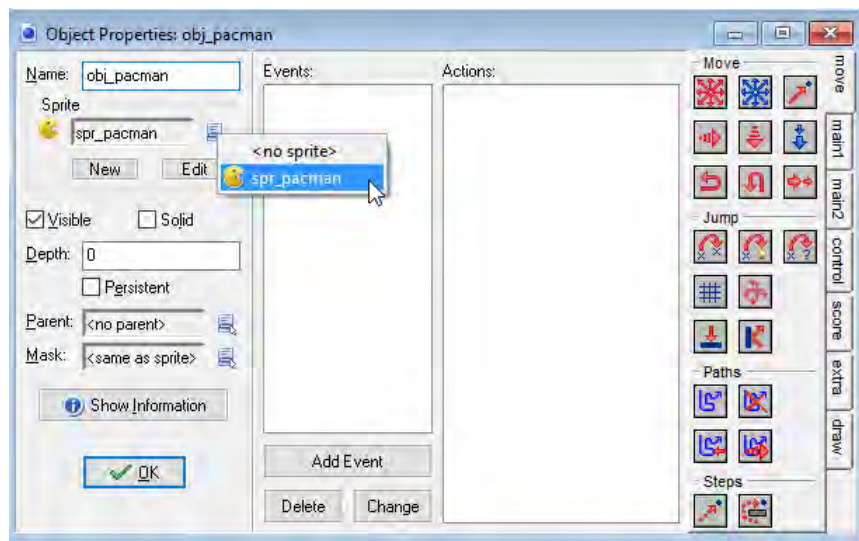
Πρόκειται για εικόνες που χρησιμοποιούνται ως φόντο στις διάφορες πίστες (rooms) του παιχνιδιού. Παρόμοια διαδικασία με τα sprites ισχύει και για τις εικόνες φόντου της κάθε πίστας για το παιχνίδι μας. Για να φορτώσουμε μία εικόνα φόντου από κάποιο αρχείο (π.χ. *mist.png*) κάνουμε δεξί κλικ στο **Backgrounds→Create Background→Load Background** και επιλέγουμε το αρχείο *mist.png*.

7. Objects

Τα **Objects** είναι η σημαντικότερη οντότητα στον κατασκευαστή παιχνιδιών. Είναι τα ενεργά αντικείμενα του παιχνιδιού, όπως ο κύριος χαρακτήρας, αντικείμενα, τμήματα τοίχου, κ.λπ. Είναι τα ενεργά αντικείμενα του παιχνιδιού. Ένα object αντιπροσωπεύεται από ένα (ή περισσότερα) sprite. Ο κατασκευαστής του παιχνιδιού προγραμματίζει τη συμπεριφορά του κάθε αντικειμένου (π.χ. την κίνησή του), καθώς επίσης και τη συμπεριφορά όταν το αντικείμενο συγκρούεται με άλλα αντικείμενα.

Για να εισάγετε ένα object στο παιχνίδι, να κάνετε δεξί Κλικ στο **Objects**→ **Create object**

Το παράθυρο με τα χαρακτηριστικά του νέου αντικειμένου (object) εμφανίζεται παρακάτω:



Name: Στο πεδίο αυτό εισάγεται το όνομα του **Object**. Να αρχίζετε την ονομασία των **Objects** συνήθως με obj, στη συγκεκριμένη περίπτωση **obj_pacman**.

Sprite: Στο πεδίο αυτό να επιλέξετε το αντίστοιχο sprite που κατασκευάσατε προηγουμένως.

Visible: Όταν είναι επιλεγμένο σημαίνει ότι το sprite θα είναι ορατό καθ' όλη τη διάρκεια του παιχνιδιού. Μπορεί να θέλουμε ένα αντικείμενο να είναι αρχικά αόρατο και να εμφανίζεται αργότερα, ή ακόμη να είναι πάντοτε αόρατο (π.χ. μία αόρατη πόρτα που όταν την ακουμπήσει κάποιο αντικείμενο να αλλάζει κάτι στη ροή του παιχνιδιού).

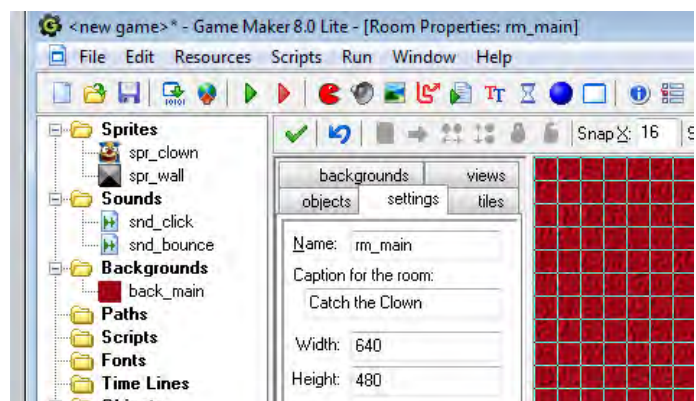
Solid: Όταν είναι επιλεγμένο, σημαίνει ότι το sprite είναι συμπαγές και, επομένως, αδιαπέραστο από οποιοδήποτε αντικείμενο που κινείται μέσα στο παιχνίδι.

8. Πίστα (Room)

Ένα παιχνίδι περιέχει μία ή περισσότερες πίστες (rooms), στις οποίες τοποθετούνται τα αντικείμενα σύμφωνα με την πλοκή του παιχνιδιού και μέσα από το παιχνίδι υπάρχει κάποιος μηχανισμός ώστε ο παίκτης να μετακινείται από τη μία στην άλλη.

Για να κατασκευάσετε μία πίστα, να κάνετε δεξί κλικ στο **Rooms** → **Create room**

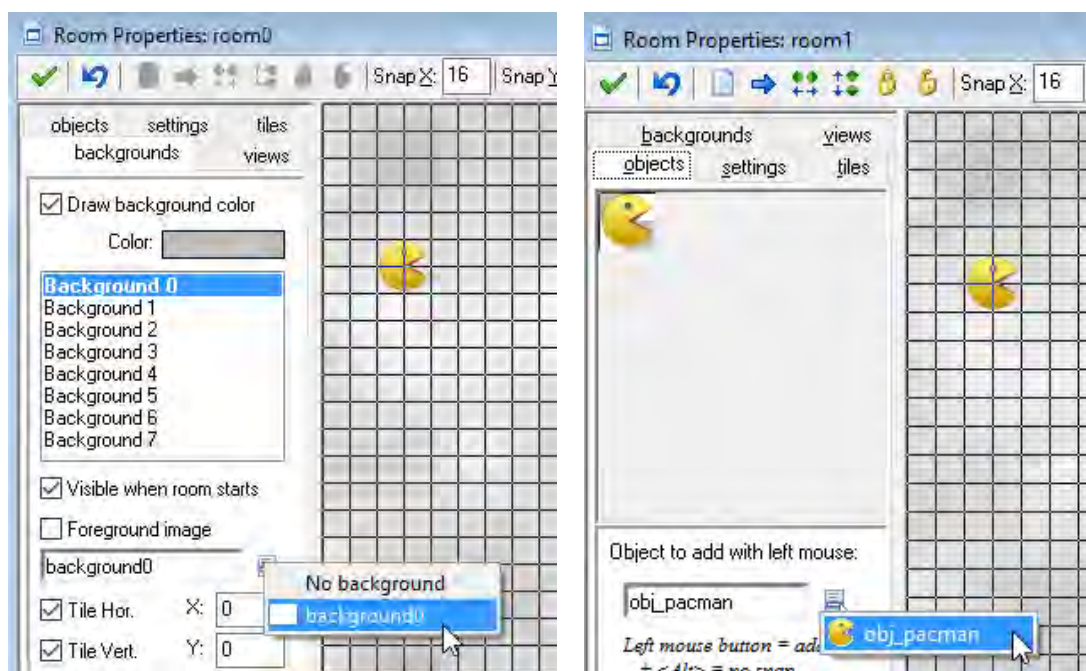
Το παράθυρο με τα χαρακτηριστικά της νέας πίστας έχει αρκετές ρυθμίσεις οργανωμένες σε καρτέλες. Η βασικότερη είναι η καρτέλα καρτέλα(tab) Settings, όπου μπορούμε να καθορίσουμε το όνομα της πίστας (όπως και με τα άλλα αντικείμενα, καλό είναι να αρχίζει με κάποιο πρόθεμα που να μας βοηθά στον εντοπισμό της, π.χ. **rm_**). Επίσης μπορείτε να καθορίσετε τη λεζάντα (caption) που θα εμφανίζεται στο παράθυρο με την πίστα, αλλά και να αλλάξετε τις διαστάσεις της πίστας, δηλαδή το πλάτος της (width) και το ύψος της (height).



Μπορείτε επίσης να αλλάξετε τις διαστάσεις του πλέγματος (Snap X και Snap Y) για να ταιριάζει με το μέγεθος των αντικειμένων.

Από την καρτέλα **Backgrounds** μπορείτε να επιλέξετε το φόντο, επιλέγοντας κάποιο από αυτά που ορίσατε προηγουμένως (στην οντότητα Backgrounds).

Τέλος από την καρτέλα **Objects** μπορείτε να επιλέξετε αντικείμενα (objects) και να τα τοποθετήσετε στην πίστα σε όποιο σημείο επιθυμείτε, απλά κάνοντας αριστερό click στο παράθυρο όπου φαίνεται η πίστα.



9. Sounds

Πρόκειται για διάφορους ήχους οι οποίοι χρησιμοποιούνται στο παιχνίδι μας, είτε ως μουσική κατά τη διάρκεια του παιχνιδιού, είτε ως κάποιος ήχος που θα ακούγεται όταν συμβεί κάτι (π.χ. ο ήχος που θα ακουστεί όταν ο παίκτης πατήσει σε κάποιο αντικείμενο, ή όταν τελειώνει το παιχνίδι).

10. Paths

Τα Paths είναι διαδρομές οι οποίες συνήθως περιγράφουν την πορεία κάποιου αντικειμένου (object). Δεν θα ασχοληθούμε ιδιαίτερα με αυτά.

11. Scripts

Πρόκειται για κομμάτια κώδικα γραμμένα στη γλώσσα προγραμματισμού του GameMaker. Θα τα μελετήσουμε προς το τέλος της ενότητας

12. Fonts

Πρόκειται για γραμματοσειρές. Δεν θα ασχοληθούμε ιδιαίτερα με αυτές.

13. Time Lines

Πρόκειται για προκαθορισμένες χρονικές διαδικασίες. Δεν θα ασχοληθούμε ιδιαίτερα με αυτές.



Ασκήσεις Κεφαλαίου

Άσκηση 2.1 Το παιχνίδι Treasure Island

Στο παιχνίδι **Treasure Island** ένας εξερευνητής συλλέγει αντικείμενα και κερδίζει βαθμούς, αλλά πρέπει να προσέχει διότι εάν ακουμπήσει κάποια στοιχεία του παιχνιδιού (π.χ. τον κόκκινο σκορπιό) χάνει μία ζωή. Το παιχνίδι αυτό δίνεται έτοιμο.

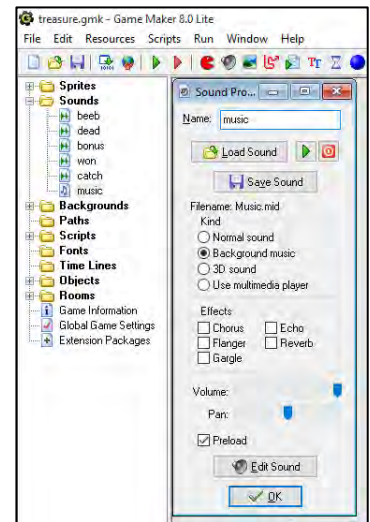
Σε αυτή την άσκηση θα το ανοίξουμε και θα το δοκιμάσουμε.

- (1) Να ανοίξετε το παιχνίδι επιλέγοντας από το μενού **File→Open** και επιλέγοντας το παιχνίδι *Treasure.gmk*.
- (2) Να εκτελέσετε το παιχνίδι πατώντας το κουμπί πράσινο δεξί βέλος.
- (3) Να διακόψετε την εκτέλεση του παιχνιδιού πατώντας το κουμπί **Esc**.

Άσκηση 2.2 Το παιχνίδι Treasure Island

Στο παιχνίδι **Treasure Island** της προηγούμενης άσκησης, να ανοίξετε διαδοχικά τις διάφορες κατηγορίες οντοτήτων και να τις μελετήσετε.

- (1) Πόσα sprites υπάρχουν;
- (2) Πόσα objects υπάρχουν; Για κάθε sprite υπάρχει κάποιο αντίστοιχο object;
- (3) Πόσα background υπάρχουν;
- (4) Ποιοι ήχοι (sounds) έχουν καθοριστεί και ποιο ρόλο έχει ο καθένας;
- (5) Πόσες πίστες (rooms) υπάρχουν;



Άσκηση 2.3 Το παιχνίδι Treasure Island

Στο παιχνίδι **Treasure Island** της προηγούμενης άσκησης, θα αλλάξουμε τη μουσική που ακούγεται όταν παίζει το παιχνίδι:

- (1) Να εμφανίσετε τους ήχους που ήδη χρησιμοποιούνται στο παιχνίδι, να επιλέξετε το **Sounds**.
- (2) Να αλλάξετε τη μουσική, κάνοντας διπλό κλικ στο **music**.
- (3) Να πατήσετε το κουμπί Load Sound, να επιλέξετε το Indiana Jones και να πατήσετε το **OK**.
- (4) Να αποθηκεύσετε το παιχνίδι επιλέγοντας **File→Save As→ Treasure2**
- (5) Να εκτελέσετε το παιχνίδι και να παρατηρήσετε την αλλαγή στη μουσική, καθώς και κατά πόσο στον τίτλο του παραθύρου εμφανίζονται οι ζωές που έχει ο εξερευνητής καθώς και οι βαθμοί που συνέλεξε (σκορ).

Άσκηση 2.4 Το παιχνίδι *Treasure Island*

Το παιχνίδι **Treasure Island**, αποτελείται από 4 πίστες (rooms).

- (1) Να αλλάξετε τη σειρά, κάνοντας drag and drop και να δοκιμάσετε το παιχνίδι.

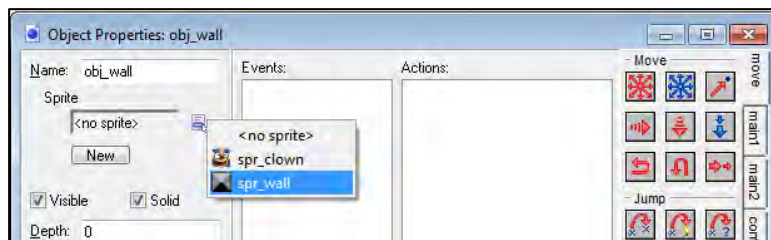
Προφανώς, οι τελευταίες πίστες έχουν μεγαλύτερο βαθμό δυσκολίας και λόγω αύξησης της ταχύτητας χρειάζονται καλύτερα αντανakλαστικά.

- (2) Να αποθηκεύσετε το παιχνίδι με όνομα treasure3.

**Άσκηση 2.5** Το παιχνίδι *Catch the clown*

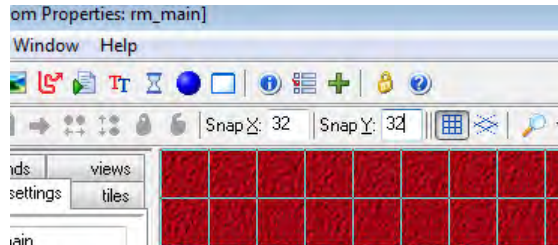
Στόχος σας είναι να δημιουργήσετε το παιχνίδι **Catch the clown**. Η ιδέα είναι απλή. Ένας κλόουν κινείται στην πίστα και ο παίκτης προσπαθεί να κάνει κλικ επάνω του με το ποντίκι. Αν τα καταφέρει, αυξάνεται το σκορ, ενώ ο κλόουν εμφανίζεται σε άλλο μέρος της οθόνης και γίνεται πιο γρήγορος. Σκοπός είναι ο παίκτης να πάρει όσο το δυνατόν ψηλότερο σκορ.

- (1) Να δημιουργήσετε το sprite για τον κλόουν. Να δώσετε στο sprite το όνομα spr_clown. Ως εικόνα για το sprite να επιλέξετε το αρχείο clown.png, που βρίσκεται στον φάκελο που θα σας υποδείξει ο καθηγητής σας.
- (2) Να δημιουργήσετε το sprite για τον τοίχο και να του δώστε όνομα spr_wall. Ως εικόνα για το sprite να επιλέξετε το αρχείο wall.png.
- (3) Να δημιουργήσετε ένα αντικείμενο ήχου. Να το ονομάσετε snd_click και να επιλέξετε το αρχείο ήχου click.wav. Αυτός ο ήχος θα ακούγεται, όταν ο παίκτης καταφέρει να κάνει κλικ πάνω στον κλόουν.
- (4) Να προσθέσετε ακόμα ένα αντικείμενο ήχου. Να το ονομάσετε snd_bounce και να επιλέξετε το αρχείο ήχου bounce.wav. Αυτός ο ήχος θα ακούγεται, όταν ο κλόουν συγκρούεται με κάποιον τοίχο.
- (5) Να εισαγάγετε ένα επιπλέον αντικείμενο ήχου και να το ονομάσετε snd_music, επιλέγοντας το αρχείο ήχου music.mid. Αυτός ο ήχος είναι η μουσική που θα παίζει σε όλη τη διάρκεια του παιχνιδιού.
- (6) Να δημιουργήσετε ένα νέο object με όνομα obj_wall. Να επιλέξετε το sprite spr_wall και να κάνετε κλικ στην επιλογή Solid.



- (7) Ακολούθως, να δημιουργήσετε το object για τον κλόουν με όνομα obj_clown. Για το αντικείμενο αυτό να επιλέξετε το sprite spr_clown. Δεν πρέπει να είναι επιλεγμένο το κουτί επιλογής solid. Θα χρειαστεί σε μετέπειτα στάδιο να προσθέσετε αρκετά events, με άλλα λόγια ενέργειες που θα θέλετε να εκτελεστούν.

- (8) Να δημιουργήσετε ένα νέο background με όνομα back_main και να φορτώσετε σε αυτό το αρχείο εικόνας background.png.
- (9) Να εισαγάγετε μία νέα πίστα (room) και να επιλέξετε ως background το back_main. Στην καρτέλα Settings να αλλάξετε το όνομα της πίστας σε rm_main και τη λεζάντα (caption) σε Catch the Clown. Να αλλάξετε το Snap X και Snap Y σε 32, ώστε το πλέγμα της πίστας σας να γίνει 32x32 για να ταιριάζει με τα αντικείμενα.



- (10) Να τοποθετήσετε αντικείμενα obj_wall γύρω από την πίστα και ένα αντικείμενο obj_clown στο κέντρο της πίστας. Η πίστα θα πρέπει να μοιάζει ως εξής:



- (11) Να δοκιμάσετε το παιχνίδι σας, πατώντας το πράσινο βέλος (run the game). Όπως βλέπετε, δεν γίνεται κάτι, γιατί δεν έχετε εισαγάγει τα κατάλληλα events-ενέργειες που θέλετε να εκτελεστούν.
- (12) Να αποθηκεύσετε το παιχνίδι στον φάκελό σας με όνομα clown.gmk.

+ Ασκήσεις Εμπλουτισμού

Άσκηση 2.6 Το παιχνίδι Treasure Island

Στο παιχνίδι **Treasure Island**:

- (1) Να αλλάξετε τα ηχητικά εφέ του παιχνιδιού, προσθέτοντας δικούς σας ήχους.
- (2) Να επιλέξετε άλλο background, που να έχει διαστάσεις 480x480.

B7.2.3 Συμβάντα (Events) και Ενέργειες (Actions)

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- Να δημιουργούμε event και action για ένα object
- Να προσθέτουμε χρονόμετρο (alarm)
- Να εκτελούμε ένα παιχνίδι
- Να παράγουμε εκτελέσιμο αρχείο.

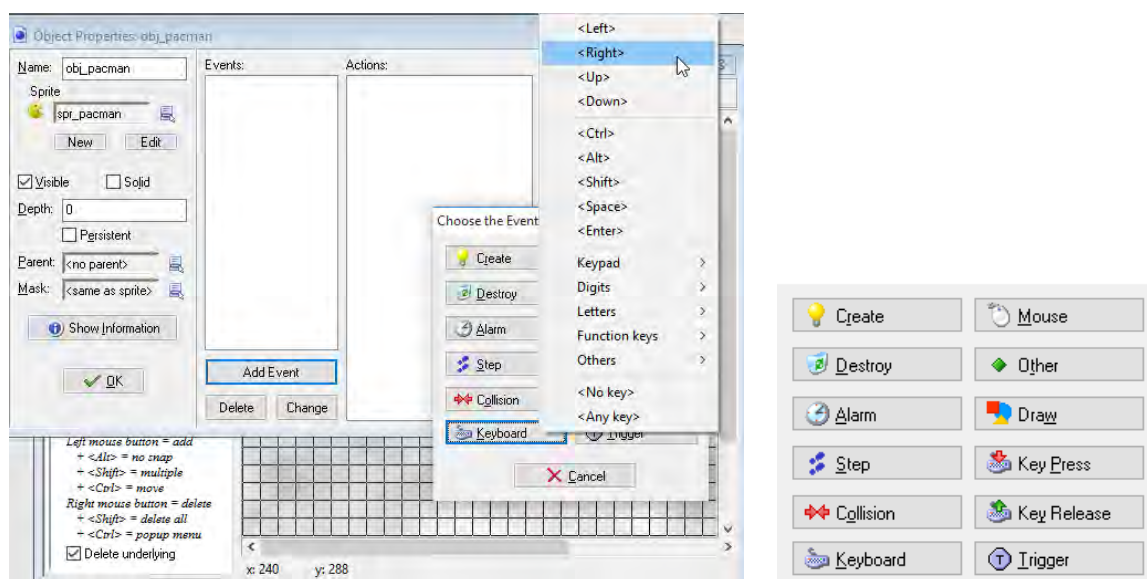
1. Εισαγωγή

Έχουμε δει πώς να δημιουργούμε πίστες (rooms) οι οποίες να περιέχουν αντικείμενα (objects), αλλά αυτό δεν είναι παρά μόνο η δημιουργία του σκηνικού για το παιχνίδι μας. Τα αντικείμενα είναι στατικά. Σε ένα ολοκληρωμένο παιχνίδι τα αντικείμενα είναι δυναμικά (κινούνται) ενώ υπάρχει και αλληλεπίδραση ή έλεγχος από τον χρήστη, ώστε το παιχνίδι να προκαλεί το ενδιαφέρον. Για να κάνουμε τα αντικείμενα να κινούνται και να αλληλοεπιδρούν μεταξύ τους, αλλά και με τον χρήστη, θα πρέπει να καθορίσουμε ενέργειες (actions). Οι ενέργειες αυτές δεν θα εκτελούνται τυχαία, αλλά ως αποτέλεσμα διάφορων συμβάντων (events) μέσα στο παιχνίδι. Για παράδειγμα, στο παιχνίδι Treasure Island, η κίνηση του εξερευνητή ελέγχεται από τον χρήστη μέσα από το πληκτρολόγιο και όταν συγκρουστεί με διάφορα αντικείμενα, είτε κερδίζει βαθμούς είτε χάνει.

2. Συμβάντα (Events)

Το περιβάλλον προγραμματισμού του GameMaker παρέχει για κάθε αντικείμενο μία σειρά από συμβάντα στα οποία το αντικείμενο μπορεί να προγραμματιστεί να ανταποκρίνεται με συγκεκριμένες σειρές από ενέργειες (actions).

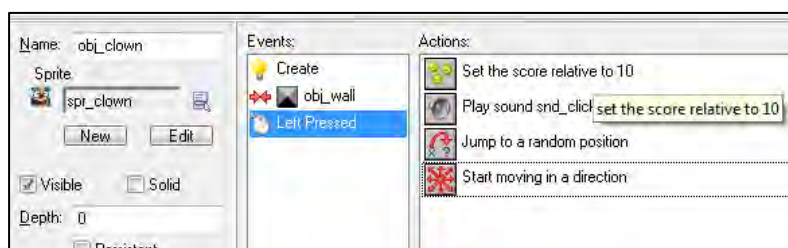
Για να δημιουργήσετε ένα νέο συμβάν, επιλέγεται το αντικείμενο και στη συνέχεια στο παράθυρο ιδιοτήτων (properties) πατάτε το κουμπί **Add Event**. Θα εμφανιστεί το παράθυρο με τα πιθανά συμβάντα για να επιλέξετε.



Δεν θα αναλύσουμε όλα τα συμβάντα, αλλά μόνον αυτά με τα οποία θα ασχοληθούμε στις δραστηριότητες της ενότητας αυτής που είναι τα ακόλουθα:

Συμβάν (Event)	Προκύπτει όταν ...
Create	Το αντικείμενο δημιουργηθεί.
Destroy	Το αντικείμενο θα καταστραφεί (διαγραφεί από το παιχνίδι).
Alarm	Όταν λήξει κάποιο χρονόμετρο που ξεκινήσαμε προηγουμένως.
Collision	Όταν το αντικείμενο συγκρουστεί με κάποιο άλλο αντικείμενο.
Keyboard	Όταν πατηθεί κάποιο (συγκεκριμένο) πλήκτρο στο πληκτρολόγιο.
Mouse	Όταν συμβεί κάτι με το ποντίκι (πάτημα αριστερού/δεξιού κουμπιού, κίνηση, κ.λπ.).
Other	Άλλα συμβάντα (π.χ. δεν έχουν απομείνει ζωές).

