ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΘΡΗΣΚΕΥΜΑΤΩΝ ΚΑΙ ΑΘΛΗΤΙΣΜΟΥ
ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

Γώγουλος Γ., Κοτσιφάκης Γ., Κυριακάκη Γ.,
Παπαγιάννης Α., Φραγκονικολάκης Μ., Χίνου Π.

Εισαγωγή στις Αρχές της Επιστήμης των Η/Υ

ΒΙΒΛΙΟ (ΣΗΜΕΙΩΣΕΙΣ) ΜΑΘΗΤΗ

Β΄ ΕΠΑ.Λ.

ΓΕΝΙΚΗΣ ΠΑΙΔΕΙΑΣ

ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΚΔΟΣΕΩΝ «ΔΙΟΦΑΝΤΟΣ»

Εισαγωγή στις Αρχές της Επιστήμης των Η/Υ

Β΄ Τάξη ΕΠΑ.Λ.

ΓΕΝΙΚΗΣ ΠΑΙΔΕΙΑΣ

ΒΙΒΛΙΟ (ΣΗΜΕΙΩΣΕΙΣ) ΜΑΘΗΤΗ

ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ
Πρόεδρος: *Σωτήριος Γκλαβάς*

ΓΡΑΦΕΙΟ ΕΡΕΥΝΑΣ, ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΕΦΑΡΜΟΓΩΝ Β'
Προϊστάμενος: *Παύλος Μάραντος*

ΤΟΜΕΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
Υπεύθυνος: *Θεοδόσιος Τσαπέλας,* Σύμβουλος Β΄ Πληροφορικής ΙΕΠ

ΣΥΓΓΡΑΦΕΙΣ:
Γώγουλος Γιώργος, Σχολικός Σύμβουλος Πληροφορικής-ΠΕ19 Δυτ. Κρήτης
Κοτσιφάκης Γιώργος, Καθηγητής Πληροφορικής-ΠΕ19
Κυριακάκη Γεωργία, Καθηγήτρια Πληροφορικής-ΠΕ19
Παπαγιάννης Αποστόλης, Καθηγητής Πληροφορικής-ΠΕ19
Φραγκονικολάκης Μανόλης, Καθηγητής Πληροφορικής-ΠΕ19
Χίνου Παναγιώτα, Καθηγήτρια Πληροφορικής-ΠΕ19

ΣΥΝΤΟΝΙΣΤΗΣ
Γώγουλος Γιώργος, Σχολικός Σύμβουλος Πληροφορικής-ΠΕ19 Δυτ. Κρήτης

ΕΠΙΤΡΟΠΗ ΚΡΙΣΗΣ:
Κανίδης Ευάγγελος, Σχ. Σύμβουλος Πληροφορικής-ΠΕ19
Κοτίνη Ισαβέλλα, Σχ. Σύμβουλος Πληροφορικής-ΠΕ19
Τζελέπη Σοφία, Σχ. Σύμβουλος Πληροφορικής-ΠΕ19

ΠΡΟΕΚΤΥΠΩΤΙΚΕΣ ΕΡΓΑΣΙΕΣ: ΔΙΕΥΘΥΝΣΗ ΕΚΔΟΣΕΩΝ / ΙΤΥΕ «ΔΙΟΦΑΝΤΟΣ»

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΘΡΗΣΚΕΥΜΑΤΩΝ ΚΑΙ ΑΘΛΗΤΙΣΜΟΥ
ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

Γώγουλος Γ., Κοτσιφάκης Γ.,
Κυριακάκη Γ., Παπαγιάννης Α.,
Φραγκονικολάκης Μ., Χίνου Π.

Εισαγωγή στις Αρχές της Επιστήμης των Η/Υ

Β΄ Τάξη ΕΠΑ.Λ.

ΓΕΝΙΚΗΣ ΠΑΙΔΕΙΑΣ

ΒΙΒΛΙΟ (ΣΗΜΕΙΩΣΕΙΣ) ΜΑΘΗΤΗ

ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΚΔΟΣΕΩΝ «ΔΙΟΦΑΝΤΟΣ»

Πρόλογος

Το βιβλίο αυτό απευθύνεται στους μαθητές της Β΄ τάξης Επαγγελματικού Λυκείου που παρακολουθούν το μάθημα γενικής παιδείας «Εισαγωγή στις Αρχές της Επιστήμης των Η/Υ», το οποίο έχει ως γενικό σκοπό να γνωρίσουν οι μαθητές βασικούς τομείς και θεμελιώδεις έννοιες της Επιστήμης των Υπολογιστών και παράλληλα να αναπτύξουν αναλυτική και συνθετική σκέψη. Το βιβλίο καλύπτει το πρόγραμμα σπουδών του μαθήματος και διαρθρώνεται σε πέντε κεφάλαια:

Στο 1ο Κεφάλαιο γίνεται μία εισαγωγή στις βασικές έννοιες και τομείς της Επιστήμης των Υπολογιστών, που ορισμένοι από αυτούς εξετάζονται αναλυτικότερα στα επόμενα κεφάλαια.