Όταν επιλέξουμε κάποιο συμβάν, εμφανίζεται στον κατάλογο με τα συμβάντα για το συγκεκριμένο αντικείμενο. Στη συνέχεια, επιλέγοντάς το μπορούμε να καθορίσουμε τον κατάλογο με τις ενέργειες (actions) που θα εκτελούνται κάθε φορά που προκύπτει το συμβάν.



3. Ενέργειες (Actions)

Οι ενέργειες (actions) που μπορούν να πραγματοποιηθούν για κάθε συμβάν (event) είναι αυτές που καθορίζονται από τους κανόνες του παιχνιδιού. Οι ενέργειες στο GameMaker είναι ομαδοποιημένες σε καρτέλες για να μπορεί εύκολα να τις επιλέγει ο δημιουργός του παιχνιδιού και να τις τοποθετεί στη σειρά.

Μία κατηγορία ενεργειών καθορίζει την κίνηση (Move) του αντικειμένου (π.χ. εάν κινείται με κάποια ταχύτητα προς μία κατεύθυνση, ή εάν μετακινείται σε κάθε βήμα του παιχνιδιού για κάποια απόσταση) εάν αναπηδά (π.χ. για συγκρούσεις με αντικείμενα κ.λπ.).

Άλλη κατηγορία έχει να κάνει με τη διαχείριση βαθμών και ζωών.

Επίσης υπάρχουν κατηγορίες ενεργειών που έχουν να κάνουν με τη δημιουργία και την διαγραφή αντικειμένων (π.χ. στο Treasure Island, όταν περάσει ο εξερευνητής πάνω από ένα αντικείμενο, αυτό διαγράφεται) και την αλλαγή ενός αντικειμένου (π.χ. την αλλαγή του sprite όταν το αντικείμενο αλλάζει κατεύθυνση κίνησης). Αυτά δεν θα παρουσιαστούν εδώ, διότι είναι πάρα πολλά και δεν θα βοηθούσε μία πολυσέλιδη περιγραφή. Παρουσιάζονται όμως στις ασκήσεις, εκεί που χρησιμοποιούνται, ώστε να φαίνεται μέσα από συγκεκριμένο παράδειγμα και ο σκοπός τους.



Ασκήσεις Κεφαλαίου

Άσκηση 3.1 Το παιχνίδι Catch the Clown

Θα ανοίξουμε το παιχνίδι **Catch the Clown** που ήδη δημιουργήσαμε και θα προσθέσουμε συμβάντα, ώστε να έχουμε ένα λειτουργικό παιχνίδι. Αρχικά, με το που αρχίζει το παιχνίδι θα κάνουμε τον κλόουν να κινείται προς τυχαία κατεύθυνση στην πίστα και όταν συγκρουστεί με κάποιον τοίχο να αναπηδά.

- (1) Να ανοίξετε το παιχνίδι clown.gmk.
- (2) Να επιλέξετε το αντικείμενο του κλόουν (obj_clown), να εισαγάγετε ένα νέο event και να επιλέξετε create (όταν δημιουργηθεί ο κλόουν).
- (3) Στις ενέργειες (actions) να διαλέξετε **move fixed**, να επιλέξετε όλες τις κατευθύνσεις (εκτός από την μεσαία, η οποία σταματά τον κλόουν) και στην ταχύτητα πληκτρολογήστε 4 (αυτό σημαίνει κίνηση προς τυχαία κατεύθυνση).

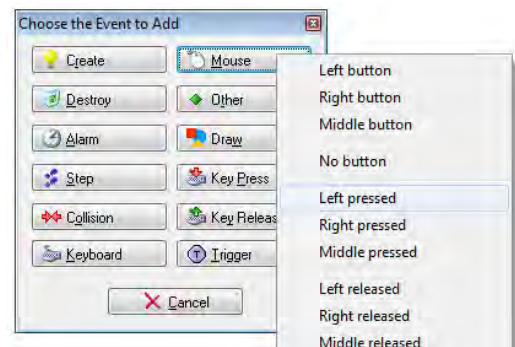


- (4) Στο ίδιο event (create) να προσθέσετε τον ήχο snd_music. Στο loop να επιλέξετε true, ώστε όταν τελειώσει η μουσική να ξαναρχίζει από την αρχή.
- (5) Να προσθέσετε ένα δεύτερο event, ώστε όταν συγκρούεται ο κλόουν με κάποιον τοίχο (collision) να αναπηδά (bounce). Επίσης, με τη σύγκρουση να ακούγεται ο ήχος snd_bounce.
- (6) Να ελέγξετε το παιχνίδι σας, πατώντας το πράσινο βέλος (run the game).



Άσκηση 3.2 Το παιχνίδι Catch the Clown

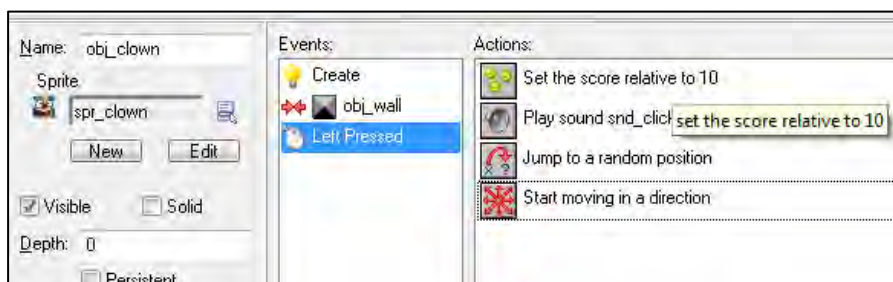
Στο παιχνίδι **Catch the Clown** θα προσθέσουμε ένα νέο συμβάν, ώστε όταν ο χρήστης πατήσει τον κλόουν με το ποντίκι, να αυξάνεται το σκορ και ο κλόουν να εμφανίζεται σε νέα θέση και να κινείται με κάπως μεγαλύτερη ταχύτητα από ότι προηγουμένως.

- (1) Να επιλέξετε το αντικείμενο του κλόουν (obj_clown), να εισαγάγετε ένα νέο event και να επιλέξετε **mouse** και στη συνέχεια **Left Pressed** create (όταν πατηθεί το αριστερό κουμπί του



ποντικιού πάνω στον κλόουν).

- (2) Στις ενέργειες, από την καρτέλα score να σύρετε το εικονίδιο **Set score** . Στο new score να γράψετε 10 και να επιλέξετε το κουτί **relative**, ώστε να προστίθενται 10 βαθμοί στο υπάρχον σκορ (όπως αν ορίζατε μεταβλητή $score=score+10$) και όχι να γίνεται 10 με κάθε πάτημα του κουμπιού του ποντικιού.
- (3) Στις ενέργειες για το ίδιο event, από την καρτέλα **Main1** να επιλέξετε να ακούγεται ο ήχος snd_click.
- (4) Συνεχίζοντας, από την καρτέλα **Move** να επιλέξετε **Jump to random**  και να πατήσετε OK χωρίς να αλλάξετε τις τιμές. Έτσι μετά το κάθε πάτημα με το αριστερό κουμπί, ο κλόουν θα μετακινείται σε τυχαία θέση.
- (5) Η τελευταία ενέργεια (action) για αυτό το συμβάν (event) είναι να βάλετε κίνηση **move fixed**. Να επιλέξετε, όπως και στη δημιουργία του κλόουν, όλες τις κατευθύνσεις εκτός από τη μεσαία, και ως ταχύτητα (speed) να πληκτρολογήσετε 0.5, επιλέγοντας όμως το κουτί relative. Έτσι με κάθε πάτημα η ταχύτητα του κλόουν θα αυξάνεται κατά 0.5, επιταχύνοντας σταδιακά τον κλόουν, ώστε να αυξάνεται ο βαθμός δυσκολίας για τον χρήστη. Η σειρά των ενεργειών θα πρέπει να είναι όπως φαίνεται στο πιο κάτω σχήμα.

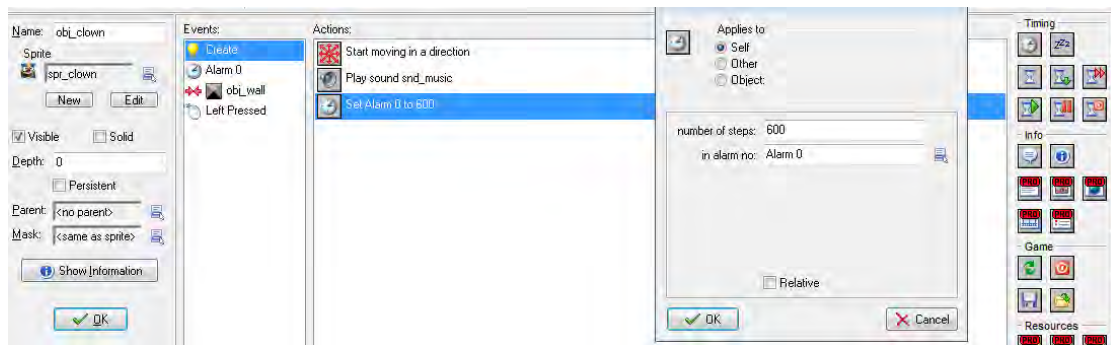


- (6) Να δοκιμάσετε το παιχνίδι σας, το οποίο έχει σχεδόν ολοκληρωθεί.

Άσκηση 3.3 Το παιχνίδι Catch the Clown

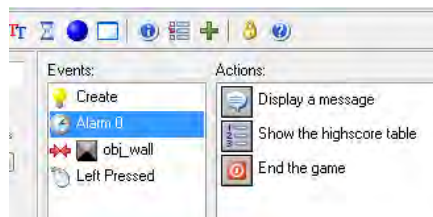
Απομένει να καθορίσουμε έναν τρόπο ώστε το παιχνίδι να τελειώνει, διαφορετικά το παιχνίδι θα συνεχίζει για πάντα (μέχρι να πατήσετε το πλήκτρο **Esc**). Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε ένα χρονόμετρο, το οποίο θα ενεργοποιείται με την έναρξη του παιχνιδιού και όταν λήξει ο χρόνος να τερματίζεται το παιχνίδι.

- (1) Να διπλαπατήσετε στο αντικείμενο **obj_clown**.
- (2) Να επιλέξετε το event **Create** και από την καρτέλα **Main2** να σύρετε το εικονίδιο **Set alarm** κάτω από το action **Play sound** snd_music, που βάλατε προηγουμένως.
- (3) Να καθορίσετε τον αριθμό των steps που χρειάζονται. Κάθε δευτερόλεπτο αντιστοιχεί σε 30 steps. Να εισαγάγετε 600 steps, τα οποία αντιστοιχούν σε 20 δευτερόλεπτα.
- (4) Να αφήσετε επιλεγμένο το Alarm 0 και να πατήσετε OK.



Με τα πιο πάνω βήματα ενεργοποιείται το χρονόμετρο Alarm 0 τη στιγμή που δημιουργείται το αντικείμενο obj_clown, δηλαδή όταν αρχίζει το παιχνίδι. Απομένει να καθορίσετε τι θα γίνει όταν ο χρόνος του χρονομέτρου περάσει.

- (5) Στο αντικείμενο obj_clown να επιλέξετε **Add Event** και στη συνέχεια **Alarm**. Να διαλέξετε το Alarm 0 και να δώσετε τα actions που θα εκτελεστούν, όταν ο χρόνος του Alarm 0 περάσει.
- (6) Να προσθέσετε το action **Display a Message** και να πληκτρολογήσετε το μήνυμα Game Over.
- (7) Να προσθέσετε το κατάλληλο action για την εμφάνιση του **High Score**.
- (8) Να προσθέσετε το κατάλληλο action για το τέλος του παιχνιδιού.
- (9) Τα actions του χρονομέτρου Alarm 0 φαίνονται στο πιο κάτω σχήμα:



- (10) Να εκτελέσετε το παιχνίδι.
- (11) Να αποθηκεύσετε το παιχνίδι.

Άσκηση 3.4 Το παιχνίδι Catch the clown

Το παιχνίδι είναι έτοιμο. Απομένει να το μετατρέψουμε σε εκτελέσιμο αρχείο, ώστε να μπορεί να δοθεί στους χρήστες και να εκτελείται χωρίς να χρειάζεται το περιβάλλον του GameMaker.

- (1) Από το μενού να επιλέξετε **File→Create executable**. Θα παρατηρήσετε ότι εμφανίζεται ένα νέο αρχείο με το ίδιο όνομα όπως και το αρχείο του παιχνιδιού (clown.gmk), αλλά με διαφορετική επέκταση (clown.exe).
- (2) Να κλείσετε το GameMaker και να διπλοπατήσετε το αρχείο clown.exe ώστε να βεβαιωθείτε ότι αποτελεί το παιχνίδι σας και ότι λειτουργεί.

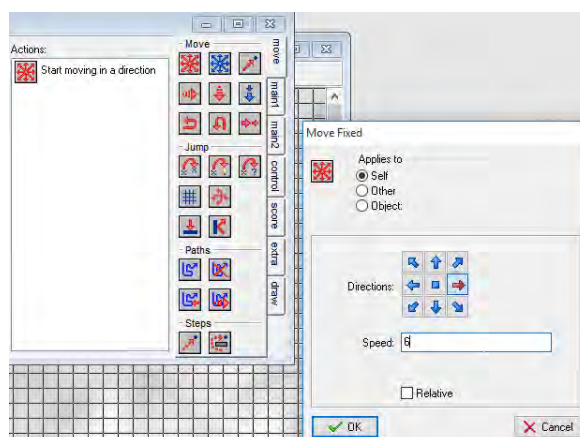
Άσκηση 3.5 Το παιχνίδι Pacman

Το **Pacman** είναι ένα από τα πιο πετυχημένα παιχνίδια της δεκαετίας του 1980. Ένας χαρακτήρας ο οποίος ελέγχεται από τον χρήστη κινείται σε έναν λαβύρινθο και τρώει διάφορα αντικείμενα (κυρίως κουκίδες) κερδίζοντας βαθμούς. Θα πρέπει να αποφύγει τα φαντάσματα που τον κυνηγούν, και όταν τον πιάσουν χάνει μία ζωή. Στο παιχνίδι αυτό, όταν

δεν έχουν απομείνει αντικείμενα για συλλογή, η πίστα τερματίζεται και αρχίζει η επόμενη, με διαφορετικό λαβύρινθο και γρηγορότερες ταχύτητες κίνησης.

Στις δραστηριότητες που ακολουθούν θα δημιουργήσουμε σταδιακά το παιχνίδι αυτό.

- (1) Να ενεργοποιήσετε το GameMaker για να δημιουργήσετε ένα νέο παιχνίδι.
- (2) Να δημιουργήσετε το sprite για τον Pacman. Να δώσετε στο sprite το όνομα **spr_pacman**. Ως εικόνα για το sprite να επιλέξετε το αρχείο **pacman.png**, που βρίσκεται στον φάκελο που θα σας υποδείξει ο καθηγητής σας.
- (3) Να δημιουργήσετε ένα νέο background και να φορτώσετε σε αυτό το αρχείο εικόνας **mist.png**.
- (4) Ακολούθως, να δημιουργήσετε το object για τον Pacman με όνομα **obj_pacman**. Για το αντικείμενο αυτό να επιλέξετε το sprite **spr_pacman**. Δεν πρέπει να είναι επιλεγμένο το κουτί επιλογής **solid**.
- (5) Να εισαγάγετε μία νέα πίστα (room) και να επιλέξετε ως background αυτό που δημιουργήσατε προηγουμένως στο (γ). Επίσης από την καρτέλα **Objects** να επιλέξετε το **obj_pacman** και να το εισαγάγετε στην πίστα σε όποιο σημείο επιθυμείτε.
- (6) Να δοκιμάσετε το παιχνίδι πατώντας το κουμπί εκτέλεσης από τη γραμμή εργαλείων. Θα παρατηρήσετε ότι ο χαρακτήρας δεν κινείται. Για να σταματήσετε την εκτέλεση του παιχνιδιού πατάτε το πλήκτρο **Esc**.
- (7) Για να βάλετε κίνηση στον χαρακτήρα και να τον ελέγχετε με τα βέλη του πληκτρολογίου, θα πρέπει να καθορίσετε κατάλληλα συμβάντα (events) και ενέργειες (actions). Να επιλέξετε το **Objects** και να κάνετε διπλό κλικ στο **obj_pacman**. Έπειτα, να επιλέξετε **Keyboard, Right**, ώστε να καθορίσετε τι θα κάνει ο χαρακτήρας σας όταν πατάτε το δεξί βέλος του πληκτρολογίου.
- (8) Να κάνετε drag and drop το πρώτο εικονίδιο του **Move ()** στο πεδίο **Actions** και να επιλέξετε το δεξί βέλος και να καθορίσετε την ταχύτητα - speed σε 6.



- (9) Να δοκιμάσετε τώρα το παιχνίδι σας, πατώντας το πράσινο βέλος (run the game). Θα παρατηρήσετε ότι μόλις πατήσετε το δεξί βέλος ο χαρακτήρας σας θα κινείται δεξιά χωρίς να σταματάει.

- (10) Να προγραμματίσετε τον χαρακτήρα σας, ώστε να κινείται προς όλες τις κατευθύνσεις <left>, <up>, <down> και να σταματά όταν δεν πατάτε κανένα πλήκτρο <no key>.
- (11) Να δοκιμάσετε το παιχνίδι σας για να βεβαιωθείτε ότι κινείται σύμφωνα με τα πλήκτρα που πατάτε και να το αποθηκεύσετε, με όνομα **pacman1.gmk**.

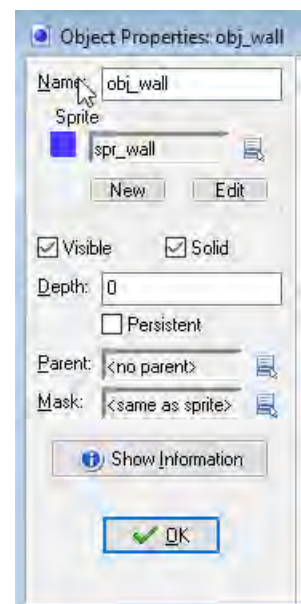
Άσκηση 3.6 Το παιχνίδι Pacman

Το παιχνίδι **Pacman** όπως το δημιουργήσαμε μέχρι το τέλος της προηγούμενης άσκησης χρειάζεται αρκετά στοιχεία ακόμη για να ολοκληρωθεί. Στην άσκηση αυτή θα επικεντρωθούμε στα εξής:

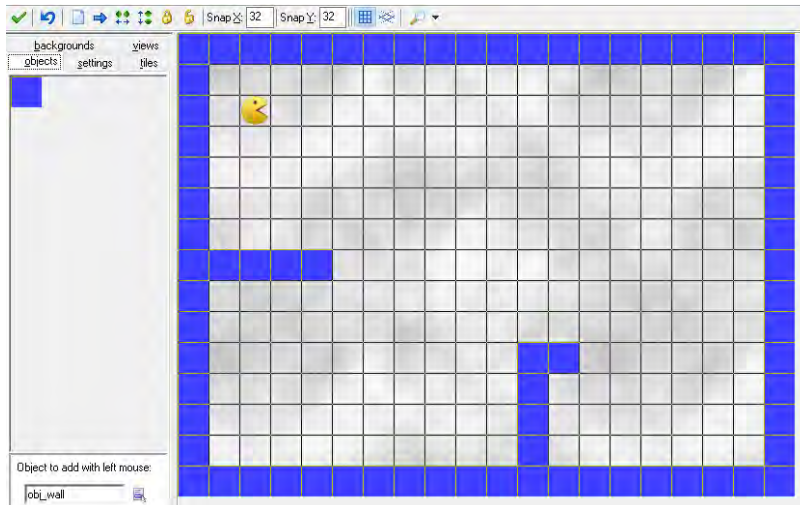
1. Δημιουργία sprite και αντικειμένου ενός τοίχου για τοποθέτηση περιμετρικά ή και εσωτερικά στην πίστα.
2. Ορισμός της συμπεριφοράς του Pacman, ώστε όταν έρχεται σε επαφή με τον τοίχο να σταματάει.
3. Δημιουργία sprite και αντικειμένου μίας κουκίδας για τοποθέτηση εντός της πίστας.
4. Ορισμός της συμπεριφοράς του Pacman, ώστε όταν έρχεται σε επαφή με μία κουκίδα:
 - ο Η κουκίδα να εξαφανίζεται.
 - ο Να ακούγεται κάποιος ήχος.
 - ο Να αυξάνεται το σκορ κατά 1 βαθμό.
5. Αλλαγή εμφάνισης του χαρακτήρα Pacman, ανάλογα με την κατεύθυνση που κινείται.

Υλοποίηση

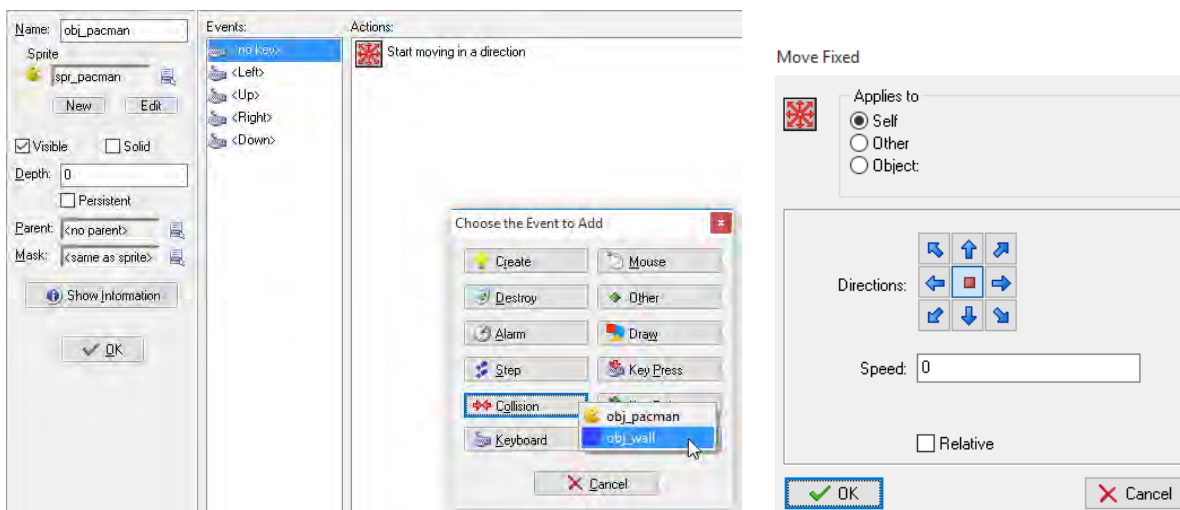
- (1) Να δημιουργήσετε ένα νέο sprite, διαστάσεων 32x32. Αυτό μπορεί να γίνει επιλέγοντας Resources→Add Sprite (ή δεξί κλικ στο sprite) → Edit Sprite → File → New. Να το ονομάσετε spr_wall και να κάνετε διπλό κλικ στο image0 για να ανοίξει το σχεδιαστικό πρόγραμμα. Σχεδιάστε ένα μπλε τετράγωνο.
- (2) Να δημιουργήσετε ένα νέο object με όνομα obj_wall. Έχει σημασία το κουτί Solid, να είναι επιλεγμένο, διότι δεν θέλετε πάνω από το αντικείμενο του τοίχου να μπορεί να περνάει κάποιο άλλο αντικείμενο.
- (3) Έπειτα, θα πρέπει να το τοποθετήσετε στην πίστα. Να μεταβείτε στο **room**. Να αλλάξετε το **SnapX** και **SnapY** σε **32**, ώστε το κάθε τετραγωνάκι του πλέγματος της πίστας να είναι 32x32, ίδιας δηλαδή διάστασης με το αντικείμενο του obj_wall, έτσι ώστε να γίνεται ευκολότερη η τοποθέτησή του στην πίστα. Να επιλέξετε το αντικείμενο obj_wall από την καρτέλα objects και να το τοποθετήσετε περιμετρικά της πίστας ώστε να περιορίσετε το χώρο στον χαρακτήρα και όπου αλλού θέλετε στην



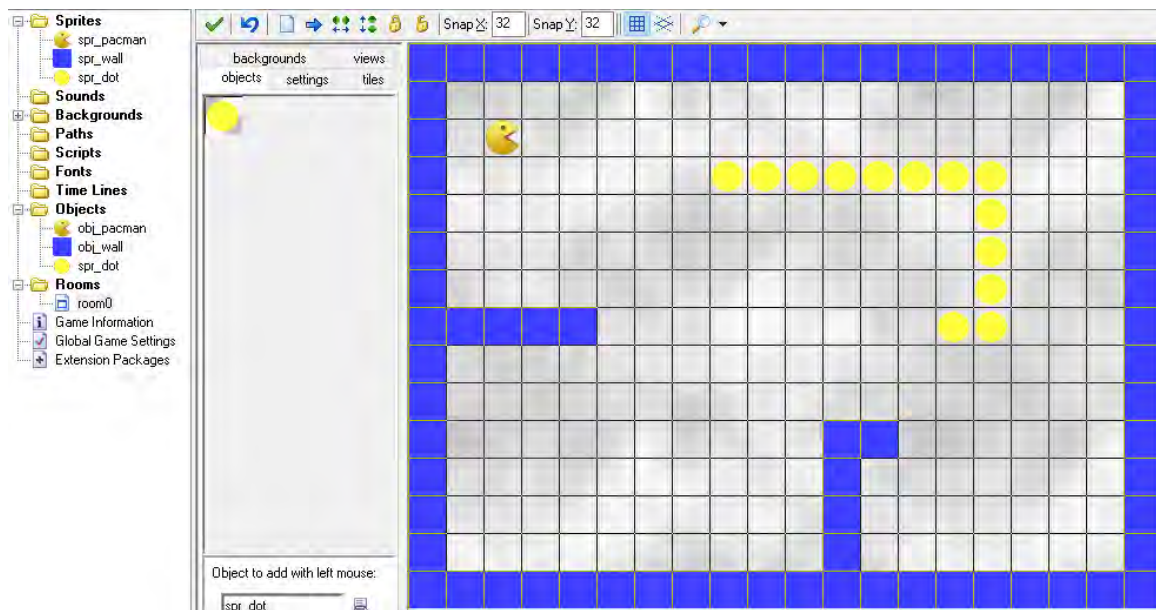
πίστα. (Συμβουλή: Κρατώντας πατημένο το shift και το αριστερό κουμπί του ποντικιού, τοποθετείτε μαζικά τέτοια αντικείμενα όπου σύρετε το mouse). Τελειώνοντας, η πίστα θα μοιάζει με την πιο κάτω εικόνα:



- (4) Να ανοίξετε την καρτέλα του obj_Pacman από την κατηγορία Objects και να προσθέσετε το event **Collision (σύγκρουση)** με το **obj_wall**, ώστε να καθορίσετε τη συμπεριφορά του Pacman όταν συγκρούεται με τον τοίχο.

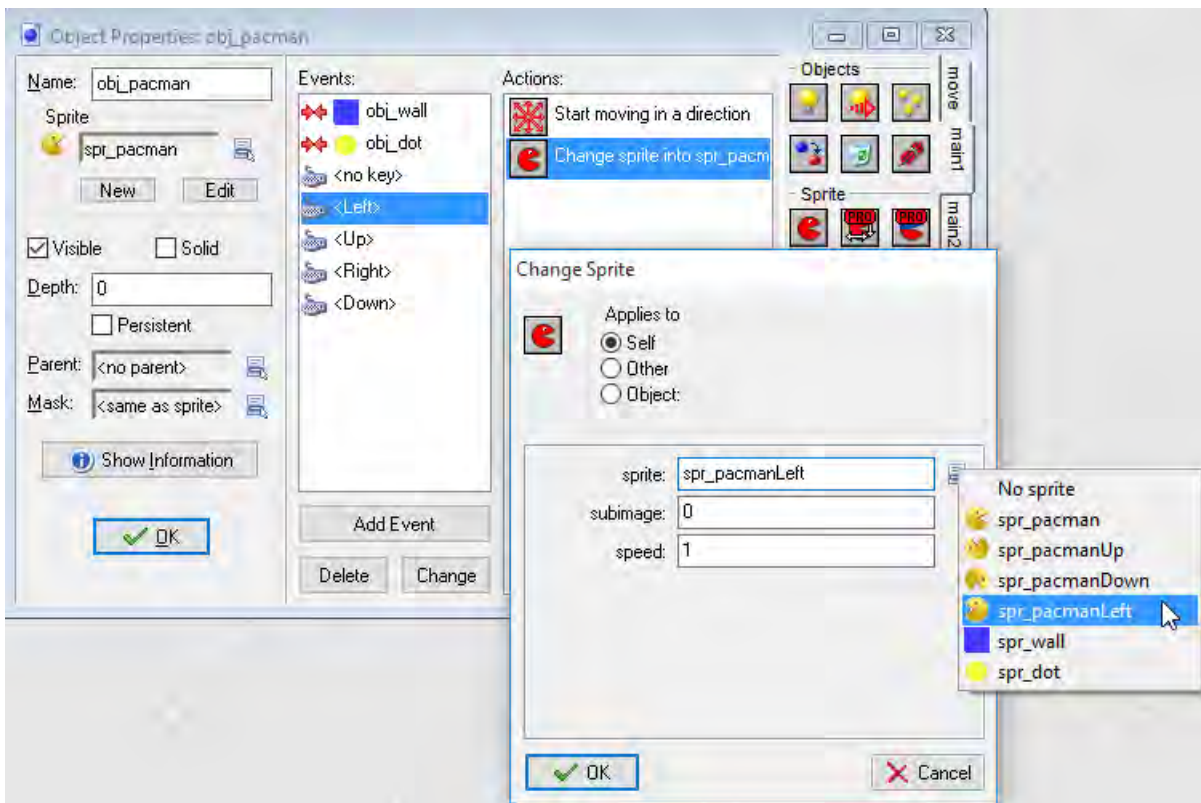


- (5) Στη συνέχεια, πρέπει καθορίσετε τις ενέργειες (actions) που θα εκτελεστούν όταν συμβεί αυτό το γεγονός (collision). Να σύρετε την ενέργεια **Move Fixed** από τη δεξιά μπάρα των ενεργειών στο πλαίσιο Actions και να πατήσετε στα **Directions** το **μμεσαίο κουμπί** (stop), έτσι ώστε ο χαρακτήρας σας να σταματάει (όπως φαίνεται στο πιο πάνω σχήμα δεξιά).
- (6) Να δοκιμάσετε το παιχνίδι σας.
- (7) Να προσθέσετε ένα νέο sprite σε σχήμα κύκλου, (spr_dot) και ένα νέο object (obj_dot), τα οποία θα αντιστοιχούν με τις κουκίδες που θα τρώει ο Pacman, όπως κάναμε στις διαδικασίες (α) και (β). Σημειώστε ότι το αντικείμενο δεν θα πρέπει να είναι solid.
- (8) Στη συνέχεια να τοποθετήσετε αντικείμενα αυτού του είδους σε όποια σημεία της πίστας επιθυμείτε.



- (9) Να προσθέσετε κάποιο αρχείο ήχου το οποίο θα ακούγεται όταν ο Pacman τρώει μία τελεία. Να εισαγάγετε ένα νέο ήχο από το αρχείο sound_chomp με όνομα snd_dot.
- (10) Να μεταβείτε στο **obj_dot** και να προσθέσετε το **event Collision** με **obj_Pacman**. Στα actions από τη δεξιά μπάρα να προσθέσετε τα εξής, για τους ισάριθμους στόχους της παρούσας προσθήκης:
- Σύρετε την **Play Sound** από την καρτέλα **Main1** και επιλέξτε τον ήχο snd_dot.
 - Σύρετε την **Destroy Instance** από την καρτέλα **Main1** και επιλέξτε το **Self**, έτσι ώστε να εξαφανίζεται η κουκίδα που έφαγε και όχι ο Pacman (στην περίπτωση που αφήσετε το προεπιλεγμένο self).
 - Σύρετε την **Set Score** από την καρτέλα **Score**. Για να αυξάνεται το σκορ κατά 1 βαθμό, να πληκτρολογήσετε 1 στο πλαίσιο **Set Score** και να επιλέξετε το κουτί **relative**, ώστε οι πόντοι να προστίθενται κάθε φορά στο σκορ. Το σκορ θα εμφανίζεται στον τίτλο του παιχνιδιού.
- (11) Να δοκιμάσετε το παιχνίδι σας.
- (12) Να παρατηρήσετε ότι ο χαρακτήρας Pacman εμφανίζεται να κοιτάζει συνεχώς δεξιά ανεξάρτητα από την κατεύθυνση που κινείται. Για να τον κάνουμε να κοιτάζει στην κατεύθυνση που κινείται, θα δημιουργήσουμε τέσσερα sprite με την εικόνα τροποποιημένη αντίστοιχα και θα χρησιμοποιούμε το κατάλληλο για την κάθε κατεύθυνση κίνησης. Να κάνετε δεξί κλικ στο spr_pacman → Duplicate τρεις φορές και να αλλάξετε τα ονόματα σε spr_pacmanUp, spr_pacmanDown, spr_pacmanLeft.
- (13) Να κάνετε **διπλό κλικ στο spr_pacmanUp → Edit Sprite → Transform → Rotate → 270 degrees → OK →** . Αυτή η ενέργεια περιστρέφει τον χαρακτήρα, ώστε να κοιτάζει προς τα πάνω. Αντίστοιχα να ενεργήσετε για το spr_pacmanDown, απλώς αντί για 270 degree (μοίρες) επιλέγετε **90 degrees**. Για το spr_pacmanLeft στο Transform να επιλέξετε **Mirror Flip**.
- (14) Τώρα, θα πρέπει κάθε φορά που θα πατάτε, για παράδειγμα, το αριστερό βέλος να αλλάζει το sprite στο spr_pacmanLeft. Στο object του Pacman να επιλέξετε το

event <Left> και στο Actions να προσθέσετε το Change Sprite. Στο πεδίο sprite να επιλέξετε το spr_pacmanLeft.



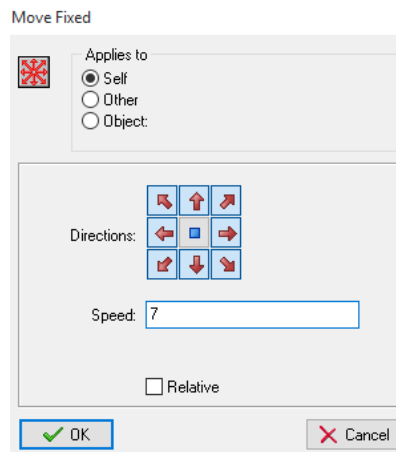
(15) Να προγραμματίσετε και τις άλλες κατευθύνσεις <Up>, <Right> και <Down>, αντίστοιχα, να δοκιμάσετε εκ νέου το παιχνίδι σας και να το αποθηκεύσετε.

Άσκηση 3.7 Το παιχνίδι Pacman

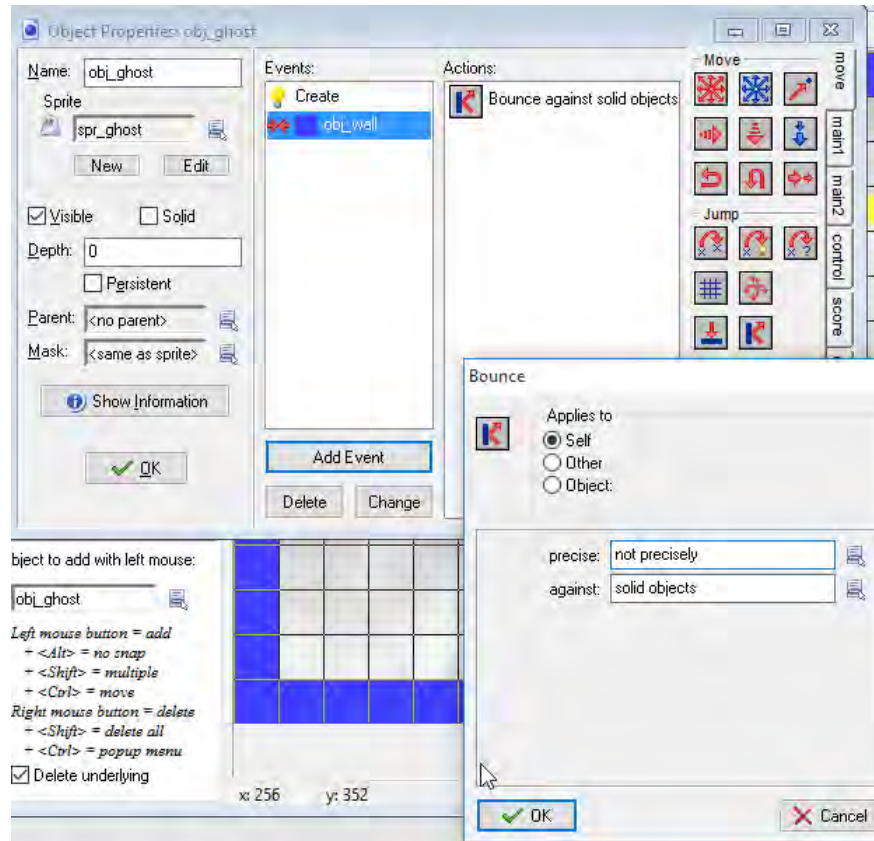
Στο παιχνίδι Pacman, αφού έχουμε ολοκληρώσει την κίνηση του χαρακτήρα Pacman, θα προσθέσουμε τους αντιπάλους του Pacman, τα φαντάσματα, σε μία υπάρχουσα πίστα όπου είναι τοποθετημένος. Τα φαντάσματα θα κινούνται μέσα στην πίστα και όταν συγκρουστούν με τον Pacman, θα χάνονται 2 βαθμοί και ο Pacman θα επιστρέφει στην αρχική του θέση (στο επόμενο κεφάλαιο θα ασχοληθούμε με τις ζωές του Pacman).

- (1) Να δημιουργήσετε το sprite για το φάντασμα, **spr_ghost**. Ως εικόνα για το sprite να επιλέξετε το αρχείο ghost.png, που βρίσκεται στον φάκελο που θα σας υποδείξει ο καθηγητής σας. Να τροποποιήσετε το sprite με την επιλογή **Remove Background**, ώστε να είναι διαφανές.
- (2) Ακολούθως, να δημιουργήσετε το αντίστοιχο object για το φάντασμα, **obj_ghost**. Όπως και με το αντικείμενο obj_pacman, δεν πρέπει να είναι επιλεγμένο το κουτί επιλογής solid.
- (3) Να καθορίσετε τη συμπεριφορά του κάθε φαντάσματος, ώστε να ξεκινάνε να κινούνται προς μία οποιαδήποτε τυχαία κατεύθυνση. Όταν φτάνουν σε κάποιον τοίχο να επιστρέφουν προς την αντίθετη κατεύθυνση. Για το αντικείμενο **obj_ghost** να καθορίσετε τα ακόλουθα events και actions:
 - **Add Event → Create**, ώστε να θέσετε την κίνηση του φαντάσματος κατά τη δημιουργία του (δηλαδή με το που εμφανίζεται στην πίστα). Στις ενέργειες (**Actions**) να σύρετε την **Move Fixed**, ώστε να κινείται προς μία τυχαία

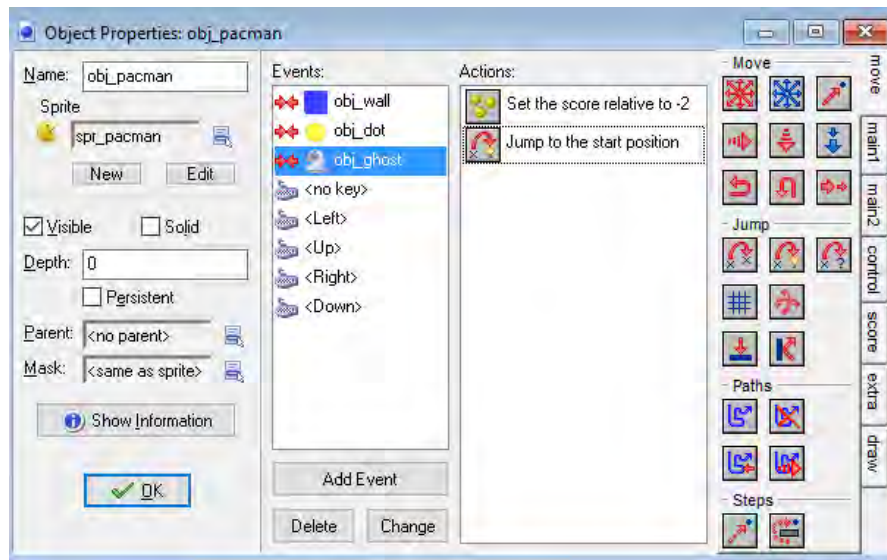
κατεύθυνση. Να επιλέξετε όλες τις κατευθύνσεις στο παράθυρο που ανοίγει (εκτός, βέβαια, από τη μεσαία) και να ορίσετε την ταχύτητα Speed σε 7.



- **Add Event → Collision with obj_Wall**, ώστε να επιστρέφει όταν πέφτει σε τοίχους: και στο **Actions** να σύρετε την ενέργεια **Bounce**.



- (4) Να καθορίσετε τη συμπεριφορά όταν ο Pacman έρχεται σε επαφή με κάποιο από τα φαντάσματα. Να κλείσετε το **obj_ghost** και να ανοίξετε το **obj_pacman**. Στα events να προσθέσετε **Collision → obj_ghost** και να σύρετε την ενέργεια **Set Score** από την καρτέλα score. Να πληκτρολογήσετε **-2** και να επιλέξετε το κουτί **relative**. Για να μεταβεί ο Pacman στην αρχική του θέση, να σύρετε την ενέργεια **Jump to start position** από την καρτέλα Move. Το ολοκληρωμένο event φαίνεται στο πιο κάτω σχήμα:



- (5) Να προσθέσετε όσα φαντάσματα και σε όποιες θέσεις θέλετε στην πίστα.
- (6) Να δοκιμάσετε το παιχνίδι σας για να βεβαιωθείτε ότι συμπεριφέρεται σύμφωνα με την πιο πάνω περιγραφή.
- (7) Να αποθηκεύσετε το παιχνίδι σας με όνομα pacman3.gmk.

+ Ασκήσεις Εμπλουτισμού

Άσκηση 3.8 Το παιχνίδι Catch the Clown

Στο παιχνίδι Catch the Clown που ήδη δημιουργήσατε:

- (1) Να προσθέσετε events για τις βόμβες, ώστε όταν ο παίκτης κάνει κλικ σε αυτές να εμφανίζεται το μήνυμα «BOOM! Game Over!!!», το σκορ και το παιχνίδι να τελειώνει.
- (2) Να προσθέσετε ακόμη ένα χρονόμετρο, ώστε κάθε 3 δευτερόλεπτα ο κλόουν να αλλάζει κατεύθυνση.

Άσκηση 3.9 Το παιχνίδι Pacman

Στο παιχνίδι Pacman που ήδη δημιουργήσατε:

- (1) Να δημιουργήσετε άλλο ένα αντικείμενο (και το αντίστοιχο sprite), ώστε αν πέσει πάνω του ο Pacman να χάνει 3 βαθμούς και να ακούγεται κάποιος άλλος ήχος.
- (2) Να δημιουργήσετε κάποιον άλλο τοίχο, ο οποίος όμως θα είναι «εύθραυστος», υπό την έννοια ότι αν πέσει πάνω του ο Pacman, ο τοίχος να σπάει (δηλαδή θα εξαφανίζεται). Να περικλείσετε κάποιες κουκίδες με τοίχο αυτού του είδους.

Άσκηση 3.10 Το παιχνίδι Pacman

Στο παιχνίδι Pacman που ήδη δημιουργήσατε:

- (1) Να προσθέσετε άλλο ένα φάντασμα, «πιο κακό», με το οποίο αν συγκρουστεί ο χαρακτήρας Pacman να χάνει περισσότερους βαθμούς. Για το sprite του «πιο

κακού» φαντάσματος, να δημιουργήσετε μία παραλλαγή του sprite για το φάντασμα που ήδη έχετε ή να ψάξετε στο Διαδίκτυο για μία κατάλληλη εικόνα.

- (2) Να κάνετε τις απαραίτητες αλλαγές έτσι ώστε κατά τη σύγκρουση του χαρακτήρα με ένα φάντασμα να ακούγεται επιπλέον κάποιος ήχος.
- (3) Να προσθέσετε ρύθμιση στο `obj_Ghost`, ώστε όταν συγκρούεται με τον χαρακτήρα Pacman να αυξάνεται η ταχύτητα της κίνησής του (σχετικά actions: `Speed Horizontal` και `Speed Vertical`).

B7.2.4 Ζωές, Σκορ και ο Ελεγκτής του παιχνιδιού

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ❖ Να προσθέτουμε ελεγκτή παιχνιδιού
- ❖ Να προσθέτουμε ζωές στον χαρακτήρα
- ❖ Να εμφανίζουμε τις διαθέσιμες ζωές
- ❖ Να ρυθμίζουμε τρόπους απώλειας ζωής
- ❖ Να εμφανίζουμε μηνύματα
- ❖ Να εμφανίζουμε τον πίνακα με τα υψηλότερα σκορ.

1. Εισαγωγή

Σε αυτό το κεφάλαιο θα προγραμματίσουμε τον ελεγκτή – controller του παιχνιδιού, έτσι ώστε να παρακολουθείτε αλλαγές σε βασικές παραμέτρους του παιχνιδιού, όπως τις ζωές, την εμφάνιση μηνυμάτων, την εμφάνιση του πίνακα των σκορ κ.α.

2. Υλοποίηση του ελεγκτή (controller) του παιχνιδιού

Ο ελεγκτής του παιχνιδιού συνήθως υλοποιείται ως ένα αντικείμενο, που δεν θα περιέχει κάποιο sprite, αλλά είναι υπεύθυνο για λειτουργίες, όπως ο έλεγχος των ζωών του παίκτη και άλλων σημαντικών στοιχείων του παιχνιδιού (είναι ο «επιβλέπων» του παιχνιδιού). Συνήθως περιλαμβάνει εξειδικευμένα συμβάντα (events) που εξυπηρετούν αυτόν τον ρόλο του. Τα πιο κύρια events που συνήθως καθορίζονται για τον ελεγκτή είναι:

Create: Όταν δημιουργηθεί ο ελεγκτής, δηλαδή στην αρχή του παιχνιδιού. Εδώ τοποθετούνται ενέργειες για αρχικοποίηση, όπως π.χ. ο αριθμός των ζωών του παίκτη και το αρχικό σκορ.

No more lives: Όταν ο παίκτης έχει χάσει όλες τις ζωές του. Εδώ συνήθως το παιχνίδι τελειώνει και εμφανίζονται κατάλληλα μηνύματα και ο πίνακας των σκορ.

Draw: Κάθε φορά που ξανασχεδιάζεται η οθόνη, συνήθως αρκετές φορές το δευτερόλεπτο. Συνήθως χρησιμοποιείται από τον ελεγκτή για να σχεδιάσει ή παρουσιάσει διάφορες πληροφορίες για την κατάσταση του παιχνιδιού, όπως το σκορ και ο αριθμός ζωών του παίκτη.

Step: Κάθε φορά που τελειώνει μία σειρά εκτέλεσης του παιχνιδιού, πολλές φορές το δευτερόλεπτο. Εδώ ο ελεγκτής συνήθως μελετά διάφορες παραμέτρους και αποφασίζει τι θα ακολουθήσει (π.χ. εάν δεν υπάρχουν άλλα αντικείμενα για συλλογή, ανοίγει την επόμενη πίστα).

Η παρουσία του ελεγκτή δεν είναι απαραίτητο συστατικό στοιχείο του παιχνιδιού. Είναι όμως ένας βολικός μηχανισμός για συγκέντρωση όλων των γενικών ελέγχων που χρειάζεται το παιχνίδι για να λειτουργήσει σε ένα αντικείμενο για εύκολη διαχείριση.

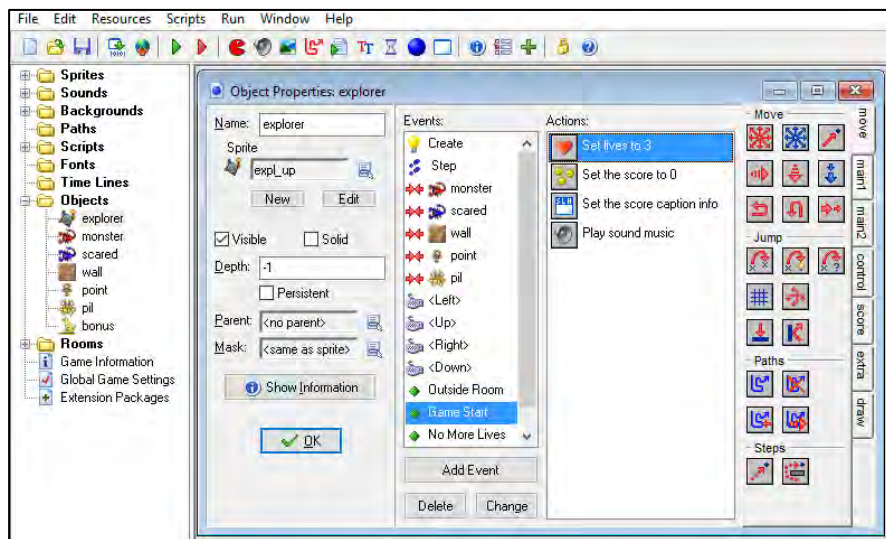


Ασκήσεις Κεφαλαίου

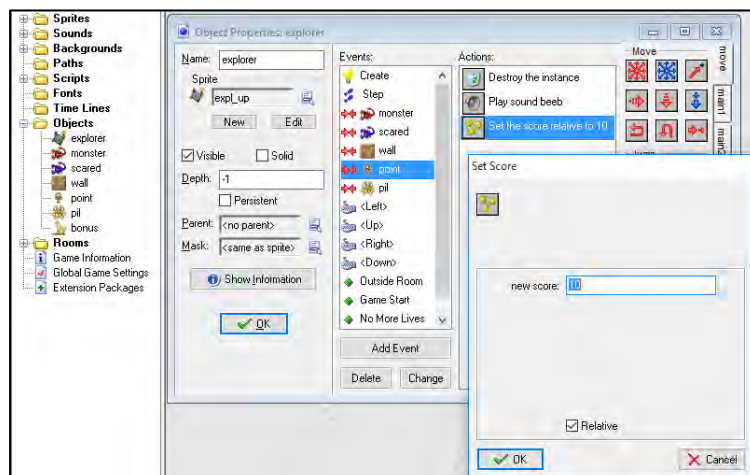
Άσκηση 4.1 Το παιχνίδι Treasure Island

Στο παιχνίδι **Treasure Island**, θα αλλάξουμε τον αριθμό αρχικών ζωών του εξερευνητή από 3 σε 10, καθώς και τον τρόπο βαθμολόγησης, ώστε με κάθε αντικείμενο που συλλέγει να παίρνει 20 αντί 10 βαθμούς και εάν πατήσει τον κόκκινο σκορπιό να χάνει 20 βαθμούς.