Στα επόμενα Κεφάλαια (2ο, 3ο, 4ο & 5ο) καλύπτονται έννοιες της Θεωρητικής Επιστήμης των Υπολογιστών όπως πρόβλημα, αλγόριθμος, γλώσσες περιγραφής αλγορίθμων και δομές δεδομένων αντίστοιχα, δίνοντας έμφαση στη διασύνδεση της θεωρητικής προσέγγισης των εννοιών με καθημερινά προβλήματα και εφαρμογές, καθώς και το σύγχρονο γίγνεσθαι στην ανάπτυξη αλγορίθμων.

Το βιβλίο επιχειρεί μέσα από δομημένα και στοχευμένα παραδείγματα να υποστηρίξει την κατανόηση των εννοιών και να δώσει ερεθίσματα για περαιτέρω διερεύνηση και μελέτη τομέων της Επιστήμης των Υπολογιστών. Ακολουθούνται σύγχρονες προτάσεις για τη διδασκαλία και τη μάθηση που προωθούν και δίνουν ιδιαίτερη έμφαση στη διαδικασία και όχι μόνο στο περιεχόμενο. Στην αρχή κάθε κεφαλαίου γνωστοποιούνται οι στόχοι του κεφαλαίου σε επίπεδο πρόθεσης ενώ στο τέλος των επιμέρους ενοτήτων δίνεται η δυνατότητα αυτό-αξιολόγησης του μαθητή σε επίπεδο προσδοκώμενων μαθησιακών αποτελεσμάτων.

Ιούλιος 2015
Οι συγγραφείς

Για την καλύτερη ανάγνωση και αξιοποίηση του περιεχομένου του βιβλίου, εκτός από σχήματα, πίνακες, πλαίσια και εικόνες έχουν χρησιμοποιηθεί διαφορετικά χρώματα σε κάθε κεφάλαιο του βιβλίου και αρκετά εικονίδια τα οποία χαρακτηρίζουν την πληροφορία που συνοδεύουν.

Χρώματα κεφαλαίων:

1 Κεφάλαιο 1

2 Κεφάλαιο 2

3 Κεφάλαιο 3

4 Κεφάλαιο 4

5 Κεφάλαιο 5

Τα εικονίδια και η σημασία τους αναφέρονται παρακάτω:

Μαθησιακοί Στόχοι

Εισαγωγικές ερωτήσεις

Ερωτήσεις - Θέματα για συζήτηση

Αυτοαξιολόγηση

Λέξεις - κλειδιά

Ορισμός

Ασκήσεις - Προβλήματα

Περιεχόμενα

1. Βασικές έννοιες 11
1.1 Πληροφορική ή Επιστήμη Υπολογιστών; 11
1.2 Θεωρητική Επιστήμη των Υπολογιστών 13
1.3 Εφαρμοσμένη Επιστήμη των Υπολογιστών 13
2. Η έννοια του προβλήματος 17
2.1 Το πρόβλημα στην επιστήμη των Η/Υ 17
2.2 Κατηγορίες προβλημάτων 17
2.3 Διαδικασία επίλυσης προβλημάτων 18
3. Αλγόριθμοι 27
3.1 Η έννοια του αλγορίθμου 27
3.2 Χαρακτηριστικά αλγορίθμου 30
3.3 Ανάλυση αλγορίθμων 33
3.4 Βασικοί τύποι αλγορίθμων 36
3.5 Αναπαράσταση αλγορίθμου 40
4. Γλώσσες αναπαράστασης αλγορίθμων 47
4.1 Γενικά στοιχεία αναπαράστασης αλγορίθμων 47
4.2 Δομή Ακολουθίας 51
4.3 Δομή Επιλογής 55
4.4 Δομή Επανάληψης 64
4.5 Συντακτικά και λογικά λάθη 74
4.6 Τεκμηρίωση 76
5. Δεδομένα & Δομές Δεδομένων 81
5.1 Τα δεδομένα και η αναπαράστασή τους 81
5.2 Δομές δεδομένων 85
5.3 Πίνακες 88
Βιβλιογραφία 99
Πηγές on-line 99
Λεξικό Βασικών Όρων 100
Ευρετήριο εικόνων 102
Ευρετήριο αλγορίθμων 103

κεφάλαιο 1

Βασικές Έννοιες

Επιστήμη των Υπολογιστών

Στόχοι

Στόχος του κεφαλαίου είναι οι μαθητές:
• να γνωρίσουν βασικές έννοιες και τομείς της Επιστήμης των Υπολογιστών.