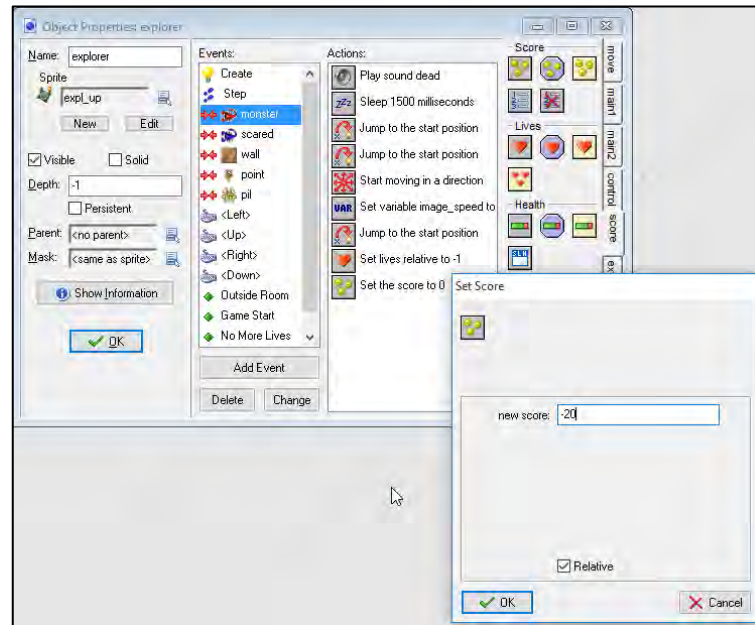
- (1) Για να αλλάξετε τις αρχικές ζωές του εξερευνητή από 3 σε 10, να επιλέξετε το αντικείμενο (object) **explorer**.
- (2) Έπειτα από τη στήλη Events το Game Start και να κάνετε διπλό κλικ από τα Actions το **Set lives**.



- (3) Να εφαρμόσετε την αλλαγή και ελέγξετε το αποτέλεσμα εκτελώντας το παιχνίδι.
- (4) Να κάνετε τώρα την αλλαγή στη λογική βαθμολόγησης του παιχνιδιού. Να επιλέξετε το αντικείμενο (object) **explorer**.
- (5) Έπειτα από τη στήλη Events το Game Start και να κάνετε διπλό κλικ από τα Actions τη σύγκρουση με το αντικείμενο **point**.
- (6) Ο εξερευνητής λαμβάνει 10 βαθμούς για κάθε αντικείμενο που συλλέγει. Να το αλλάξετε σε 20 βαθμούς.



- (7) Να προσθέσετε τώρα αρνητικό σκορ -20, σε περίπτωση που ο εξερευνητής συγκρουστεί με έναν κόκκινο σκορπιό. Να επιλέξετε από τα events τη σύγκρουση και κάτω από τα Actions να κάνετε drag and drop από την καρτέλα **score** το **Set Score**.
- (8) Θέτοντας new score -20, αφαιρεί 20 βαθμούς και κάνοντας κλικ στο **Relative**, λαμβάνει υπόψη την προηγούμενη βαθμολογία του. Λειτουργεί όπως η εντολή $\text{score} = \text{score} - 20$.



- (9) Να εκτελέσετε το παιχνίδι σας για να επιβεβαιώσετε ότι ο τρόπος βαθμολόγησης λειτουργεί όπως τον έχετε καθορίσει.

Άσκηση 4.2 Το παιχνίδι Octopus

Να ανοίξετε το παιχνίδι με όνομα **Octopus**. Στόχος του παιχνιδιού είναι να πετύχετε τα χταπόδια.

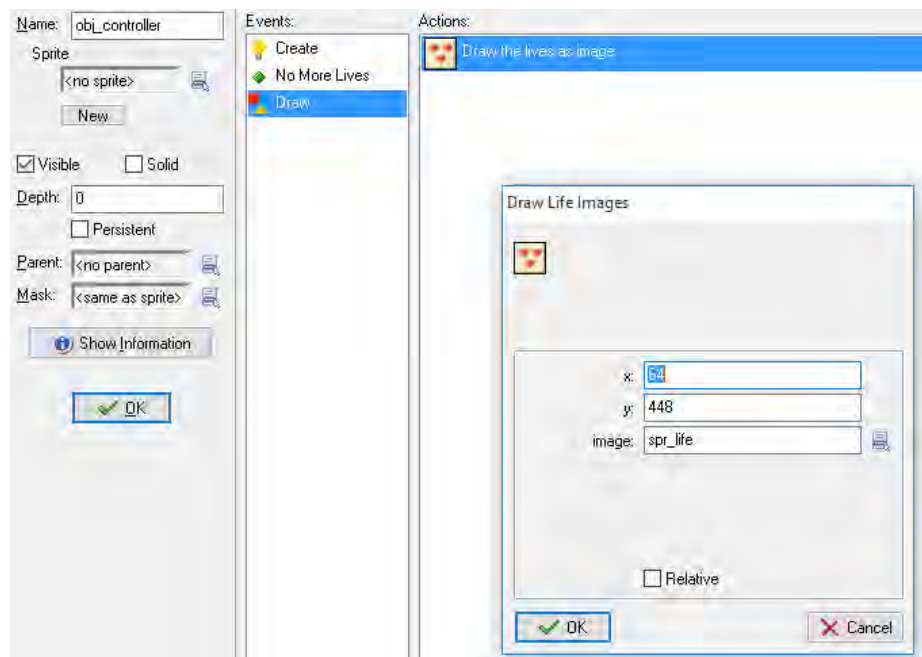
- (1) Να αλλάξετε τον αριθμό των ζώων σε 10 (να αναζητήσετε το αντικείμενο που «ελέγχει» το παιχνίδι).
- (2) Να αλλάξετε τον ρυθμό βαθμολόγησης ώστε σε κάθε περίπτωση να δίνονται διπλάσιοι βαθμοί.
- (3) Να ανοίξετε το room1 και να διαγράψετε ένα αντικείμενο χταποδιού και μερικά τουβλάκια της επιλογής σας, κάνοντας δεξί κλικ σε αυτά.
- (4) Να προσθέσετε ένα αντικείμενο στο room1. Να επιλέξετε την καρτέλα objects και ακριβώς κάτω από την καρτέλα, με αριστερό κλικ, να διαλέξετε το obj_octopus. Έπειτα, με αριστερό κλικ να το τοποθετήσετε στην πίστα (room).
- (5) Να αλλάξετε τη μουσική και τα ηχητικά εφέ του παιχνιδιού.

Άσκηση 4.3 Το παιχνίδι Pacman

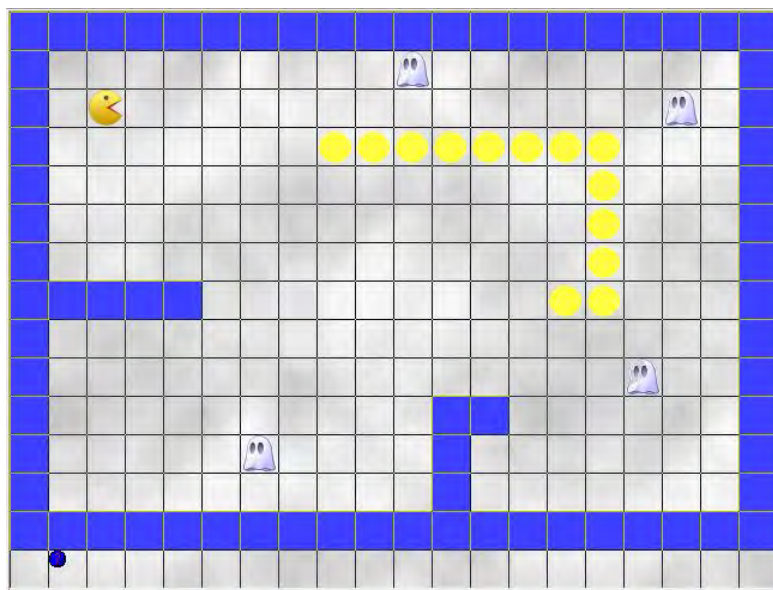
Θα δημιουργήσουμε έναν ελεγκτή για το παιχνίδι Pacman, για να διαχειρίζεται τις ζωές και το σκορ.

- (1) Να ανοίξετε το παιχνίδι pacman3.gmk.
- (2) Να δημιουργήσετε ένα νέο αντικείμενο και να το ονομάσετε **obj_Controller**. Το αντικείμενο αυτό δε θα έχει sprite.
- (3) Στα events του αντικειμένου να προσθέσετε το **Add Event → Create**, ώστε να θέσετε τις **αρχικές ρυθμίσεις** του παιχνιδιού: Να σύρετε στα actions την ενέργεια **Set Lives** από την καρτέλα score και να εισαγάγετε την τιμή 3, αν θέλετε να έχει 3 ζωές ο χαρακτήρας. Επίσης, να αρχικοποιήσετε το score σε 0 (**action Set Score**).
- (4) Να ορίσετε το τι θα γίνεται εάν ο παίκτης **δεν έχει άλλες ζωές**, με το να εμφανίζεται κάποιο μήνυμα (π.χ. Game over), στη συνέχεια να εμφανίζεται ο πίνακας High Score και ακολούθως το παιχνίδι να τελειώνει. Να προσθέσετε το event **Add Event → Other → No more lives**. Στα actions να σύρετε το **Display Message** από την καρτέλα **Main2** και πληκτρολογήστε το μήνυμα που θέλετε να εμφανίζεται. Στη συνέχεια να σύρετε την ενέργεια **Show Highscore** από την καρτέλα score και τελικά να σύρετε την ενέργεια **end game** από την καρτέλα Main2.
- (5) Για να εμφανίζονται οι ζωές που έχουν απομείνει στον παίκτη, θα χρησιμοποιήσουμε το event Draw. Για να εμφανίσουμε γραφικά τον αριθμό ζώων, θα δημιουργήσουμε ένα sprite για τις ζωές του χαρακτήρα, ώστε να ζωγραφίζονται σε κάποιο σημείο του παιχνιδιού, δηλώνοντας πόσες ζωές έχει. Για τον σκοπό αυτό, να κάνετε ένα αντίγραφο του spr_pacman (λειτουργία duplicate), να το ονομάσετε **spr_Life** και να μειώστε το μέγεθός του, επιλέγοντας **Edit sprite→Transform → Stretch** και να καθορίσετε **16 pixel**, ώστε το sprite μας να έχει μέγεθος 16x16.
- (6) Να προσθέσετε το συγκεκριμένο event, **Add Event → Draw**. Στόχος είναι να σχεδιάζονται στο κάτω μέρος του παραθύρου οι ζωές του παίκτη, χρησιμοποιώντας το spr_Life και αυτόματα να αφαιρείται μία εικόνα spr_Life, σε περίπτωση που χάνει ζωή ο παίκτης. Στα actions να σύρετε το **Draw the lives as images** και εισαγάγετε τις πιο κάτω ρυθμίσεις:
 - Image: spr_life
 - X: 64
 - Y: 448

Τα X, Y είναι οι συντεταγμένες εμφάνισης των ζώων. Μπορείτε να πειραματιστείτε για τη θέση μέσα από το room. Το X=0 και Y=0 είναι το πάνω αριστερά σημείο του παραθύρου του παιχνιδιού.



- (7) Στο τέλος να τοποθετήσετε στην πίστα, σε κάποιο σημείο, το `obj_controller` το οποίο, επειδή δεν έχει συνδεδεμένο `sprite`, εμφανίζεται ως αγγλικό ερωτηματικό. Να κάνετε επίσης τις απαραίτητες τροποποιήσεις ώστε ο χώρος εμφάνισης των διαθέσιμων ζωών να είναι εκτός των ορίων του παίκτη (έξω από τον τοίχο). Με δεξί κλικ να αφαιρέσετε το κάτω μέρος του τοίχου και (αφού επιλέξετε το αντικείμενο `obj_wall`) με αριστερό κλικ να προσθέσετε τον τοίχο πιο ψηλά.



- (8) Να προσθέσετε στα `actions` του κατάλληλου `event` το `obj_pacman`, την ενέργεια **Set lives** με επιλεγμένο το **Relative** και τιμή **-1**, έτσι ώστε, όταν συγκρούεται με φάντασμα, να χάνει μία ζωή.
- (9) Να εκτελέσετε και να δοκιμάσετε το παιχνίδι σας για να βεβαιωθείτε ότι λειτουργεί σύμφωνα με την πιο πάνω περιγραφή.
- (10) Να αποθηκεύσετε το παιχνίδι σας με το όνομα `pacman4.gmk`.

+ Ασκήσεις Εμπλουτισμού**Άσκηση 4.4** Το παιχνίδι Pacman

Στο παιχνίδι Pacman που δημιουργήσατε:

- (1) Να κάνετε τις απαραίτητες αλλαγές έτσι ώστε, αν δεν υπάρχουν άλλες ζωές να ακούγεται επιπλέον κάποιος ήχος.
- (2) Να κάνετε τις απαραίτητες αλλαγές έτσι ώστε, εκτός από την γραφική απεικόνιση των διαθέσιμων ζώων, να εμφανίζονται και αριθμητικά οι ζωές.

B7.2.5 Δομή διακλάδωσης

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ◆ Να βελτιώνουμε τη συνολική λογική του παιχνιδιού
- ◆ Να προσθέτουμε δομή διακλάδωσης
- ◆ Να προσθέτουμε ένθετη δομή διακλάδωσης
- ◆ Να προσθέτουμε νέο room
- ◆ Να δημιουργούμε μία εισαγωγική οθόνη ενός παιχνιδιού

1. Εισαγωγή

Έχουμε πετύχει μέχρι τώρα να δημιουργήσουμε σχετικά σύνθετα παιχνίδια, χωρίς να χρειαστεί κάποια δομή διακλάδωσης (π.χ. if-then-else), η οποία θα ήταν απαραίτητη σχεδόν από την αρχή σε μία πιο «παραδοσιακή» γλώσσα προγραμματισμού. Αυτό οφείλεται στον τρόπο προσέγγισης του GameMaker ο οποίος βασίζεται σε συμβάντα (events) πίσω από τα οποία κρύβεται η λογική της δομής διακλάδωσης. Υπάρχουν, όμως, περιπτώσεις όπου η δομή διακλάδωσης χρειάζεται, γι' αυτό και το GameMaker την υποστηρίζει.

2. Η δομή διακλάδωσης

Στην κάρτα **Control** υπάρχουν διάφοροι έτοιμοι έλεγχοι συνθηκών. Για τον σκοπό αυτού του μαθήματος οι πιο χρήσιμοι είναι:



Test Instance Count: Ελέγχει κατά πόσο στο παιχνίδι υπάρχει συγκεκριμένος αριθμός αντικειμένων συγκεκριμένου είδους (π.χ. εάν υπάρχουν 2 αντικείμενα τύπου obj_dot).



Check Question: παρουσιάζει ένα ερώτημα στον χρήστη το οποίο μπορεί να απαντήσει με ναι (yes) ή όχι (no).



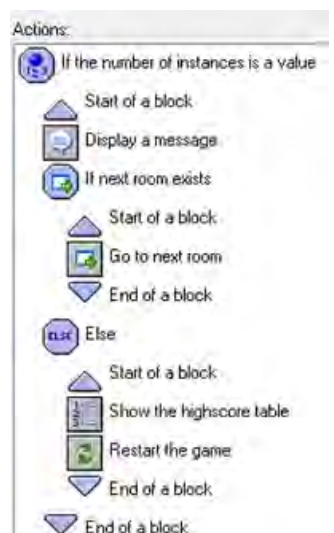
Test Expression: Ελέγχει μία έκφραση που καθορίζει ο προγραμματιστής (π.χ. count >=10). Αποτελεί τη γενικότερη μορφή δομής διακλάδωσης.

Όπως και σε πολλές γλώσσες προγραμματισμού η δομή εφαρμόζεται στην αμέσως ακόλουθη ενέργεια, δηλαδή εάν τη δομή διακλάδωσης ακολουθούν τρεις ενέργειες, μόνον η πρώτη θα εκτελείται εάν το αποτέλεσμα της συνθήκης είναι ορθό (true/yes). Για να ομαδοποιήσουμε ενέργειες (actions) ώστε να εκτελούνται ως αποτέλεσμα της συνθήκης, χρησιμοποιούμε τα σύμβολα της αρχής (▲) και του τέλους (▼) ενός μπλοκ. Αυτά έχουν τον ρόλο του **begin** και του **end** στη γλώσσα προγραμματισμού Pascal ή του { και του } στη γλώσσα προγραμματισμού C++. Επίσης παρέχεται το



Else ως ξεχωριστό σύμβολο, το οποίο τοποθετείται μόνο όταν χρειάζεται.

Στην εικόνα πάνω δεξιά δίνεται ένα παράδειγμα με απλή δομή διακλάδωση και χρήση μπλοκ. Είναι επίσης δυνατό να έχουμε ένθετη δομή διακλάδωσης (τη μία μέσα στην άλλη, όπως φαίνεται στην εικόνα κάτω δεξιά).



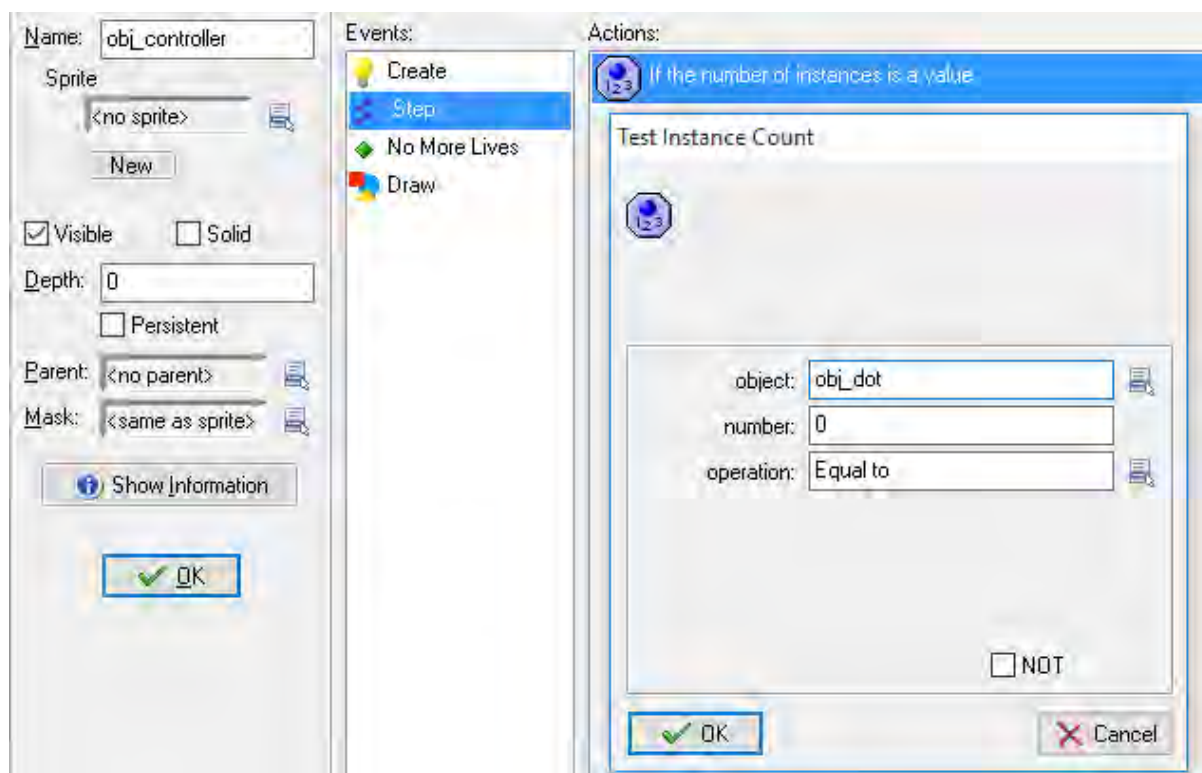


Ασκήσεις Κεφαλαίου

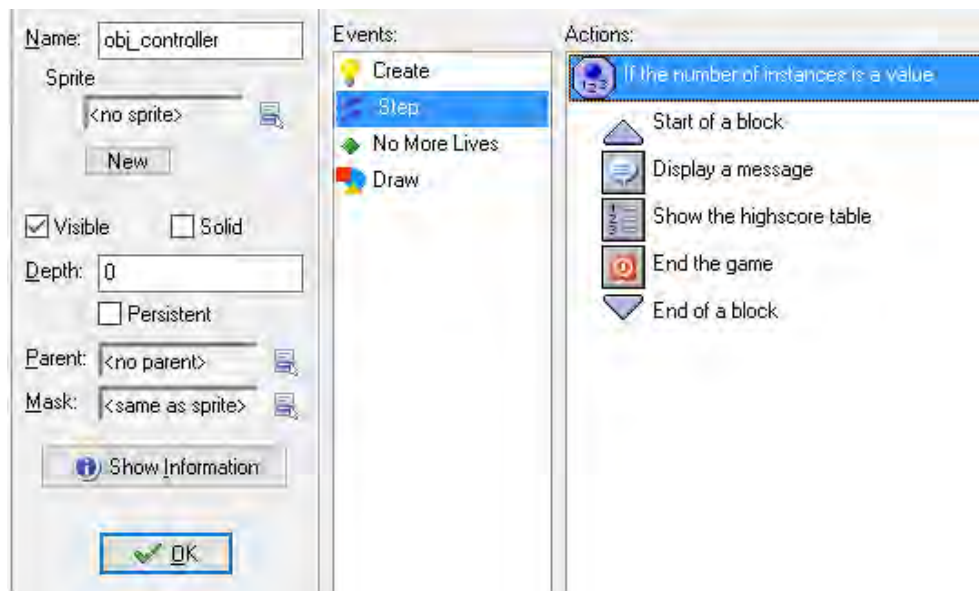
Άσκηση 5.1 Το παιχνίδι Pacman

Να τροποποιήσετε τον ελεγκτή που δημιουργήσατε για το παιχνίδι Pacman, ώστε αν έχουν τελειώσει οι κουκίδες να τερματίζει το παιχνίδι, εμφανίζοντας κάποιο μήνυμα και τον πίνακα των σκορ.

- (1) Στο αντικείμενο του **controller**, να προσθέσετε το καινούριο event **Step: Add Event→ Step → Step**. Αυτό το event εκτελείται κάθε φορά που εκτελείται ένα βήμα του παιχνιδιού (δηλαδή αφού κινηθούν όλα τα αντικείμενα σύμφωνα με τους κανόνες κίνησής τους). Εκτελούνται πολλά τέτοια βήματα κάθε δευτερόλεπτο, συνήθως 30 (καθορίζονται από την ιδιότητα Speed της πίστας, συνήθως).
- (2) Από την καρτέλα **Control** να προσθέσετε στα **Actions** του event **Step** το **If the number of instances is a value**, ώστε να ελέγχει αν οι κουκίδες τελείωσαν, δηλαδή έχουν την τιμή 0, όπως φαίνεται στην πιο κάτω εικόνα:



- (3) Θα πρέπει όταν τελειώνουν οι κουκίδες, να εμφανίζεται το μήνυμα «Congratulations», να παρουσιάζεται ο πίνακας των high scores και να τελειώνει το παιχνίδι. Επομένως, να διαμορφώσετε τις ενέργειες όπως φαίνονται στην πιο κάτω εικόνα (σημειώστε ότι θα πρέπει να τις τοποθετήσετε μέσα σε μπλοκ). Τα actions **Display Message** και **End Game** θα τα βρείτε στην καρτέλα **Main2**. Στο Display Message να εμφανίσετε το μήνυμα «Congratulations».

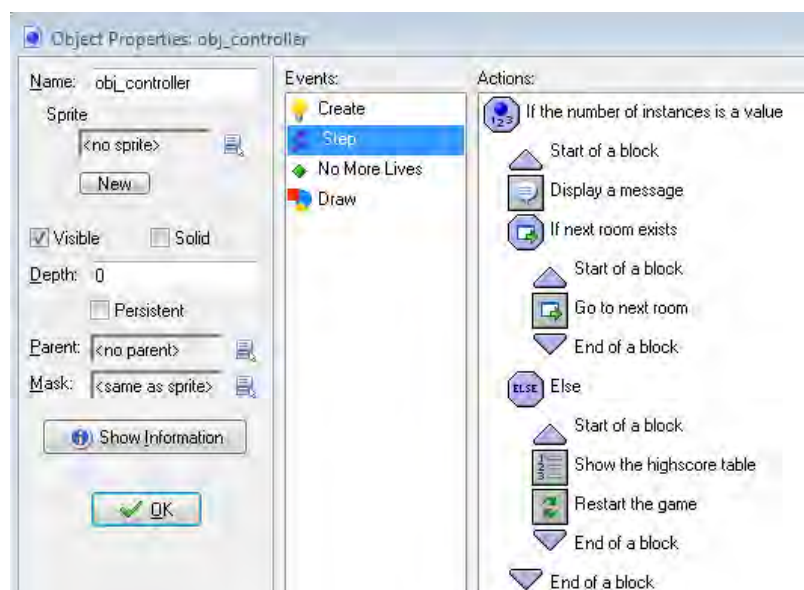


- (4) Να εκτελέσετε το παιχνίδι σας για να βεβαιωθείτε ότι λειτουργεί ορθά η πιο πάνω διαδικασία και να το αποθηκεύσετε.

Άσκηση 5.2 Το παιχνίδι Pacman

Να τροποποιήσετε τον ελεγκτή που δημιουργήσατε για το παιχνίδι Pacman, ώστε αν έχουν τελειώσει οι κουκίδες να προχωρεί στην επόμενη πίστα και στην περίπτωση που το παιχνίδι βρίσκεται στην τελευταία πίστα, να τυπώνει κάποιο μήνυμα και τον πίνακα των σκορ και στη συνέχεια να αρχίζει το παιχνίδι από την αρχή.

- (1) Να τροποποιήσετε το event **Step** στο obj_controller ώστε όταν τελειώνουν οι κουκίδες, να ελέγχει αν υπάρχει επόμενη πίστα τότε να προχωρά σε αυτήν, διαφορετικά να τυπώνει ένα σχετικό μήνυμα και να ξεκινάει το παιχνίδι από την αρχή. Θα χρειαστεί να τοποθετήσετε τη δομή διακλάδωσης **If next room exists** και την ενέργεια (action) **Go to next room**, όπως φαίνεται στην πιο κάτω εικόνα:



- (2) Για να έχουν εφαρμογή αυτά τα actions, θα πρέπει να δημιουργήσουμε κάποιες επιπρόσθετες πίστες (rooms). Για δική μας ευκολία, θα δημιουργήσουμε

αντίγραφο της πίστας που ήδη έχουμε. Να επιλέξετε το room και με το δεξί κουμπί το Duplicate. Θα δημιουργηθεί ένα αντίγραφο του δωματίου. Να του δώσετε κατάλληλο όνομα και να κάνετε κάποιες μικρές αλλαγές (π.χ. να προσθέσετε ή να αλλάξετε τους τοίχους ώστε να το ξεχωρίζετε από το αρχικό).

- (3) Να επαναλάβετε τη διαδικασία ακόμη μία φορά ώστε να έχετε άλλη μία πίστα (δηλαδή συνολικά τρεις).
- (4) Να εκτελέσετε το πρόγραμμά σας για να επιβεβαιώσετε ότι όταν ο Pacman φάει όλες τις τελείες, τότε προχωρά στο επόμενο δωμάτιο και στο τέλος το παιχνίδι ξεκινά από την αρχή.
- (5) Να αποθηκεύσετε το παιχνίδι σας με το όνομα pacman5.gmk.

Άσκηση 5.3 Το παιχνίδι Arkanoid

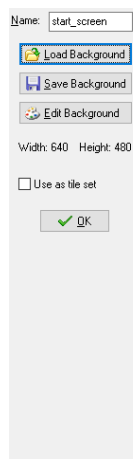
Να δημιουργήσετε ένα παιχνίδι, το οποίο ονομάζεται arkanoid. Πρόκειται για ένα κλασικό παιχνίδι τύπου arcade, το οποίο δημιουργήθηκε το 1985 από την ιαπωνική εταιρία Taito.

Η λογική του παιχνιδιού είναι ότι χειρίζεστε μία ρακέτα στο κάτω μέρος της οθόνης και την ελέγχετε, μετακινώντας την αριστερά ή δεξιά. Υπάρχει μία μπάλα, που όταν χτυπά στην ρακέτα κάνει γκελ, ενώ όταν χτυπά στα τούβλα που υπάρχουν στο πάνω μέρος της πίστας, αυτά εξαφανίζονται. Στόχος σας είναι να εξαφανίσετε όλα τα τούβλα που υπάρχουν.

Στο παιχνίδι αυτό θα έχουμε οθόνη εισαγωγής από όπου όταν πατήσει ο χρήστης κάποιο πλήκτρο ή το ποντίκι θα μεταφέρεται στην πίστα του παιχνιδιού.



- (1) Να δημιουργήσετε ένα νέο Background, start_screen, και να φορτώσετε την εικόνα της οθόνης εισαγωγής από το αρχείο που θα σας υποδείξει ο/η καθηγητής/τρια σας.



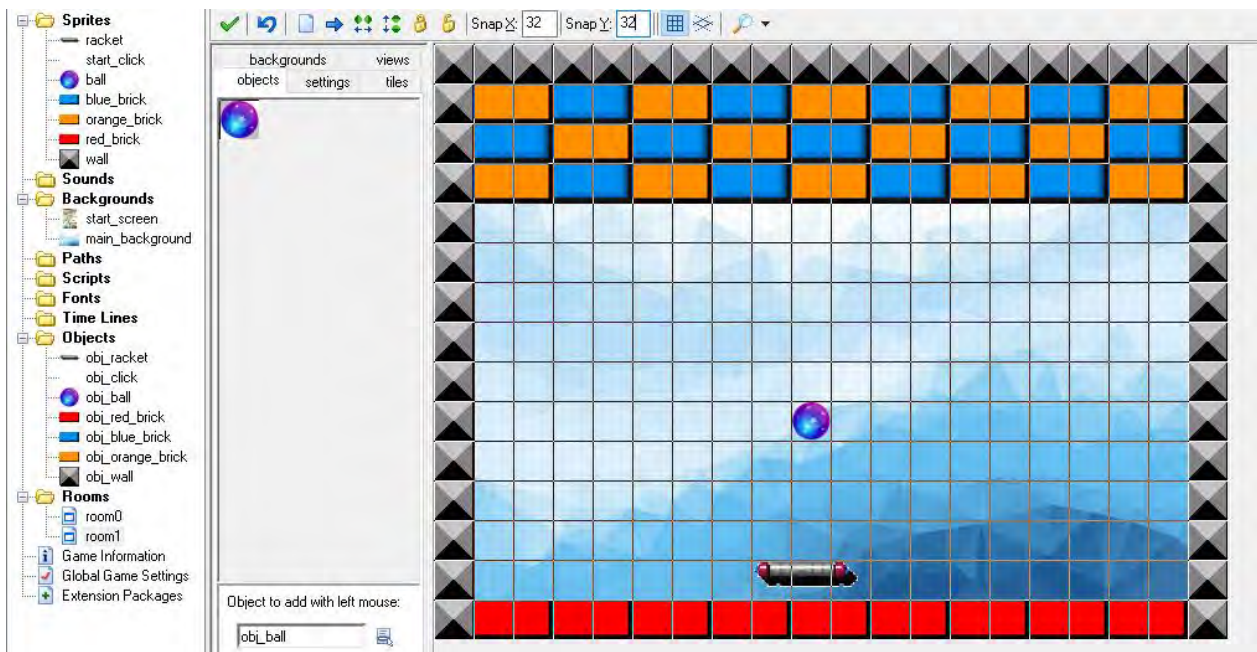
- (2) Να δημιουργήσετε ένα αρχικό room, το οποίο θα είναι η οθόνη εισαγωγής του παιχνιδιού. Να αλλάξετε το snap X και το snap Y σε 32, στο backgrounds να επιλέξετε το start_screen.
- (3) Θέλετε, τώρα, μόλις κάνει κλικ οπουδήποτε πάνω στην οθόνη, να προχωρά στο κυρίως παιχνίδι. Η λογική είναι ότι θα δημιουργήσετε ένα αντικείμενο χωρίς εικόνα, με τις ίδιες διαστάσεις του room (640x480) και μόλις κάνετε κλικ, να προχωρά στο επόμενο room.

- (4) Να δημιουργήσετε ένα νέο sprite **χωρίς εικόνα** με όνομα `spr_start_click` και διαστάσεις όπως και η πίστα (Width: 640, Height: 480).
- (5) Να δημιουργήσετε ένα object με όνομα `obj_start_click`, χρησιμοποιώντας το sprite `spr_start_click`. Να **απενεργοποιήσετε** το **precise collision checking**, να πατήσετε το κουμπί Modify Mask και να επιλέξετε **Full Image**.
- (6) Να προσθέσετε στο `obj_start_click` ένα event για να χειρίζεται το πάτημα του κουμπιού του ποντικιού (Add Event→ Mouse → Left button). Στα Actions, από την καρτέλα Main1 να εισαγάγετε Next Room, με transition της επιλογής σας.
- (7) Να προσθέσετε στο `obj_start_click` ένα δεύτερο event για να χειρίζεται το πάτημα πληκτρού στο πληκτρολόγιο (Add Event→Keyboard→<Any key>). Τα Actions να είναι όπως ακριβώς και στο (στ).
- (8) Να δημιουργήσετε μία νέα πίστα `room1` και να βεβαιωθείτε ότι είναι κάτω από την πίστα `room0` (η σειρά αυτή είναι και η σειρά που θα έχουν στο παιχνίδι).
- (9) Στο `room0` να προσθέσετε το αντικείμενο `obj_start_click`, το οποίο δεν θα φαίνεται, αφού δεν έχει sprite.
- (10) Να αποθηκεύσετε το παιχνίδι σας με το όνομα `arkanoid1.gmk` και να το δοκιμάσετε. Αν όλα πήγαν καλά, θα εμφανίζεται η αρχική οθόνη και όπου κάνουμε κλικ με το mouse ή όταν πατούμε κάποιο πλήκτρο στο πληκτρολόγιο θα εμφανίζεται το επόμενο room κενό.

Άσκηση 5.4 Το παιχνίδι Arkanoid

Στην άσκηση αυτή θα δημιουργήσουμε τα βασικά αντικείμενα και την πίστα του παιχνιδιού Arkanoid καθώς και τον τρόπο συμπεριφοράς (events και actions) των αντικειμένων.

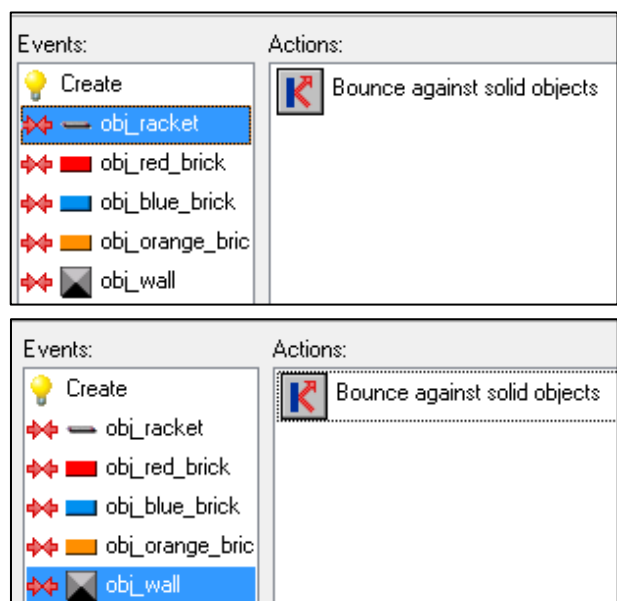
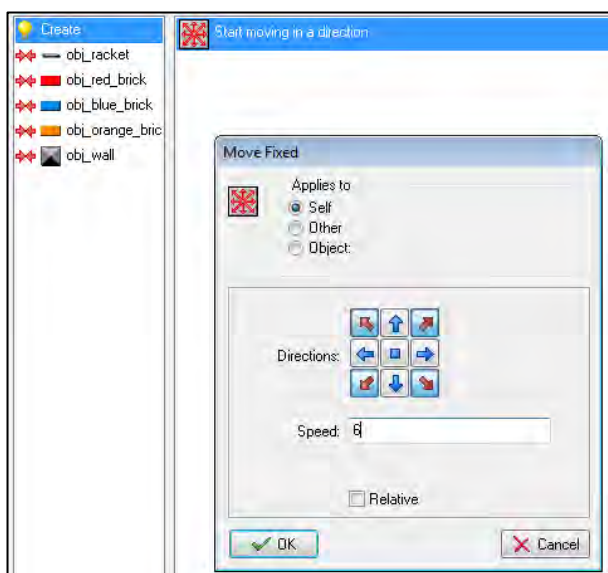
- (1) Να δημιουργήσετε ένα νέο Background από το αρχείο `Main_background.jpg` για την πίστα του παιχνιδιού (`room1`).
- (2) Στο `room1` να αλλάξετε το snap X και το snap Y σε 32, να επιλέξετε το Background που μόλις δημιουργήσατε και να κάνετε κλικ στο stretch, ώστε να καταλάβει όλη την πίστα.
- (3) Να δημιουργήσετε τα έξι (6) sprites που χρειάζεται το παιχνίδι (`spr_racket`, `spr_ball`, `spr_red_brick`, `spr_blue_brick`, `spr_orange_brick` και `spr_wall`), φορτώνοντας τις αντίστοιχες εικόνες από αρχεία.
- (4) Να δημιουργήσετε τα έξι (6) objects που χρειάζεται το παιχνίδι (`obj_racket`, `obj_ball`, `obj_red_brick`, `obj_blue_brick`, `obj_orange_brick` και `obj_wall`). Για καθένα από αυτά να επιλέξετε το αντίστοιχο sprite και να επιλέξετε να είναι **solid**.
- (5) Να τοποθετήσετε τα αντικείμενα στην πίστα ώστε αυτή να είναι όπως στην πιο κάτω εικόνα:

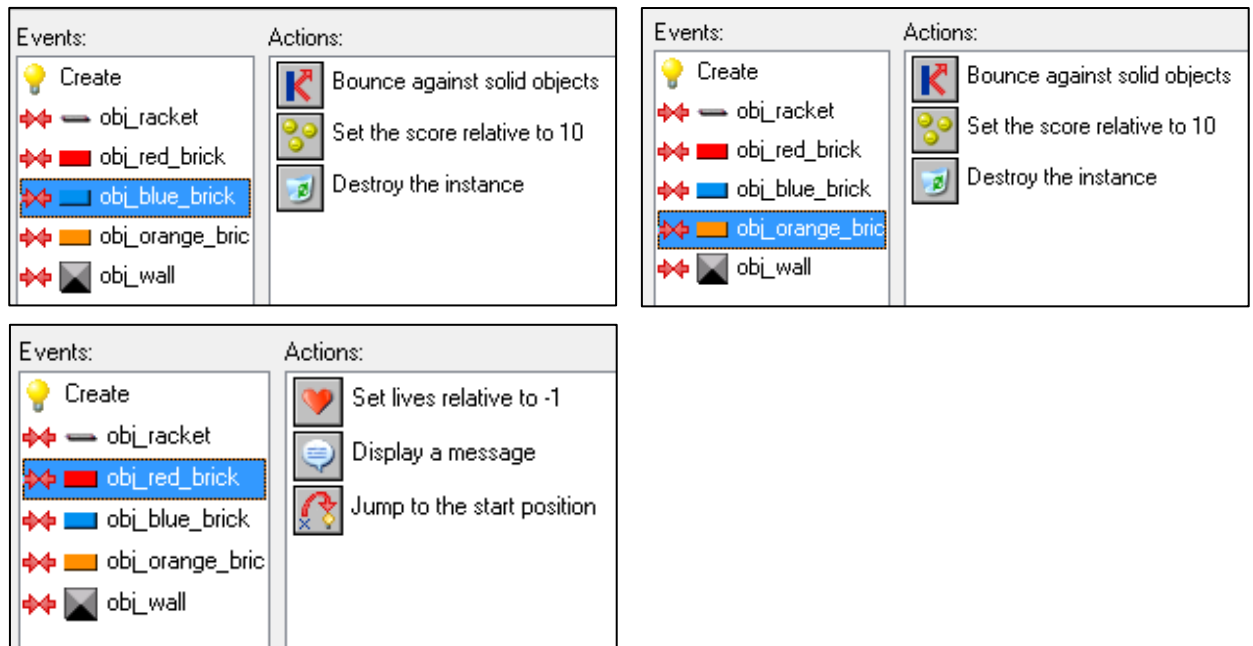


(6) Να προγραμματίσετε το obj_ball, ώστε:

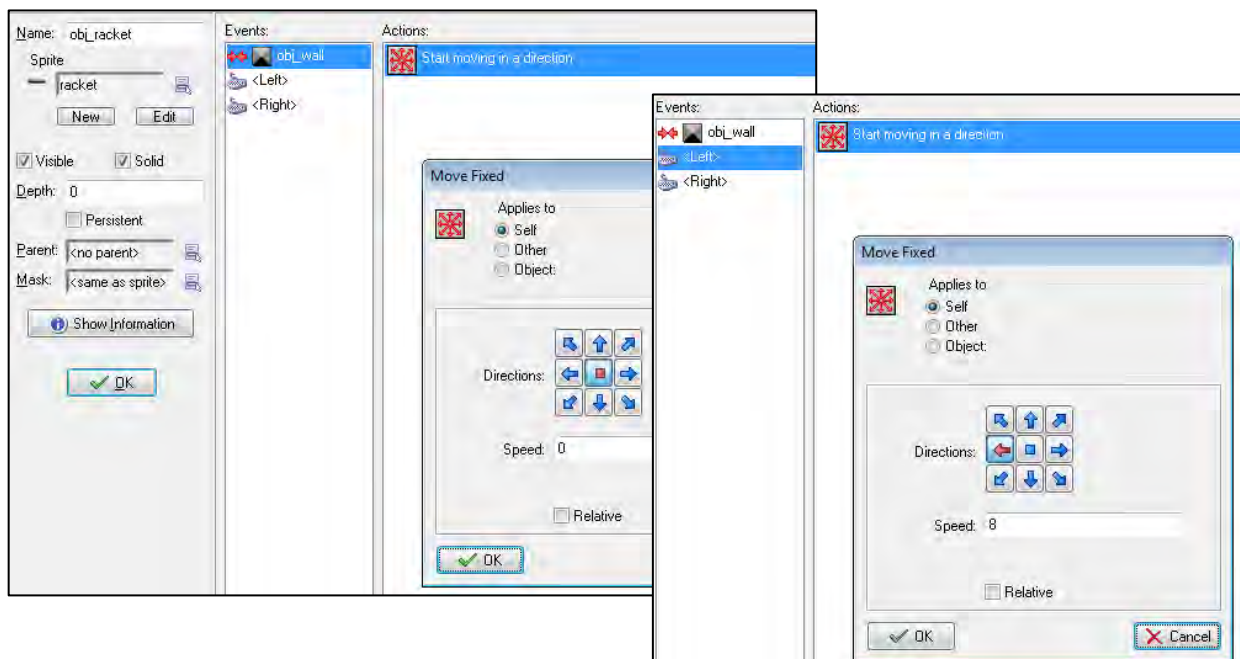
- Να κινείται αρχικά διαγώνια προς όλες τις κατευθύνσεις.
- Να κάνει bounce, όταν χτυπά (collision) στα αντικείμενα obj_wall, obj_racket, obj_blue_brick και obj_orange_brick.
- Επιπλέον, όταν χτυπά στο μπλε ή στο πορτοκαλί τούβλο, αυτά να εξαφανίζονται.
- Όταν χτυπά στο obj_red_brick να χάνει μία ζωή.

Συνοπτικά ο προγραμματισμός για το αντικείμενο obj_ball φαίνεται στις πιο κάτω εικόνες:





- (7) Έπειτα, να προγραμματίσετε την κίνηση της ρακέτας. Η ρακέτα κινείται δεξιά και αριστερά με τα αντίστοιχα πλήκτρα από το πληκτρολόγιο και σταματά να κινείται όταν ακουμπήσει τον τοίχο. Στις πιο κάτω εικόνες φαίνεται το πρόγραμμα για τον τοίχο και για το αριστερό κουμπί (να προγραμματίσετε και το δεξί κουμπί με παρόμοιο τρόπο):

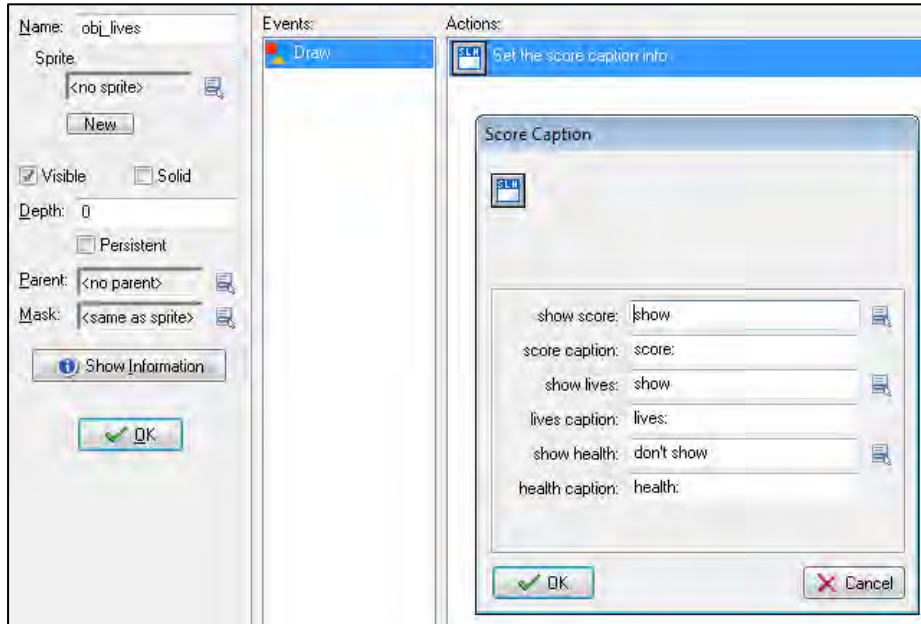


- (8) Να αποθηκεύσετε το παιχνίδι σας με το όνομα arkanoid2.gmk και να το δοκιμάσετε. Το παιχνίδι σας πρέπει να λειτουργεί, με τη ρακέτα να ελέγχεται δεξιά και αριστερά με τα πλήκτρα, τη μπάλα να κινείται και όταν χτυπά σε πορτοκαλί ή σε μπλε τούβλο αυτά να εξαφανίζονται.

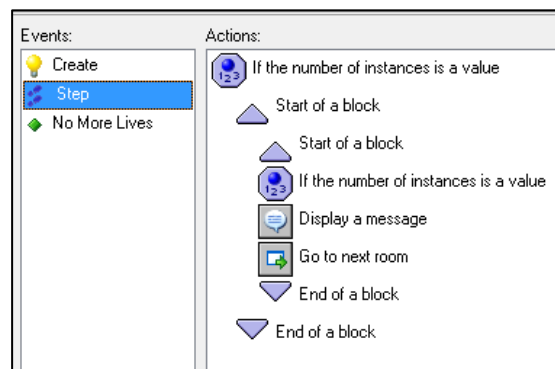
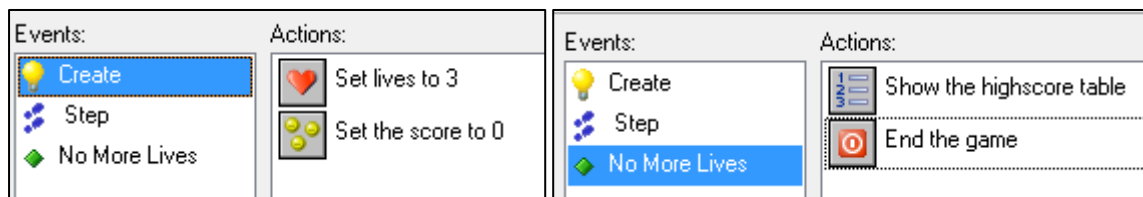
Άσκηση 5.5 Το παιχνίδι **Arkanoid**

Στην άσκηση αυτή θα προσθέσουμε τον ελεγκτή στο παιχνίδι **Arkanoid** καθώς και την ένδειξη των ζώων και του σκορ.

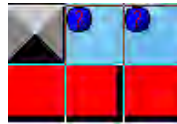
- (1) Για να εμφανίζονται ο αριθμός ζώων και το σκορ στον τίτλο του παραθύρου, να δημιουργήσετε ένα **αντικείμενο**, `obj_lives`, **χωρίς sprite** και εισαγάγετε σε αυτό το event draw με τα πιο κάτω actions (αυτή η τεχνική είναι διαφορετική από εκείνη που χρησιμοποιήσαμε στο παιχνίδι Pacman):



- (2) Να τοποθετήσετε επίσης τον ελεγκτή του παιχνιδιού, το αντικείμενο `obj_controller`, το οποίο να ορίζει τον αρχικό αριθμό ζώων και το αρχικό σκορ. Έπειτα να ελέγχει τις ζωές και αν είναι ίσες με 0 να δείχνει τον πίνακα των σκορ και να τερματίζει το παιχνίδι, ενώ όταν ολοκληρώνεται με επιτυχία μία πίστα, να προχωρεί στην επόμενη (το επόμενο επίπεδο). Αυτά μοιάζουν με τον ελεγκτή του παιχνιδιού Pacman και η υλοποίησή τους φαίνεται στις πιο κάτω εικόνες:



- (3) Να προσθέσετε τα αντικείμενα `obj_lives` και `obj_controller` στο `room1`, σε έναν κενό χώρο (θα εμφανίζονται ως ερωτηματικά)



- (4) Να δημιουργήσετε ένα επιπρόσθετο δωμάτιο
- (5) Να αποθηκεύσετε το παιχνίδι σας με το όνομα `arkanoid3.gmk` και να το δοκιμάσετε παρατηρώντας τη λειτουργία του `controller` και της ένδειξης με το σκορ και του αριθμού ζώων.