Λέξεις κλειδιά

Επιστήμη των Υπολογιστών	Πληροφορική, Θεωρητική Επιστήμη των Υπολογιστών, Εφαρμοσμένη Επιστήμη των Υπολογιστών

Εισαγωγικές ερωτήσεις

1. Γνωρίζετε τι είναι Επιστήμη και τι είναι Τεχνολογία; Πιστεύετε ότι η Πληροφορική είναι Επιστήμη ή Τεχνολογία;
2. Γνωρίζετε κλάδους της Επιστήμης των Υπολογιστών;

1. Βασικές έννοιες

1.1 Πληροφορική ή Επιστήμη Υπολογιστών;

Ο όρος «πληροφορική» προέρχεται από τη λέξη πληροφορία. Εμφανίστηκε τη δεκαετία του 1960 στη Γαλλία (Informatique) και τη Γερμανία (Informatik). Από την ετυμολογία της λέξης προκύπτει ο ορισμός:

Πληροφορική είναι η επιστήμη που ασχολείται με την αναπαράσταση, αποθήκευση και επεξεργασία της πληροφορίας.

Η Πληροφορική είναι άρρηκτα συνδεδεμένη με τη μηχανή που λέγεται ηλεκτρονικός υπολογιστής. Γι' αυτό και στον Αγγλόφωνο χώρο, έχει επικρατήσει η ονομασία Επιστήμη Υπολογιστών (Computer Science). Στη νέα επιστήμη όσο σημαντικός είναι ο υπολογιστής (computer), άλλο τόσο σημαντικές είναι οι διαδικασίες υπολογισμού (computing) που μπορεί να κάνει.

Επιστήμη Υπολογιστών είναι η επιστήμη που ασχολείται με τους υπολογιστές και τους υπολογισμούς.

Για μας, στο εξής, Πληροφορική και Επιστήμη Υπολογιστών αναφέρονται στην ίδια επιστήμη. Θεμελιώδεις έννοιες στην Επιστήμη των Υπολογιστών είναι:

- η ίδια η μηχανή, ο υπολογιστής και
- η υπολογιστική διαδικασία που η μηχανή μπορεί να επιτελέσει, το πρόγραμμα.

Ιστορικοί σταθμοί της Επιστήμης Υπολογιστών	
300 π.Χ.	Αλγόριθμος του Ευκλείδη για τον υπολογισμό του μέγιστου κοινού διαιρέτη δύο φυσικών αριθμών, ένας από τους αρχαιότερους αλγόριθμους
100 π.Χ.	Μηχανισμός των Αντικυθήρων, μηχανικός υπολογιστής και όργανο αστρονομικών παρατηρήσεων
820 μ.Χ.	Ο al-Khwarizmi, Πέρσης μαθηματικός και αστρονόμος γράφει βιβλίο για τους αλγόριθμους
1822	Ο Charles Babbage σχεδιάζει την Αναλυτική Μηχανή του, την πρώτη μηχανή με αποθηκευμένα προγράμματα.
1940	Ο John von Neumann θέτει τις θεμελιώδεις αρχές σχεδίασης των σύγχρονων ΗΥ, γνωστές ως "αρχιτεκτονική von Neumann", στο Πανεπιστήμιο του Πρίνστον.
1944	Ο Aiken κατασκευάζει τον Mark I. Χρόνος για πρόσθεση: 1/3 s., για πολλαπλασιασμό: 6 s
1946	Οι Eckert και Maughly κατασκευάζουν τον ENIAC, τον πρώτο ηλεκτρονικό υπολογιστή με 18000 λυχνίες. Χρόνος για πολλαπλασιασμό: 6 ms
1949	Ο Wilkes κατασκευάζει τον EDSAC, τον πρώτο ψηφιακό υπολογιστή γενικής χρήσης με αποθηκευμένο πρόγραμμα (υπολογιστής τύπου von Neumann)
1969	Γέννηση του διαδικτύου με τη δημιουργία του δικτύου ARPANET
1991	Δημιουργία του Παγκόσμιου Ιστού (WWW) στο CERN