+ Ασκήσεις Εμπλουτισμού

Άσκηση 5.6 Το παιχνίδι Pacman

- (1) Να προσθέσετε μουσική και ήχους στο παιχνίδι σας.
- (2) Να προσθέσετε ακόμη ένα αντικείμενο, το φρούτο (`obj_fruit`), παρόμοιο με την τελεία στο παιχνίδι σας, ώστε όταν το φάει ο Pacman να κερδίζει 3 βαθμούς. Να εισαγάγετε ένα νέο `sprite` σε σχήμα φρούτου για το αντικείμενο αυτό. Θα χρειαστούν αλλαγές στον `Controler` σας, ώστε να ελέγχει ότι δεν υπάρχουν ούτε τελείες, ούτε φρούτα για να προχωρήσει στο επόμενο επίπεδο.

Άσκηση 5.7 Το παιχνίδι Arkanoid

- (1) Να προσθέσετε μουσική και ήχους στο παιχνίδι σας.
- (2) Να προσθέσετε ακόμη ένα είδος τούβλου με χρώμα μαύρο, το οποίο όταν το χτυπήσει η μπάλα θα αλλάζει σε τούβλο με χρώμα πορτοκαλί και θα δίνει 20 βαθμούς (θα χρειαστείτε το `Action Change Instance` από την καρτέλα `Main1`).
- (3) Να προσθέσετε και άλλες πίστες στο παιχνίδι σας με διαφορετικό βαθμό δυσκολίας (περισσότερες σειρές από τούβλα και μεγαλύτερη ταχύτητα κίνησης).

B7.2.6 Προγραμματισμός με GML

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- ❖ Να προγραμματίζουμε τη συμπεριφορά των objects
- ❖ Να δημιουργούμε scripts, χρησιμοποιώντας τη GameMaker Language (GML)
- ❖ Να εισάγουμε τα scripts σε objects, ώστε να επηρεάζουμε τη λογική του παιχνιδιού.

1. Εισαγωγή

Στα προηγούμενα κεφάλαια είχαμε προγραμματίσει ενέργειες (actions) για διάφορα συμβάντα χρησιμοποιώντας εικονίδια τα οποία επιλέγαμε από τις καρτέλες. Αυτός ο τρόπος προγραμματισμού είναι εύκολος για αρχάριους, αλλά σύντομα φαίνεται περιοριστικός, καθώς απαιτείται χρόνος για εντοπισμό, επιλογή και ρύθμιση των παραμέτρων για το κάθε εικονίδιο και γενικά η τροποποίηση του «προγράμματος» με το ποντίκι. Ο προγραμματισμός με μία πιο παραδοσιακή προσέγγιση, γράφοντας δηλαδή εντολές με το πληκτρολόγιο είναι ταχύτερος και αποτελεσματικότερος για έναν «έμπειρο» προγραμματιστή.

Το GameMaker υποστηρίζει μία δική του γλώσσα προγραμματισμού, τη GameMaker Language (GML), η οποία μοιάζει πολύ στη σύνταξη και τον τρόπο λειτουργίας της με τη C (και τη C++).

2. Προγραμματίζοντας με τη GameMaker Language (GML)

Παρακάτω αναφέρονται τα βασικά χαρακτηριστικά της GML. Συγκεκριμένα, γίνεται μία σύντομη αναφορά στις μεταβλητές, τους τελεστές, τις εκφράσεις και παραστάσεις, τη δομή διακλάδωσης, δομές επανάληψης.

2.1 Μεταβλητές

Κάθε τμήμα προγράμματος (στη συνέχεια θα αναφέρεται ως script), αποτελεί μέρος κάποιου αντικειμένου, αφού συνδέεται με κάποιο συμβάν (event) του αντικειμένου αυτού. Γι' αυτό έχει αυτόματα πρόσβαση σε όλα τα χαρακτηριστικά του αντικειμένου ως μεταβλητές. Η θέση του (μεταβλητές **x** και **y**), η ταχύτητά του (μεταβλητές **hspeed** και **vspeed**) κ.λπ.

Μπορούμε επίσης να δηλώσουμε τοπικές μεταβλητές οι οποίες είναι προσωρινές και εξαφανίζονται μόλις τελειώσει η εκτέλεση του script ως ακολούθως:

```
var i, num, str;  
i = 0;  
num = 24.5;  
str = "Hello!";
```

Μπορούμε ακόμα να δηλώσουμε καθολικές μεταβλητές, οι οποίες μετά τη δήλωσή τους διατηρούνται στη μνήμη μέχρι τον τερματισμό του παιχνιδιού. Π.χ. για να δηλώσουμε μία καθολική μεταβλητή με το όνομα food και αρχική τιμή 5 γράφουμε:

```
global.food = 5
```

2.2 Τελεστές, εκφράσεις και παραστάσεις

Η GML υποστηρίζει τους συνήθεις τελεστές +, -, *, /, τις παρενθέσεις (), τον τελεστή εκχώρησης τιμής =, τους συγκριτικούς τελεστές >, <, >=, <=, ==, !=, αλλά και όλους τους άλλους τελεστές της γλώσσας προγραμματισμού C, %, ++, --, +=, -=, &&, || κ.λπ.

Οι τελεστές μαζί με μεταβλητές, συναρτήσεις κ.λπ. μπορούν να συνδυαστούν σε εκφράσεις, για παράδειγμα $(x+100)/h$, $(x \leq 200) \&\& (x \geq x_{\max})$. Μπορούν επίσης να σχηματίσουν ολοκληρωμένες παραστάσεις (εντολές). Αυτές πρέπει να τερματίζονται με το σύμβολο `;`, όπως και στη γλώσσα προγραμματισμού C. Για παράδειγμα:

```
var yy=0;
  yy = y - 100 *vspeed;
y = yy + 200;
x = x -10;
```

2.3 Δομές διακλάδωσης

Υποστηρίζονται οι δομές διακλάδωσης της γλώσσας προγραμματισμού C, π.χ. η εντολή **if**. Χρησιμοποιούνται άγκιστρα για ομαδοποίηση των εντολών όπου χρειάζονται και το **else** είναι προαιρετικό. Για παράδειγμα:

```
if (x <=5)
{
  x = hspeed;
  hspeed = 0;
}
else
{
  x = x + hspeed;
  draw_text(x-16,y,string(score));
};
```

Στο προηγούμενο παράδειγμα φαίνεται και η χρήση της έτοιμης συνάρτησης, **draw_text**, η οποία είναι αντίστοιχη με το action **Draw Text** από την καρτέλα **Draw** του γραφικού τρόπου προγραμματισμού. Η GML υποστηρίζει με έτοιμες συναρτήσεις όλες τις ενέργειες (actions) του γραφικού προγραμματισμού που δεν μπορούν να διατυπωθούν ως εντολές (π.χ. **hspeed = 0**), καθώς και πολλές άλλες λειτουργίες.

2.4 Δομές επανάληψης

Υποστηρίζονται οι δομές επανάληψης της γλώσσας προγραμματισμού C, καθώς και μία επιπρόσθετη, η **repeat**, η οποία επαναλαμβάνεται για καθορισμένο αριθμό. Η δομή **do-while()** της C έχει αντικατασταθεί από τη δομή **do-until()**. Παρακάτω δίνονται παραδείγματα των δομών αυτών:

```
while(x <=5)
  x = x + hspeed;
```

```
for (i=0; i < 5; i++)
  instance_create(x+i*10, y, obj_blotch);
```

```
repeat (5)
  instance_create(random(200), random(200), obj_dot);
```



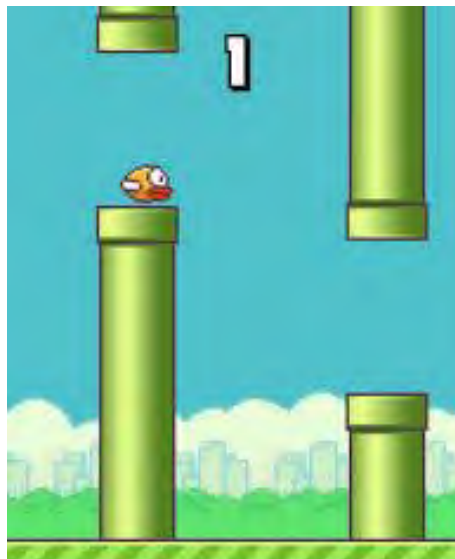
```
do {
  xx = xx + hspeed;
  yy = yy + 10;
  instance_create(xx, yy, obj_splat);
} until (xx > 400);
```



Ασκήσεις Κεφαλαίου

Άσκηση 6.1 Το παιχνίδι Flappy Bird

Να δημιουργήσετε το παιχνίδι Flappy Bird. Το Flappy Bird αναπτύχθηκε το 2013 για φορητές συσκευές από τον, Βιετναμικής καταγωγής, Dong Nguyen και διανεμήθηκε από την εταιρεία. GEARS Studios. Το παιχνίδι έχει απλά γραφικά και ο παίκτης ελέγχει ένα πουλάκι, το οποίο προσπαθεί να περάσει ανάμεσα από σωλήνες. Για κάθε επιτυχές πέρασμα λαμβάνει 1 βαθμό. Μόλις χτυπήσει σε σωλήνα ή στο έδαφος, το παιχνίδι τερματίζεται. Παρόλη την απλή πλοκή και τα απλά γραφικά του, είχε αναπάντεχη παγκόσμια επιτυχία καθώς ήταν εθιστικό και δύσκολο.



(1) Να δημιουργήσετε τα πιο κάτω sprites μαζί με τα αντίστοιχα objects:.

spr_bird (obj_bird): Ο κεντρικός σας χαρακτήρας

spr_pipe_top (obj_pipe_top): Το πάνω μέρος του σωλήνα

spr_pipe_bottom (obj_pipe_bottom): Το κάτω μέρος του σωλήνα

spr_grass (obj_grass): Το γρασίδι στο κάτω μέρος της οθόνης

spr_point (obj_point) με Visible false: αόρατο αντικείμενο ανάμεσα από τους σωλήνες, ώστε, όταν ο χαρακτήρας «συγκρούεται» (collision) με αυτό, να παίρνει έναν βαθμό και να συνεχίζει.

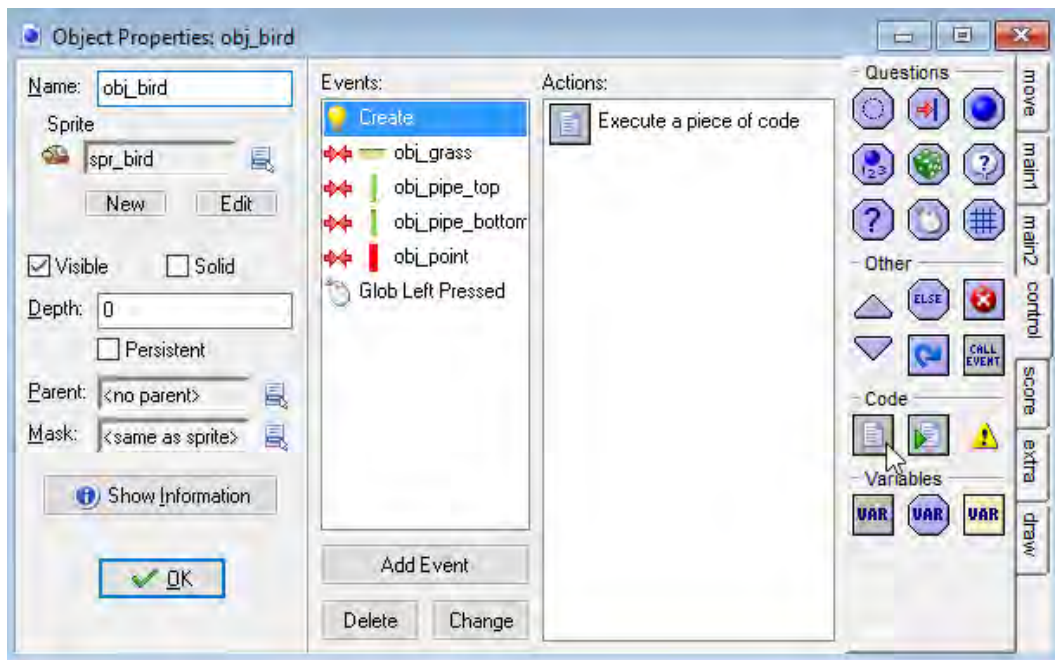
(2) Να προσθέσετε ως background την εικόνα bg_city.png.

- (3) Να προσθέσετε **font**, Terminal, Size 30, ώστε να εμφανίζεται το σκορ με μεγάλους χαρακτήρες στο πάνω μέρος της οθόνης.
- (4) Να εισαγάγετε ένα room και να βάλετε το background, το obj_bird πάνω από τα σύννεφα και το obj_grass στο κάτω μέρος του δωματίου.

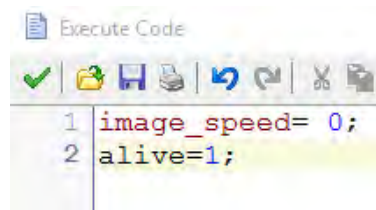
Άσκηση 6.2 Το παιχνίδι Flappy Bird

Σε αυτή την άσκηση θα προγραμματίσουμε events για το πουλάκι, obj_bird, χρησιμοποιώντας την GML για τα actions.

- (1) Να προσθέσετε το event **Create** στο obj_bird. Στα Actions να προσθέσετε από την καρτέλα Control το Execute Code.



- (2) Στο παράθυρο του κώδικα που θα ανοίξει να προσθέσετε τις πιο κάτω εντολές:



Το `image_speed=0` δίνει αρχικά μηδενική ταχύτητα στον χαρακτήρα σας. Η λογική του παιχνιδιού είναι ότι ο χαρακτήρας σας θα αρχίσει να κινείται, μόλις πατήσουμε το αριστερό κλικ του ποντικιού. Παρακάτω θα δούμε πώς να μετακινούμε το background αριστερά, ώστε να δίδεται στον χρήστη η ψευδαίσθηση της κίνησης του χαρακτήρα προς τα δεξιά.

Το `alive=1` είναι μία δική σας μεταβλητή, που δηλώνει, προφανώς, ότι ο χαρακτήρας σας δεν έχασε και έχει μία ζωή.

- (3) Να προσθέσετε το event **Collision**, ώστε όταν έχουμε σύγκρουση (collision) με το γρασίδι, τότε να εμφανίζεται ο πίνακας των ψηλότερων σκορ, να μηδενίζεται το σκορ και να ξαναρχίζει το παιχνίδι. Αυτά θα γίνουν με την εκτέλεση του κώδικα:

```
highscore_show(score);
score=0;
game_restart();
```

- (4) Να προσθέσετε events **Collision**, ώστε όταν έχουμε σύγκρουση με το πάνω ή με το κάτω μέρος του σωλήνα, να χάνουμε. Να εισαγάγετε τον κώδικα:

```
alive=0;
```

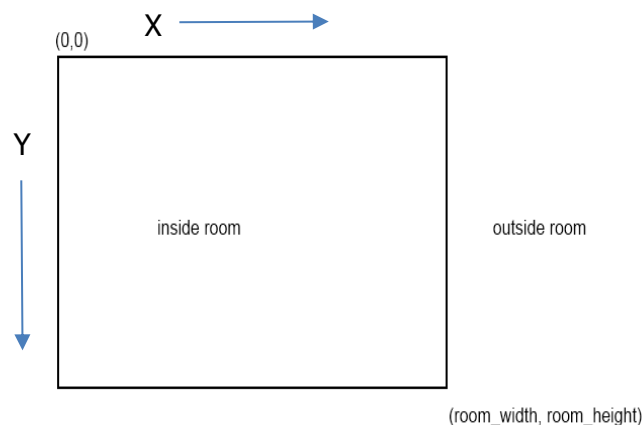
- (5) Να προσθέσετε το event **Collision**, ώστε όταν συγκρούεται με το obj_point, το οποίο είναι το αόρατο αντικείμενο που θα μπαίνει ανάμεσα από τους σωλήνες, θα ελέγχει αν το πουλάκι είναι ζωντανό. Στην περίπτωση που είναι, θα καταστρέφει το obj_point και θα προσθέτει έναν βαθμό. Το **with (other)** εφαρμόζει την ενέργεια στο άλλο αντικείμενο, που είναι το obj_point (θυμηθείτε ότι προγραμματίζουμε το obj_bird).

```
if (alive==1)
{
    with (other)
    {
        instance_destroy();
    }
    score +=1;
}
```

- (6) Να προσθέσετε το event **Global left button** (Event Mouse→Global Mouse→Global left button), ώστε κάθε φορά που πατούμε το ποντίκι το πουλάκι να παίρνει κάποια ταχύτητα προς τα πάνω, χωρίς την οποία σταδιακά πέφτει. Να προσθέσετε τον πιο κάτω κώδικα:

```
gravity=2;
if (alive==1)
{
    vspeed=-20;
}
```

Εδώ χρειάζεται κάποια εξήγηση. Οι άξονες που ισχύουν σε ένα δωμάτιο παιχνιδιού είναι:



Για να συνειδητοποιήσουμε την έννοια της κίνησης, πρέπει να μελετήσουμε τι συμβαίνει με τους άξονες του παιχνιδιού. Πάνω αριστερά βρίσκεται το σημείο (x,y) = (0,0). Για να κινηθεί προς τα κάτω ένα αντικείμενο, θα πρέπει να αυξήσουμε το y. Για να κινηθεί αριστερά, θα πρέπει να μειώσουμε το x. Αυτός είναι ο λόγος που θέτουμε με αρνητικό πρόσημο την κάθετη ταχύτητα, vertical speed: `vspeed=-20`

Το **gravity** είναι μία ενσωματωμένη μεταβλητή του GameMaker, η οποία προσθέτει βαρύτητα στον χαρακτήρα (επιτάχυνση προς τα κάτω).

- (7) Να δοκιμάσετε το παιχνίδι σας και να ελέγξετε την κίνηση που κάνει το πουλάκι. Στη συνέχεια να αποθηκεύσετε το παιχνίδι με το όνομα `flappybird1.gmk`.

Άσκηση 6.3 Το παιχνίδι Flappy Bird

Σε αυτή την άσκηση θα προγραμματίσουμε events για τα υπόλοιπα αντικείμενα. Η γενική προσέγγιση σε αυτό το παιχνίδι είναι ότι όλα τα αντικείμενα κινούνται αριστερά, ώστε να φαίνεται ότι το πουλάκι κινείται δεξιά (ενώ στην πραγματικότητα παραμένει στην ίδια οριζόντια θέση).

- (1) Να προσθέσετε το event **Create** στο `obj_grass`, και ως action τον πιο κάτω κώδικα, ώστε με τη δημιουργία του το γρασίδι να κινείται αριστερά με σταθερή ταχύτητα.

```
hspeed=-6
```

- (2) Να προσθέσετε το event **End Step** (Add Event → Step → End Step) στο `obj_grass` και ως action τον πιο κάτω κώδικα, ώστε στο τέλος του κάθε βήματος που εκτελείται να ελέγχει αν το `x < -24` (δηλαδή εκτός του δωματίου) και να το μετακινεί στο τέλος του δωματίου δεξιά ώστε να ξαναεμφανίζεται.

```
if (x<-24) {
    x+= room_width + 24;
}
```

- (3) Να προσθέσετε το event **Create** στο `obj_pipe_top` και στο `obj_pipe_bottom` και ως action τον πιο κάτω κώδικα, ώστε με τη δημιουργία τους να κινούνται αριστερά με σταθερή ταχύτητα.

```
hspeed=-6
```

- (4) Να προσθέσετε το event **End Step** (Add Event → Step → End Step) στο `obj_pipe_top` και στο `obj_pipe_bottom` και ως action τον πιο κάτω κώδικα, ώστε στο τέλος του κάθε βήματος που εκτελείται να ελέγχει αν ο σωλήνας έχει βγει εκτός του δωματίου (`x < -100`), να τον καταστρέφει και αν ο χαρακτήρας σας έχει χάσει (`alive==0`) να τον σταματάει από το να κινείται:

```
if (x < -100) instance_destroy();
if (obj_bird.alive == 0) hspeed=0;
```

- (5) Να προσθέσετε το event **Create** στο `obj_point` και ως action τον πιο κάτω κώδικα, ώστε με τη δημιουργία του να κινείται αριστερά με σταθερή ταχύτητα (δεν χρειάζεται να προγραμματίσουμε κάτι άλλο για το αντικείμενο αυτό):

```
hspeed=-6
```

Αν εκτελέσετε το παιχνίδι, θα διαπιστώσετε ότι δεν υπάρχουν σωλήνες, αφού δεν τοποθετήσαμε κάποιον στην πίστα. Αυτοί θα τοποθετούνται από τον ελεγκτή του παιχνιδιού, τον οποίο θα δημιουργήσουμε στη συνέχεια.

(6) Να αποθηκεύσετε το παιχνίδι με το όνομα flappybird2.gmk.

Άσκηση 6.4 Το παιχνίδι Flappy Bird

Σε αυτή την άσκηση θα προγραμματίσουμε τον ελεγκτή του παιχνιδιού (obj_controller), ώστε να δημιουργεί τους σωλήνες κατά διαστήματα, σε τυχαία θέση και να δείχνει το σκορ.

(1) Για τον obj_controller, να δημιουργήσετε το event Create και να προσθέσετε Action→Execute Code:

```
alarm[0]=60;
```

Το alarm[0] είναι ένα από δώδεκα «ξυπνητήρια» (0 μέχρι και 11), που κάνει αντίστροφη μέτρηση σε βήματα παιχνιδιού (game steps). Ένα δευτερόλεπτο είναι συνήθως 30 game steps, οπότε ουσιαστικά καθορίζουμε την αντίστροφη μέτρηση στα 2 δευτερόλεπτα. Τα βήματα ανά δευτερόλεπτο μπορείτε να τα αλλάξετε, αν επιθυμείτε, από το rooms→settings→speed.

(2) Να προσθέσετε το αντίστοιχο event για το alarm (Add Event→Alarm→Alarm 0) και να προσθέσετε Action→Execute Code:

```
alarm[0]=60;
var position, spread;
position= random (300)+150;
spread=150;
instance_create (room_width,position - spread/2,obj_pipe_top);
instance_create (room_width,position + spread/2,obj_pipe_bottom);
instance_create (room_width,position,obj_point);
```

Εδώ χρησιμοποιούνται δύο μεταβλητές, για τη θέση (**position**) και για την απόσταση (**spread**) μεταξύ του πάνω και του κάτω μέρους του σωλήνα. Με βάση αυτές τις μεταβλητές, δημιουργούνται κάθε 2 δευτερόλεπτα (60 game steps), σε τυχαία θέση, το πάνω μέρος του σωλήνα, το κάτω μέρος του και το αντικείμενο βαθμών ανάμεσά τους στη δεξιά άκρη της πίστας (**room_width**).

(3) Να προσθέσετε το Event **Draw** και στα **Actions** να εισαγάγετε από την καρτέλα **Draw** το **set font**, επιλέγοντας το **font0** που δημιουργήσατε στην αρχή του παιχνιδιού. Στη συνέχεια, να εισαγάγετε το **Execute Code**, ώστε να εμφανίζεται το σκορ στη μέση του πάνω μέρους της οθόνης του παιχνιδιού:

```
draw_text(room_width/2,20,string(score));
```

(4) Το παιχνίδι σας είναι έτοιμο! Να το αποθηκεύσετε με το όνομα flappybird3.gmk. Να το δοκιμάσετε και να παίξετε ελεύθερα!



+ Ασκήσεις Εμπλουτισμού**Άσκηση 6.5** Το παιχνίδι Flappy Bird

Στο παιχνίδι Flappy Bird:

- (1) Να προσθέσετε μουσική και ηχητικά εφέ.
- (2) Να αλλάξετε τον τρόπο βαθμολόγησης, έτσι ώστε να αυξάνεται κατά 10 βαθμούς.
- (3) Να μειώσετε την ταχύτητα που κινούνται οι σωλήνες.
- (4) Να αυξήσετε την απόσταση μεταξύ των σωλήνων.
- (5) Να τροποποιήσετε το παιχνίδι σας ώστε αρχικά οι σωλήνες να είναι αραιοί και να κινούνται αργά και σταδιακά να γίνονται πιο πυκνοί και να κινούνται πιο γρήγορα.

B7.2.7 Η διαδικασία Ανάπτυξης Ενός Ηλεκτρονικού Παιγνιδιού

Τι θα μάθουμε σε αυτό το κεφάλαιο:

- Να δημιουργούμε ένα ολοκληρωμένο παιχνίδι
- Τη διαδικασία ανάπτυξης ενός ηλεκτρονικού παιχνιδιού (ανάλυση-πλοκή, σχεδιασμός, υλοποίηση, έλεγχος).

1. Η διαδικασία ανάπτυξης ενός ηλεκτρονικού παιχνιδιού

Έχοντας μάθει πώς να χρησιμοποιούμε αποτελεσματικά ένα εργαλείο δημιουργίας παιχνιδιών όπως το GameMaker, αποτελεί μόνο μέρος των ικανοτήτων που θα πρέπει να κατέχει κάποιος για να μπορεί να δημιουργεί επιτυχημένα παιχνίδια. Όπως μάθαμε και στο πρώτο κεφάλαιο, η διαδικασία δημιουργίας ενός παιχνιδιού είναι πολύπλοκη, χρονοβόρα και δύσκολη.