Η παραπάνω ιστορική αναδρομή εμφανίζει κυρίως τεχνολογικά επιτεύγματα και όχι επιστημονικά. Μήπως, λοιπόν, η Πληροφορική είναι απλώς η τεχνολογία των υπολογιστών και δεν είναι επιστήμη; Η απάντηση είναι ότι η Πληροφορική είναι δισδιάστατη, είναι και τεχνολογία και επιστήμη. Απλά, ο ηλεκτρονικός υπολογιστής καταλαμβάνει τόσο κυρίαρχο ρόλο σε αυτήν που δεν είναι δυνατό να διαχωρίσουμε τα τεχνολογικά από τα επιστημονικά επιτεύγματα.

Μια από τις δόκιμες διακρίσεις της Επιστήμης Υπολογιστών διακρίνει τη Θεωρητική από την Εφαρμοσμένη Επιστήμη. Στον παρακάτω πίνακα φαίνονται κύριοι τομείς της Επιστήμης Υπολογιστών με διαφορετική απόχρωση ανάλογα με το αν ανήκει στη Θεωρητική ή Εφαρμοσμένη.

Εικόνα 1-1. Ταξινόμηση της Επιστήμης Υπολογιστών από την ACM (Association for Computing Machinery)

Επιστήμη Υπολογιστών (Computer Science)	
Λογισμικό και μηχανική λογισμικού (Software and its engineering)	Γλώσσες προγραμματισμού, προγραμματιστικά υποδείγματα, Λειτουργικά Συστήματα
Θεωρία υπολογισμού (Theory of computation)	Ανάλυση και σχεδίαση αλγορίθμων, Πολυπλοκότητα και υπολογισιμότητα αλγορίθμων
Μαθηματικά της Πληροφορικής (Mathematics of computing)	Διακριτά μαθηματικά, Θεωρία γράφων, Πιθανότητες και στατιστική, Μαθηματική ανάλυση και μαθηματική βελτιστοποίηση
Υλικό (Hardware)	Ολοκληρωμένα κυκλώματα, μνήμες και μέσα αποθήκευσης
Οργάνωση και Αρχιτεκτονική Η/Υ (Computer organization)	Έλεγχος και δοκιμή υλικού, συστήματα πραγματικού χρόνου
Δίκτυα υπολογιστών (Networks)	Αρχιτεκτονικές, πρωτόκολλα και υπηρεσίες δικτύων
Πληροφοριακά συστήματα (Information systems)	Συστήματα διαχείρισης δεδομένων, βάσεις δεδομένων, ανάκτηση πληροφοριών, συστήματα υποστήριξης αποφάσεων
Ασφάλεια και Ιδιωτικότητα (Security and privacy)	Κρυπτογράφηση, συστήματα ασφάλειας δεδομένων
Ανθρωποκεντρικός υπολογισμός (Human-centered computing)	Διεπαφή χρήστη-υπολογιστή, προσβασιμότητα
Μεθοδολογίες υπολογισμού (Computing methodologies)	Παράλληλοι υπολογιστές, Τεχνητή νοημοσύνη, Ρομποτική, Μηχανική όραση, Γραφική Υπολογιστών, Κατανεμημένα συστήματα
Εφαρμογές της Πληροφορικής σε άλλους τομείς (Applied computing)	Φυσικές επιστήμες, Μηχανική, Ηλεκτρονικό εμπόριο, Επιχειρήσεις, Αστρονομία, Αεροναυπηγική, Χημεία, Φυσική, Βιολογία, Εκπαίδευση, Νομικές και κοινωνικές επιστήμες, Καλές τέχνες

1.2 Θεωρητική Επιστήμη των Υπολογιστών

Θεμελιώδεις έννοιες της Θεωρητικής Επιστήμης των Υπολογιστών είναι ο **αλγόριθμος** και το **πρόγραμμα**. Μια μηχανικά εκτελέσιμη υπολογιστική διαδικασία αποτελεί αλγόριθμο. Το θεωρητικό μέρος της Επιστήμης Υπολογιστών ασχολείται με τη σχεδίαση, τη δημιουργία και την αναπαράσταση των αλγορίθμων.

Η μεταφορά του αλγορίθμου σε μια μορφή κατανοητή από τον ηλεκτρονικό υπολογιστή, δηλαδή σε μια **γλώσσα προγραμματισμού**, δημιουργεί ένα πρόγραμμα. Στη Θεωρητική Επιστήμη των Υπολογιστών εξετάζονται τεχνικές σχεδίασης προγραμμάτων και μελετώνται οι γλώσσες προγραμματισμού. Ένας άλλος τομέας της θεωρητικής Πληροφορικής ασχολείται με τα δεδομένα και τους τρόπους οργάνωσης αυτών, τις **δομές δεδομένων**.