Σε γενικές όμως γραμμές αποτελείται από τέσσερις φάσεις, την ανάλυση, τον σχεδιασμό, την υλοποίηση και τον έλεγχο.

Στη φάση της **ανάλυσης** είναι σημαντικό να καθοριστεί η πλοκή του παιχνιδιού και η ροή της κατά τη διάρκεια του παιχνιδιού, καθώς και η μορφή και η γενική συμπεριφορά των χαρακτήρων και των άλλων αντικειμένων του παιχνιδιού. Θα πρέπει να αποφασιστούν:

- Ποιο είναι το κοινό στο οποίο απευθύνεται;
- Ποια θα είναι η κατηγορία του παιχνιδιού;
- Ποια θα μπορούσε να είναι η πλοκή του παιχνιδιού και ποια πρωτοτυπία έχει το παιχνίδι;
- Πώς μπορεί η πλοκή να γίνει ενδιαφέρουσα και διασκεδαστική;
- Σε ποια επιμέρους στάδια ή επίπεδα θα διαχωρίζεται η ανάπτυξη της πλοκής;

Η φάση του **σχεδιασμού** περιλαμβάνει τόσο το δημιουργικό σχεδιασμό όσο και τον τεχνικό σχεδιασμό. Ο **δημιουργικός σχεδιασμός** περιλαμβάνει την επιλογή και τη δημιουργία των διαφόρων απεικονίσεων των χαρακτήρων και ήχων για το παιχνίδι ώστε να υπάρχει καλαισθησία και ομοιογένεια στο παιχνίδι. Ο τεχνικός σχεδιασμός περιλαμβάνει τα αντικείμενα του παιχνιδιού και πώς συμπεριφέρονται, τον αριθμό και τα είδη που θα υπάρχουν σε κάθε πίστα ή επίπεδο, τους επιμέρους στόχους του κάθε επιπέδου, το σύστημα απονομής βαθμών και απώλειας πόρων (π.χ. ζωών) και άλλα τεχνικά χαρακτηριστικά του παιχνιδιού.

Στη φάση της **υλοποίησης** το παιχνίδι δημιουργείται με το εργαλείο δημιουργίας παιχνιδιών (στην περίπτωση μας, το GameMaker). Είναι πιθανό να εντοπιστούν ζητήματα σε σχέση με την πλοκή και τη λειτουργία του παιχνιδιού που δεν είχαν προβλεφθεί κατά το σχεδιασμό και να χρειαστεί να γίνουν αλλαγές. Για παράδειγμα, ένα επίπεδο μπορεί αν μην είναι δυνατό να ολοκληρωθεί διότι ο αριθμός των δυσκολιών για τον παίκτη είναι υπερβολικός, ή επειδή ο στόχος δεν είναι ξεκάθαρος.

Τέλος, η φάση του **ελέγχου**, επικεντρώνεται στον εντοπισμό τυχόν προβλημάτων που πιθανό να συναντήσει ο παίκτης, για παράδειγμα κάποια κίνηση του παίκτη που οδηγεί σε απρόβλεπτη κατάσταση, ατέλειες στον τρόπο λειτουργίας του παιχνιδιού, κ.λπ.

**Ασκήσεις Κεφαλαίου****Άσκηση 7.1** Εργασία - Project

Στόχος σας είναι να δημιουργήσετε ένα παιχνίδι της δική σας επιλογής και θεματολογίας.

Συστήνεται να εργαστείτε ομαδικά και στο τέλος η κάθε ομάδα να παρουσιάσει το παιχνίδι της.

**Κριτήρια και Ποσοστά Αξιολόγησης του παιχνιδιού**

Τα κριτήρια αξιολόγησης του παιχνιδιού και η βαρύτητα του καθενός έχουν ως ακολούθως:

- (1) Διαδικασία επιλογής και σχεδιασμού πλοκής – βαρύτητα 15%
- (2) Εκπαιδευτικός χαρακτήρας του παιχνιδιού – βαρύτητα 5%
- (3) Σχεδιασμός κόσμου, αντικειμένων και μέσων του παιχνιδιού – βαρύτητα 10%
- (4) Σχεδιασμός βέλτιστου μηχανισμού χειρισμού–ελέγχου του παιχνιδιού από τον χρήστη – βαρύτητα 20%
- (5) Ύπαρξη πολλαπλών επιπέδων δυσκολίας με σταδιακή κλιμάκωση – βαρύτητα 20%
- (6) Κλιμακωτό σύστημα βαθμολογίας παίκτη – βαρύτητα 10%
- (7) Δυναμική συμπεριφορά χαρακτήρων που ελέγχονται από τον υπολογιστή – βαρύτητα 10%
- (8) Τεχνική αρτιότητα παιχνιδιού – βαρύτητα 10%.



ΠΑΡΑΡΤΗΜΑ Α. Λογικά Διαγράμματα και Προκαταρτική Εκτέλεση

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΠΟΛΙΤΙΣΜΟΥ

ΠΛΗΡΟΦΟΡΙΚΗ

ΣΗΜΕΙΩΣΕΙΣ

Παράρτημα

Σχεδίαση Λογικών Διαγραμμάτων

Προκαταρκτική Εκτέλεση Λογικού Διαγράμματος και Προγράμματος

Σεπτέμβριος 2017

ΠΕΡΙΕΧΟΜΕΝΑ

1. Πρόλογος.....	3
2. Λογικά Διαγράμματα	4
2.1 Δομημένος Προγραμματισμός	4
2.2 Σύμβολα Λογικών Διαγραμμάτων.....	5
2.3 Εισηγήσεις για ομοιόμορφο σχεδιασμό λογικών διαγραμμάτων.....	6
2.4 Δομές Προγραμματισμού.....	8
2.4.1 Ακολουθιακή Δομή.....	8
2.4.2 Δομή Διακλάδωσης.....	9
2.4.3 Επαναληπτική Δομή	10
3. Προκαταρκτική Εκτέλεση	14
3.1 Δομές Προγραμματισμού.....	15
3.1.1 Ακολουθιακή Δομή.....	15
3.1.2 Δομή Διακλάδωσης.....	16
3.1.3 Επαναληπτική Δομή	18
4. Βιβλιογραφία.....	29

Συνάδελφοι,

Οι σημειώσεις αυτές έχουν γραφτεί για να καλύψουν τις ανάγκες του μαθήματος της Πληροφορικής σε Γυμνάσιο και Λύκειο. Έχουν ως στόχο να μας βοηθήσουν να χρησιμοποιήσουμε όλοι ομοιόμορφες μεθόδους στον σχεδιασμό λογικών διαγραμμάτων, αλλά και στην προκαταρκτική εκτέλεση των προγραμμάτων ή και των λογικών διαγραμμάτων. Οι σημειώσεις αποτελούνται από δύο ενότητες με παραδείγματα.

Η πρώτη ενότητα αφορά στον σχεδιασμό των λογικών διαγραμμάτων, διαδικασία αναγκαία στην ανάπτυξη των προγραμμάτων. Τα σύμβολα που χρησιμοποιήθηκαν ακολουθούν το πρότυπο ANSI - Flowchart Symbols.

Η δεύτερη ενότητα αφορά στην προκαταρκτική εκτέλεση προγραμμάτων ή/και λογικών διαγραμμάτων.

Κατά την προκαταρκτική εκτέλεση θα πρέπει να δίνεται ιδιαίτερη έμφαση στην πιστή εκτέλεση όλων των εντολών του προγράμματος ή και του λογικού διαγράμματος. Θα πρέπει να καταγράφονται αναλυτικά όλες οι σταθερές, οι μεταβλητές, οι συνθήκες και οι αντίστοιχες τιμές τους, ώστε να βοηθηθούν οι μαθητές στην κατανόηση της ροής και των αλλαγών των δεδομένων στο πρόγραμμα ή και στο λογικό διάγραμμα.

Στην προσπάθειά μας να συμβαδίσουμε με τα υφιστάμενα εγχειρίδια του Γυμνασίου και του Λυκείου, χρησιμοποιήσαμε μεθοδολογία από διάφορες έγκυρες πηγές (βλέπε βιβλιογραφία), χωρίς να χάνεται η συνοχή και η ομοιομορφία. Βέβαια, δεν αποκλείονται άλλες προσεγγίσεις παρουσίασης των λογικών διαγραμμάτων και προκαταρκτικών εκτελέσεων, νοουμένου ότι είναι επιστημονικά αποδεκτές.

Συγγραφική ομάδα:

Μιλτιάδου Μάριος, Β.Δ.

Σχίζα Ιουλία, Β.Δ.

Σωτικόπουλος Κωνσταντίνος, Β.Δ.

Τορτούρης Μιχάλης, Β.Δ.

Εποπτεία:

Χατζησάββας Γιώργος, ΕΜΕ

Αναθεώρηση για την C++:

Άκης Συκοπετρίτης, Καθηγητής Πληροφορικής

Κωνσταντίνος Σωφρονίου, Β.Δ. και Σύμβουλος Πληροφορικής

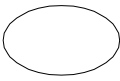


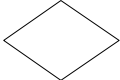


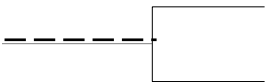
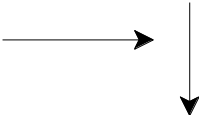
Δομημένος Προγραμματισμός

Ο δομημένος προγραμματισμός είναι η τεχνική που διαιρεί το πρόγραμμα σε λογικά τμήματα, χρησιμοποιώντας βασικές δομές. Η τεχνική αυτή μάς βοηθά στην κωδικοποίηση, στον έλεγχο και στη συντήρηση ενός προγράμματος. Στην παραγωγή λογισμικού, ο δομημένος προγραμματισμός χρησιμοποιείται ως βασική αρχή προγραμματισμού. Βασικός στόχος του δομημένου προγραμματισμού είναι η παραγωγή προγραμμάτων που έχουν μια καθορισμένη μορφή και για αυτό γίνονται πιο εύκολα κατανοητά, όχι μόνο από τον δημιουργό τους αλλά και από άλλους προγραμματιστές, οι οποίοι στο μέλλον θα ασχοληθούν με τα προγράμματα αυτά.

Ο δομημένος προγραμματισμός χρησιμοποιεί μόνο τρεις βασικές δομές ελέγχου, που είναι:

1. Η Ακολουθιακή Δομή (Sequence)
2. Η Δομή Διακλάδωσης (Selection) και η επέκτασή της, η Περιπτωσιακή Δομή (Switch)
3. Η Δομή Επανάληψης (Iteration)

Σύμβολα Λογικών Διαγραμμάτων

Σύμβολο	Επεξήγηση
	Αρχή - Τέλος Κυρίως Προγράμματος Είσοδος - Έξοδος Υποπρογραμμάτων
	Επεξεργασία
	Είσοδος Δεδομένων – Έξοδος Αποτελεσμάτων
	Απόφαση
	Υποπρόγραμμα (=Συνάρτηση)
	Σύνδεσμος
	Σχόλιο
	Ροή

ANSI Flowchart symbols (Capron, H. L., Perron, J. D., p. 520)

Εισηγήσεις για Ομοιόμορφο Σχεδιασμό Λογικών Διαγραμμάτων

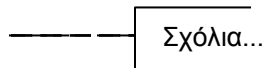
1. Το λογικό διάγραμμα πρέπει να αντιστοιχεί με το πρόγραμμα. Αντιστοιχία λογικού διαγράμματος και προγράμματος σημαίνει ότι:
 - i. Οι τεχνικές και οι δομές προγραμματισμού πρέπει να είναι οι ίδιες στο λογικό διάγραμμα και στο πρόγραμμα. Για παράδειγμα, αν χρησιμοποιείται η δομή if-else στο λογικό διάγραμμα, θα πρέπει να αναμένεται η ίδια δομή και στο πρόγραμμα.
 - ii. Οι μεταβλητές πρέπει να είναι οι ίδιες στο λογικό διάγραμμα και στο πρόγραμμα.
2. Τα μηνύματα εισόδου και εξόδου στο λογικό διάγραμμα δεν είναι υποχρεωτικά. Το πρόγραμμα, όμως, πρέπει να περιλαμβάνει τα κατάλληλα μηνύματα εισόδου και εξόδου (είτε το ζητά η άσκηση είτε όχι, στο πλαίσιο του καλού προγραμματισμού).
3. Οι μεταβλητές που χρησιμοποιούνται σε λογικό διάγραμμα πρέπει να γράφονται με λατινικούς χαρακτήρες. Οι οδηγίες μπορούν να είναι στα Ελληνικά (διάβασε, τύπωσε, κ.τ.λ.).
4. Στις αποφάσεις στο λογικό διάγραμμα μπορούν να χρησιμοποιηθούν τα εξής: Αληθής – Ψευδής ή A – Ψ ή Ναι – Όχι ή N – Ο ή True – False ή T – F ή Yes – No ή Y – N.
5. Ως σύμβολο εκχώρησης τιμής σε μεταβλητή συστήνεται όπως χρησιμοποιείται το βέλος (\leftarrow) για ομοιομορφία, χωρίς όμως να απαγορεύονται άλλα σύμβολα όπως το =.
6. Σε περίπτωση μαθηματικών πράξεων στο λογικό διάγραμμα, μπορούν να χρησιμοποιηθούν είτε μαθηματικά σύμβολα (όπως $\frac{a+b+c}{3}$, $\sqrt{a-b}$, x^2 , \leq , κ.τ.λ.), είτε σύμβολα που, συνήθως, χρησιμοποιούνται σε γλώσσες προγραμματισμού, όπως: * για πολλαπλασιασμό, / για διαίρεση, MOD, %, DIV, <=, >=, κ.τ.λ.
7. Σε περίπτωση εκτύπωσης (πραγματικών ή και ακεραίων αριθμών, χαρακτήρων και ακολουθίας χαρακτήρων) σε λογικό διάγραμμα, δεν χρειάζονται οι δείκτες πλάτους εκτύπωσης. Στο αντίστοιχο πρόγραμμα, όμως, συστήνεται όπως χρησιμοποιούνται στο πλαίσιο του καλού προγραμματισμού. Για παράδειγμα:

Pascal: `writeln(A:6:2, B:8, 'Onoma':8);`

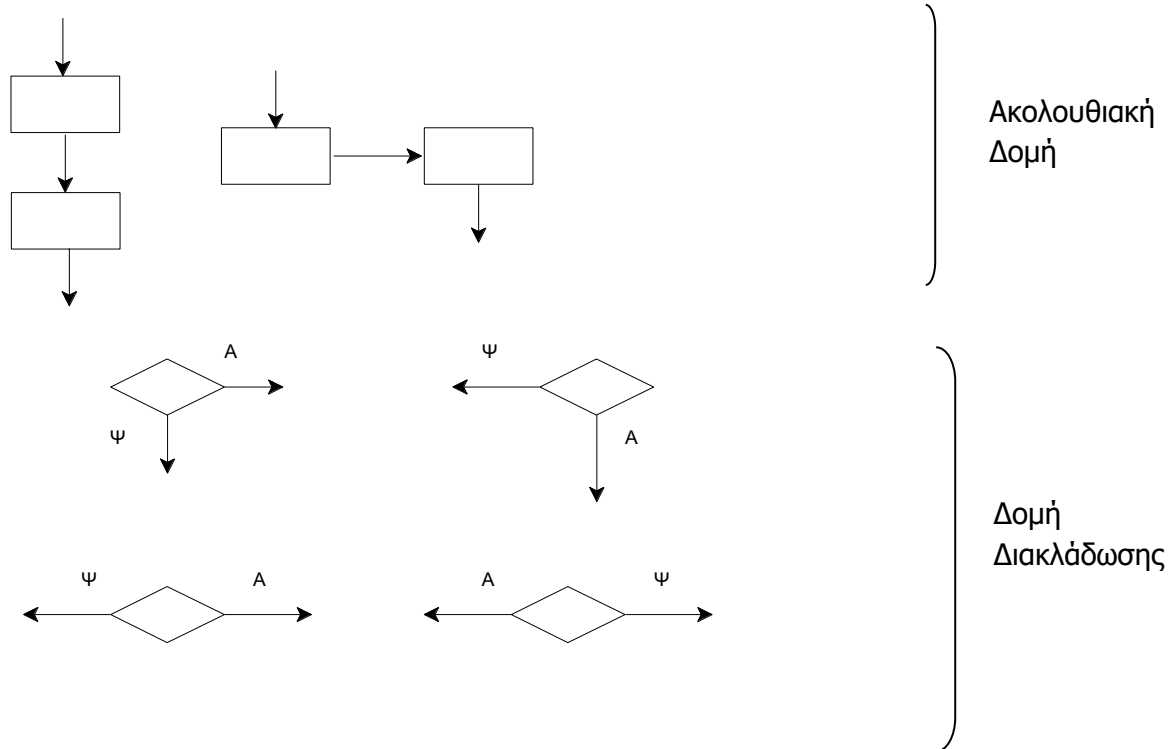
C++:

```
cout<<fixed<<setw(8)<<setprecision(2)<<A<<setw(8)<<B<<setw(8)<<"Onoma";
```

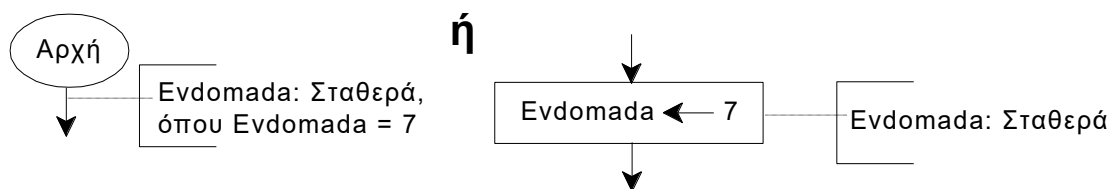
8. Στο λογικό διάγραμμα μπορούν να γραφτούν σχόλια, χρησιμοποιώντας το πιο κάτω σύμβολο:



9. Η ροή στο λογικό διάγραμμα υποδεικνύεται από τα βέλη. Τα πιο κάτω σχήματα είναι αποδεκτά:

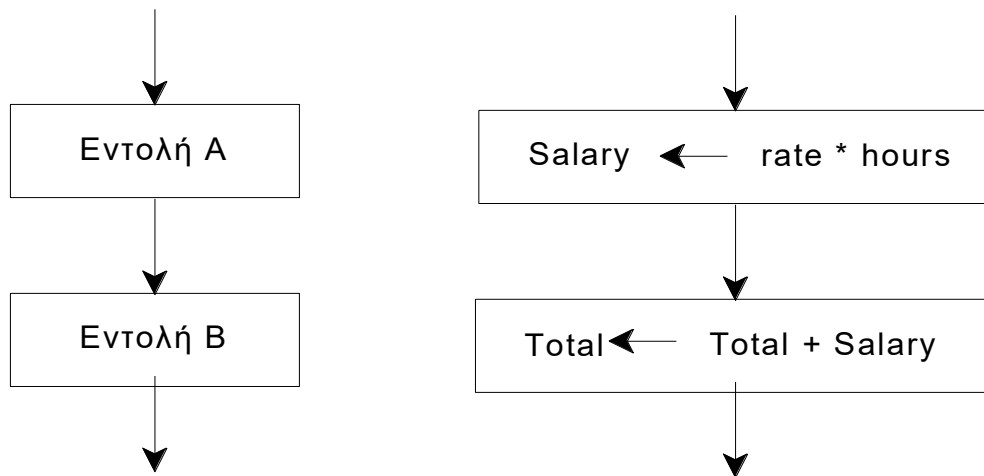


10. Οι σταθερές πρέπει να διαφοροποιούνται από τις μεταβλητές με τη χρήση σχολίου. Για παράδειγμα:



Δομές Προγραμματισμού

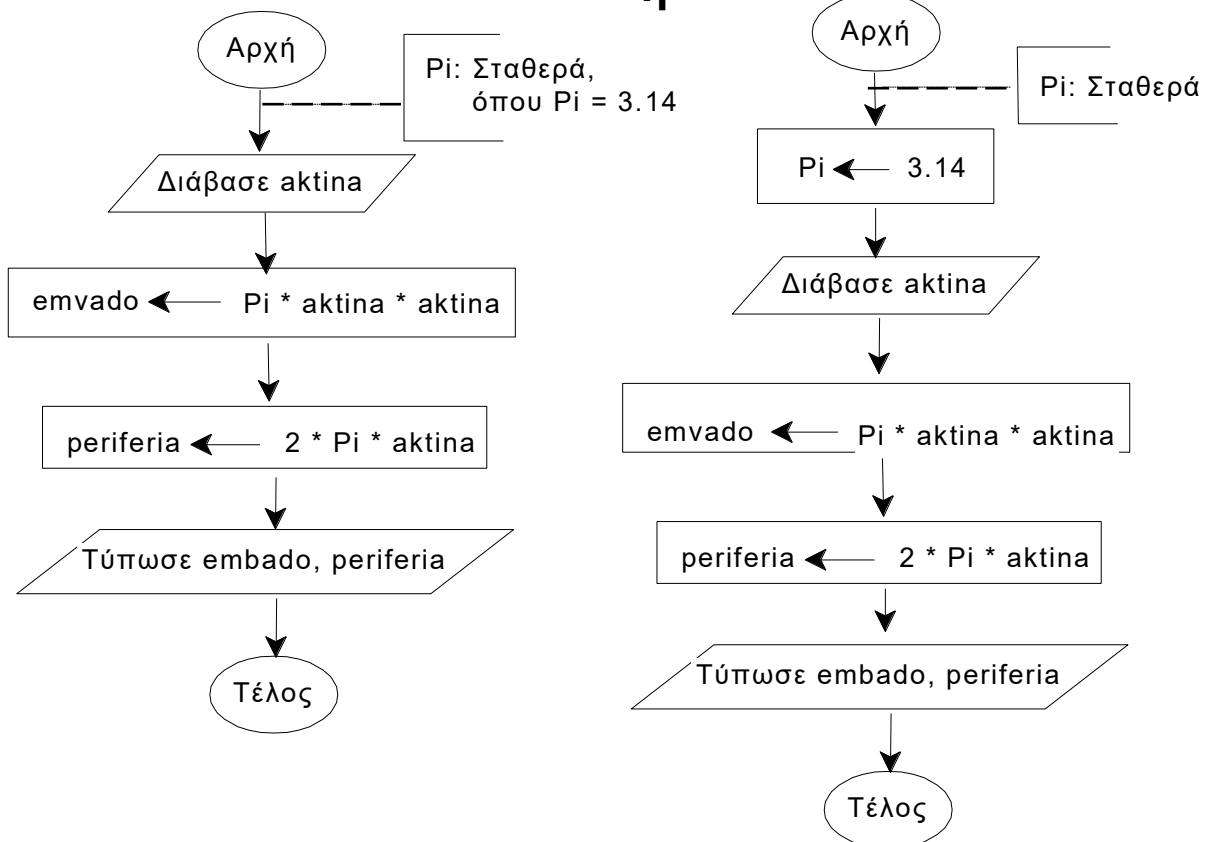
(α) Ακολουθιακή δομή



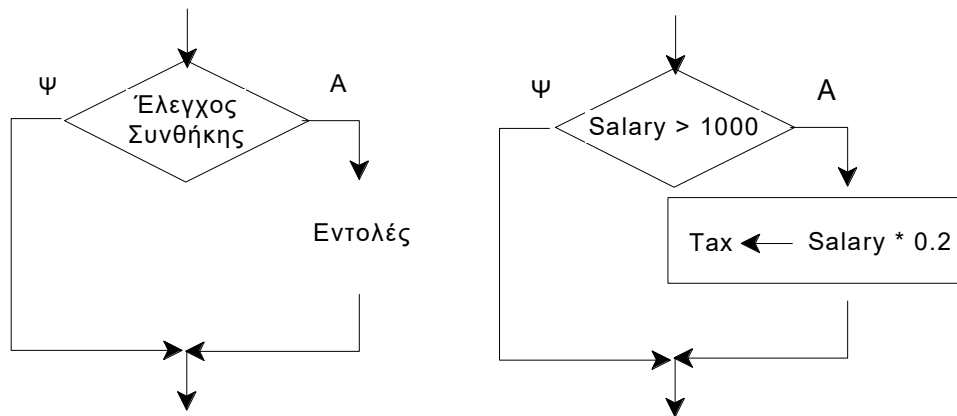
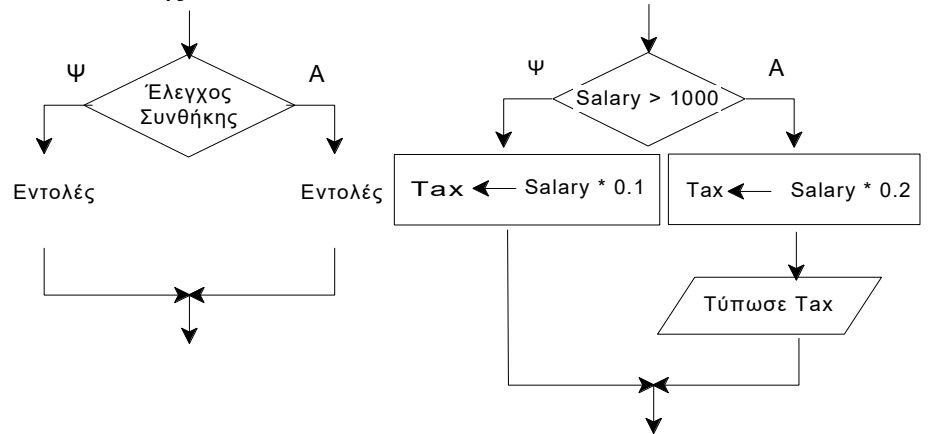
Παράδειγμα Ακολουθιακής Δομής

Να σχεδιάσετε λογικό διάγραμμα, το οποίο να ζητά την ακτίνα του κύκλου και να υπολογίζει και να παρουσιάζει το εμβαδόν και την περιφέρειά του.
(Σημείωση: Εμβαδόν Κύκλου = πr^2 , Περιφέρεια Κύκλου = $2\pi r$, $\pi = 3.14$)

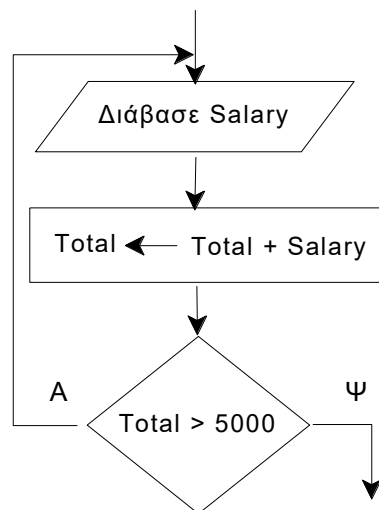
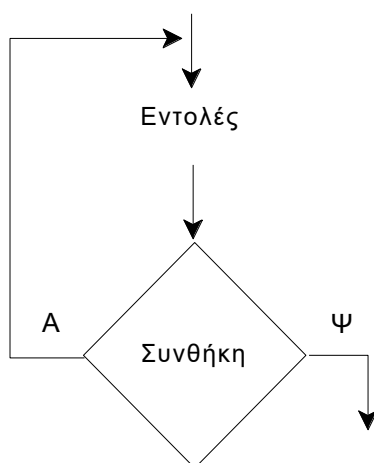
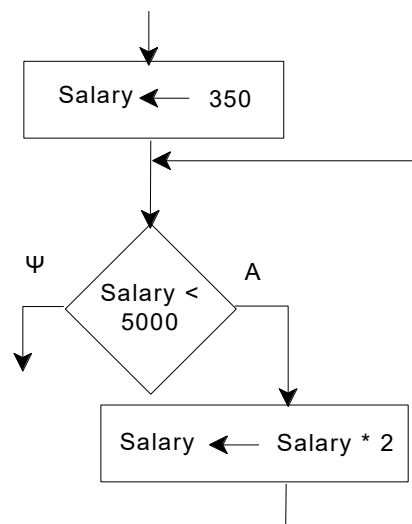
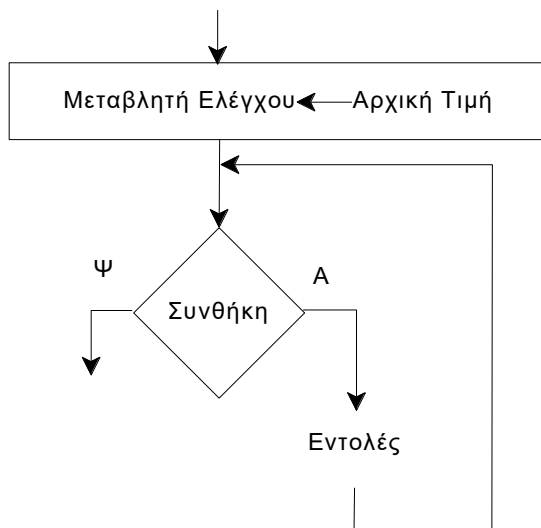
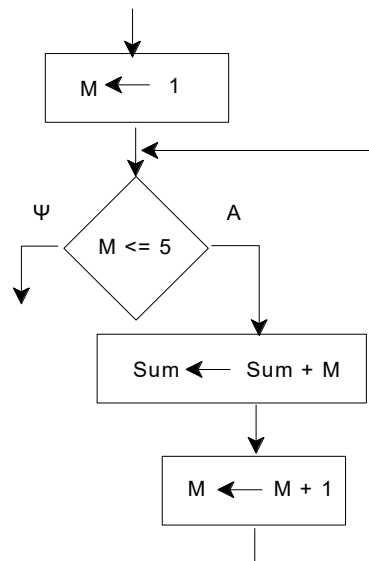
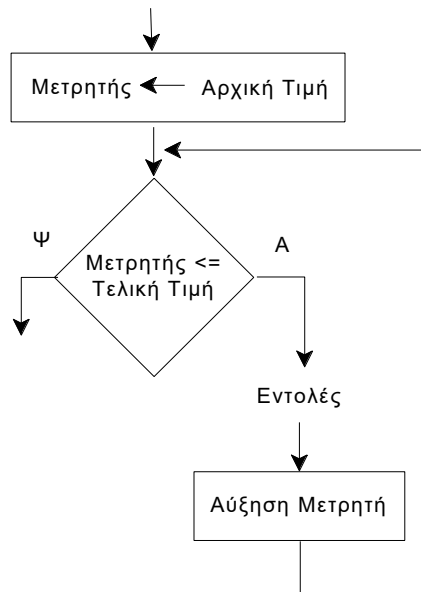
ή



(β) Δομή Διακλάδωσης

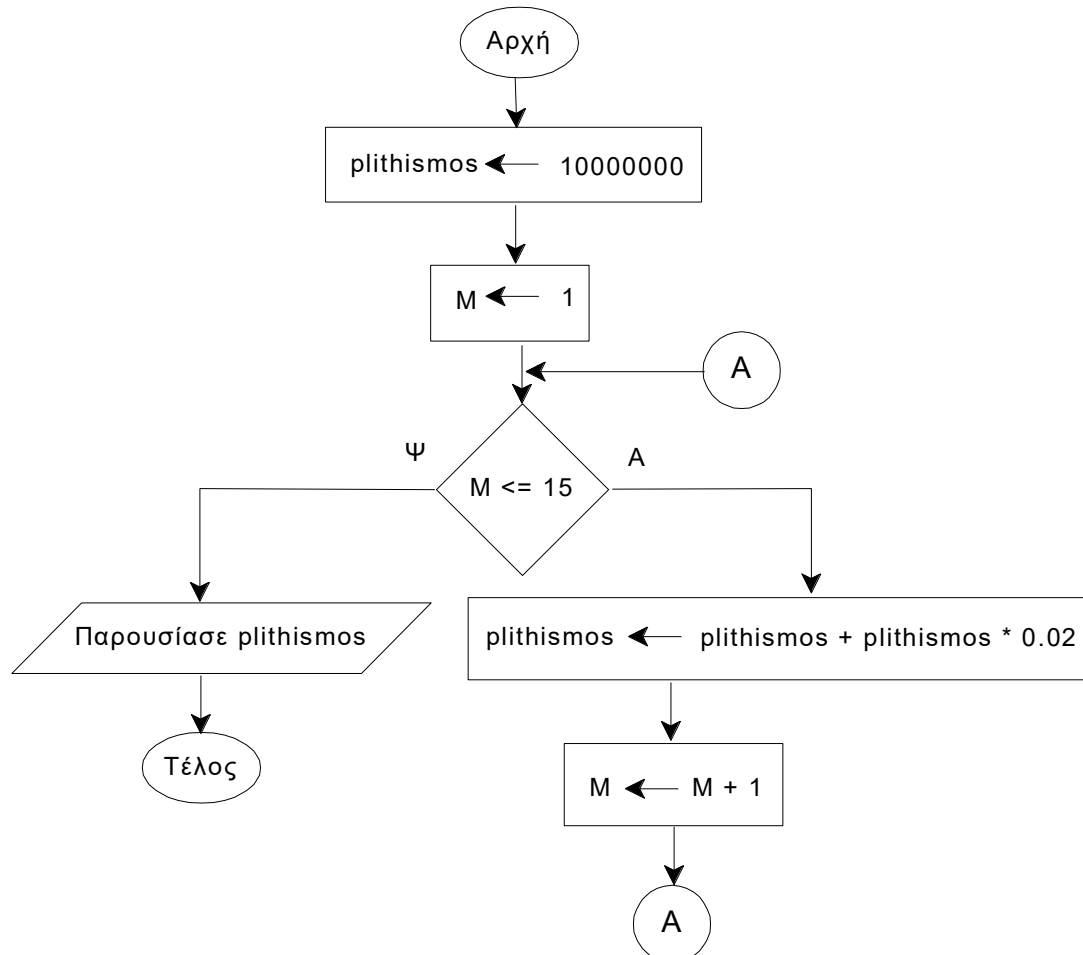


(γ) Επαναληπτική Δομή

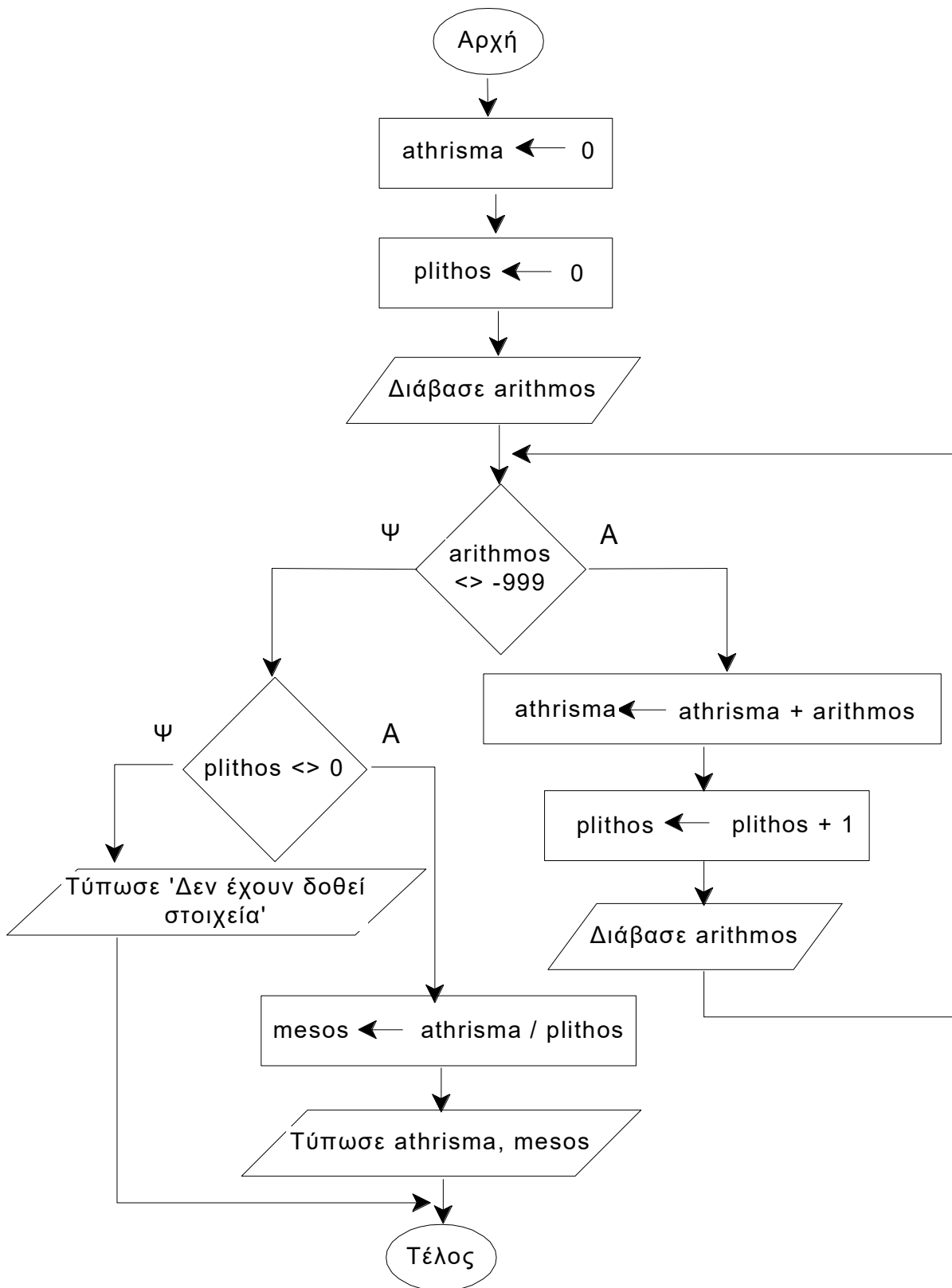


Παραδείγματα Επαναληπτικής Δομής

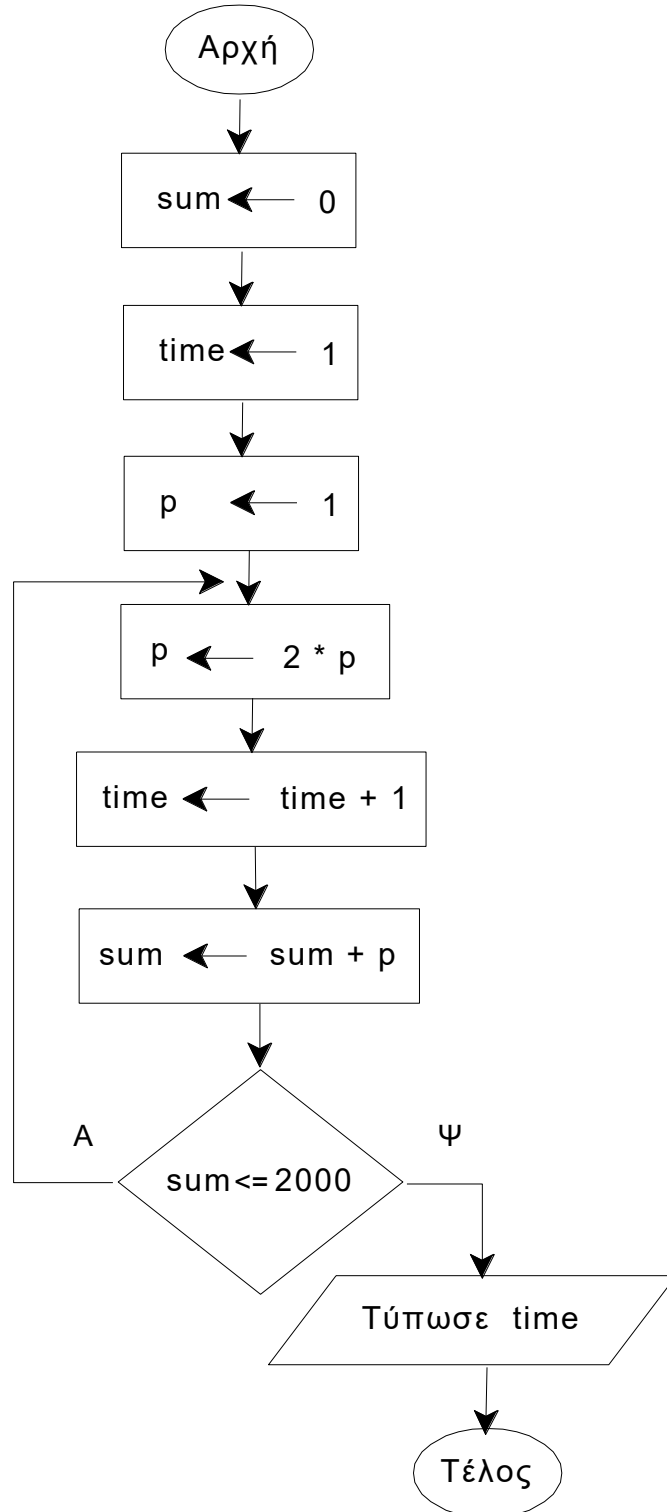
1. Ο πληθυσμός μιας χώρας είναι 10 εκατομμύρια και παρουσιάζει ετήσια αύξηση 2%. Να σχεδιάσετε λογικό διάγραμμα, το οποίο να υπολογίζει και να παρουσιάζει τον πληθυσμό της χώρας μετά από 15 χρόνια.



2. Να σχεδιάσετε λογικό διάγραμμα, το οποίο να δέχεται ένα πλήθος θετικών ακεραίων αριθμών και να υπολογίζει και να παρουσιάζει το άθροισμά τους και τον μέσο όρο τους. Η διαδικασία να τερματίζεται με την εισαγωγή αρνητικού αριθμού.



3. Ο πατέρας του Γιάννη δίνει στο παιδί του διπλάσιο ποσό από τον προηγούμενο μήνα. Να σχεδιάσετε λογικό διάγραμμα το οποίο να υπολογίζει και να παρουσιάζει το μικρότερο χρονικό διάστημα, κατά το οποίο ο Γιάννης θα πάρει ποσό που να υπερβαίνει τις 2000 ευρώ. Να υποθέσετε ότι ο Γιάννης πήρε την πρώτη φορά 1 ευρώ.



Προκαταρκτική Εκτέλεση

Κατά την προκαταρκτική εκτέλεση, ο μαθητής εκτελεί βήμα προς βήμα όλες τις εντολές του προγράμματος ή του λογικού διαγράμματος, όπως θα τις εκτελούσε ο ηλεκτρονικός υπολογιστής. Σε κάθε βήμα σημειώνει τις τιμές των μεταβλητών ή των αποφάσεων που αλλάζουν. Για τον σκοπό αυτό, χρησιμοποιείται, συνήθως, ένας πίνακας στον οποίο καταχωρούνται:

- i. Οι τιμές των σταθερών (εάν υπάρχουν).
- ii. Οι αρχικές τιμές των μεταβλητών, καθώς και οι τιμές τους μετά από κάθε αλλαγή.
- iii. Οι αποφάσεις και οι τιμές τους όταν αλλάζουν.
- iv. Οι δείκτες και το περιεχόμενο των πινάκων.
- v. Η παρουσίαση των μηνυμάτων και των αποτελεσμάτων του προγράμματος ή του λογικού διαγράμματος. Η παρουσίαση θεωρείται ανεξάρτητη από το κύριο πρόγραμμα ή τα υποπρογράμματα (συνάρτηση ή διαδικασία, αν υπάρχουν) και είναι ενιαία (δηλαδή, όπως θα εμφανιζόταν στην οθόνη του Η.Υ. μετά την εκτέλεση του προγράμματος).

Για σκοπούς εξοικονόμησης χώρου τα βήματα της προκαταρκτικής εκτέλεσης δεν είναι αναγκαίο να παρουσιάζονται οπωσδήποτε σε ξεχωριστές γραμμές (βλέπε παραδείγματα).

Δομές Προγραμματισμού(α) Ακολουθιακή Δομή

Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου: 5.5, 8.0

```
// Τρίγωνο
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
int main() {
    float katheti1, katheti2, ypoteinouse, perimetros, emvadon;
    cout << "Δώσε το μήκος της κάθετης 1: ";
    cin >> katheti1;
    cout << "Δώσε το μήκος της κάθετης 2: ";
    cin >> katheti2;
    ypoteinouse = sqrt( pow( katheti1, 2.0 ) + pow( katheti2, 2.0 ) );
    perimetros = katheti1 + katheti2 + ypoteinouse ;
    emvadon = katheti1 * katheti2 / 2;
    cout << "Κάθετη 1: " << fixed << setprecision(2) << katheti1 << endl;
    cout << " Κάθετη 2: " << fixed << setprecision(2) << katheti2 << endl;
    cout << "Υποτείνουσα: " << fixed << setprecision(2) << ypoteinouse << endl;
    cout << "Περίμετρος: " << fixed << setprecision(2) << perimetros << endl;
    cout << "Εμβαδόν: " << fixed << setprecision(2) << emvadon << endl;
    return 0;
}
```

Μεταβλητές					Παρουσίαση
katheti1	katheti2	ypoteinouse	perimetros	emvadon	
5.5					Δώσε το μήκος της κάθετης 1: Δώσε το μήκος της κάθετης 2: Κάθετη 1: 5.50 Κάθετη 2: 8.00 Υποτείνουσα: 9.71 Περίμετρος: 23.21 Εμβαδόν: 22.00
	8.0				
		9.71	23.21	22.00	

(β) Δομή Διακλάδωσης

Παράδειγμα 1: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου: 2000, 100

```
// Rent_A_Car
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    int kivismos;
    float timi, poso, ekptosi;
    ekptosi = 0.0;
    cout << "Δώσε τον κυβισμό: ";
    cin >> kivismos;
    cout << "Δώσε την αρχική τιμή: ";
    cin >> timi;
    if ( kivismos >= 1500 )
        ekptosi = timi * 10/100;
    poso = timi - ekptosi;
    cout << "Ο κυβισμός είναι: " << kivismos << endl;
    cout << "Η αρχική τιμή είναι: " << fixed << setprecision(2) << timi << endl;
    cout << "Η έκπτωση είναι: " << fixed << setprecision(2) << ekptosi << endl;
    cout << "Η τιμή πληρωμής είναι: " << fixed << setprecision(2) << poso << endl;
    return 0;
}
```

Μεταβλητές				Απόφαση		Παρουσίαση
ekptosi	kivismos	timi	poso	kivismos>=1500	A/Ψ	Δώσε τον κυβισμό: Δώσε την αρχική τιμή: Ο κυβισμός είναι: 2000 Η αρχική τιμή είναι: 100.00 Η έκπτωση είναι: 10.00 Η τιμή πληρωμής είναι: 90.00
0.0						
	2000	100		2000>=1500	A	
10.0			90.0			

Παράδειγμα 2: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου για την περίπτωση 1: M, 45 και για την περίπτωση 2: F, 28 .

```
// Insurance
#include <iostream>
using namespace std;
int main() {
    int age, asfalistra;
    char sex;
    cout << "Δώσε το φύλο: ";
    cin >> sex;
    cout << "Δώσε την ηλικία: ";
    cin >> age;
    if ( sex == 'M' )
        if ( age < 25 )
            asfalistra = 300;
        else
            asfalistra = 250;
    else if ( age < 21 )
        asfalistra = 200;
    else
        asfalistra = 150;
    cout << "Ασφάλιστρα: " << asfalistra;
    return 0;
}
```

Περίπτωση 1

Μεταβλητές			Απόφαση				Παρουσίαση
age	asfalistra	sex	sex=='M'	A/Ψ	age<25	A/Ψ	Δώσε το φύλο: Δώσε την ηλικία: Ασφάλιστρα: 250
		M					
45							
	250		M==M	A	45<25	Ψ	

Περίπτωση 2

Μεταβλητές			Απόφαση				Παρουσίαση
age	asfalistra	sex	sex=='M'	A/Ψ	age<21	A/Ψ	Δώσε το φύλο: Δώσε την ηλικία: Ασφάλιστρα: 150
		F					
28							
	150		F==M	Ψ	28<21	Ψ	

(γ) Επαναληπτική Δομή

Παράδειγμα 1: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος.

```
// Average
#include <iostream>
#include <iomanip>
using namespace std;
int main() {
    int no, sum, metritis;
    float mesos;
    sum = 0;
    metritis = 0;
    for ( no=10; no<=15; no++ ) {
        sum = sum + no;
        metritis = metritis + 1;
    }
    mesos = (float) sum / metritis;
    cout << "Ο μέσος όρος είναι: " << fixed << setprecision(2) << mesos;
    return 0;
}
```

Μεταβλητές				Απόφαση		Παρουσίαση
no	sum	metritis	mesos	no<=15	A/Ψ	Ο μέσος όρος είναι: 12.50
10	0	0		10<=15	A	
	10	1				
11	21	2		11<=15	A	
12	33	3		12<=15	A	
13	46	4		13<=15	A	
14	60	5		14<=15	A	
15	75	6		15<=15	A	
16				16<=15	Ψ	
			12.50			

Παράδειγμα 2: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος.

```
// Sum_Of_Odd_Nos;

#include <iostream>

using namespace std;

int main() {

    int arithmos, sum;

    sum = 0;

    arithmos = 1;

    while ( arithmos <= 11 ) {

        sum = sum + arithmos;

        arithmos = arithmos + 2;

    }

    cout << "Το άθροισμα των μονών αριθμών (1-11) είναι: " << sum;

    return 0;

}
```

Μεταβλητές		Απόφαση		Παρουσίαση
arithmos	sum	arithmos<=11	A/Ψ	Το άθροισμα των μονών αριθμών (1-11) είναι: 36
1	0	1<=11	A	
	1			
3	4	3<=11	A	
5	9	5<=11	A	
7	16	7<=11	A	
9	25	9<=11	A	
11	36	11<=11	A	
13		13<=11	Ψ	

Παράδειγμα 3: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος.

```
// Sum_Of_Odd_Nos;
#include <iostream>
using namespace std;
int main() {
    int arithmos, sum;
    sum = 0;
    arithmos = 1;
    do {
        sum = sum + arithmos;
        arithmos = arithmos + 2;
    } while ( arithmos <= 11);
    cout << "Το άθροισμα των μονών αριθμών (1-11) είναι: " << sum;
    return 0;
}
```

Μεταβλητές		Απόφαση		Παρουσίαση
arithmos	sum	arithmos<=11	A/Ψ	Το άθροισμα των μονών αριθμών (1-11) είναι: 36
1	0			
3	1	3<=11	A	
5	4	5<=11	A	
7	9	7<=11	A	
9	16	9<=11	A	
11	25	11<=11	A	
13	36	13<=11	Ψ	