Στο θεωρητικό μέρος της Επιστήμης Υπολογιστών ανήκει επίσης ο τομέας της **θεωρίας υπολογισμού**. Ο τομέας αυτός αποτελεί τη μαθηματική βάση της Επιστήμης των Υπολογιστών που έχει ως αντικείμενο να εκφράσει και να μελετήσει, με τη χρήση μαθηματικών μοντέλων, τη λειτουργία των υπολογιστικών μηχανών, την υπολογισιμότητα των προβλημάτων (αν μπορούν ή όχι να επιλυθούν από υπολογιστή), καθώς και τους υπολογιστικούς πόρους που απαιτούνται για την επίλυση προβλημάτων.

Θεωρητική Επιστήμη των Υπολογιστών

Επίλυση Προβλημάτων
Αλγόριθμοι
Ανάλυση Αλγορίθμων
Δομές Δεδομένων
Μέθοδοι Προγραμματισμού
Γλώσσες Προγραμματισμού
Μεταγλωττιστές
Θεωρία Αυτομάτων
Θεωρία Υπολογισμών
Θεωρία Πολυπλοκότητας
Τεχνολογία Λογισμικού

1.3 Εφαρμοσμένη Επιστήμη των Υπολογιστών

Η άρρηκτη σχέση της θεωρητικής επιστήμης με την υπολογιστική μηχανή δημιουργεί πεδία εφαρμογής τόσο στην ίδια την Πληροφορική όσο και σε άλλες Επιστήμες.

Η Εφαρμοσμένη Επιστήμη των Υπολογιστών ασχολείται με το **υλικό** (hardware) και την **οργάνωση** και **αρχιτεκτονική υπολογιστών**. Ασχολείται επίσης με τα **λειτουργικά συστήματα**, προγράμματα μεγάλης κλίμακας που τοποθετούνται ανάμεσα στο υλικό και το λογισμικό του υπολογιστή. Τα **πληροφοριακά συστήματα** συνδυάζουν υλικό, λογισμικό, δεδομένα, ανθρώπους και διαδικασίες. Η διασύνδεση υπολογιστών σε δίκτυα, δημιουργεί έναν άλλο τομέα. Ο τομέας της **τεχνητής νοημοσύνης** ασχολείται με τη δημιουργία προγραμμάτων που κάνουν τον υπολογιστή να προσομοιάζει σε νοήμον ον.

Τέλος, ένα μέρος της Εφαρμοσμένης Επιστήμης των Υπολογιστών διερευνά τις δυνατότητες εφαρμογής των ηλεκτρονικών υπολογιστών σε άλλες επιστήμες, όπως στη Μηχανική, στην Οικονομία, στην Ιατρική, στις Φυσικές επιστήμες, στις Ανθρωπιστικές επιστήμες και στις Καλές τέχνες. Οι εφαρμογές αυτές, συχνά δημιουργούν νέους επιστημονικούς τομείς όπως η Βιοϊατρική.

Εφαρμοσμένη Επιστήμη των Υπολογιστών

Υλικό Υπολογιστή
Ολοκληρωμένα κυκλώματα
Μικροεπεξεργαστές
Οργάνωση και αρχιτεκτονική υπολογιστών
Δίκτυα υπολογιστών
Λειτουργικά Συστήματα
Επικοινωνία Ανθρώπου-Μηχανής
Πληροφοριακά συστήματα
Συστήματα διαχείρισης βάσεων δεδομένων
Τεχνητή νοημοσύνη
Εφαρμογές σε άλλες επιστήμες

Καλώς ήρθατε στον κόσμο της Πληροφορικής!

Ερωτήσεις - Θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Η Πληροφορική και η Επιστήμη Υπολογιστών αναφέρονται στην ίδια επιστήμη.
2. Πληροφορική είναι απλώς η τεχνολογία των υπολογιστών και δεν είναι επιστήμη.
3. Η υπολογισιμότητα των προβλημάτων αφορά στο αν μπορούν ή όχι να επιλυθούν από υπολογιστή.

Β. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Θεμελιώδεις έννοιες στην Επιστήμη των Υπολογιστών είναι η ίδια η μηχανή, ο _____ και η _____ διαδικασία.
2. Η Επιστήμη των Υπολογιστών διακρίνεται στη _____ και στην _____ Επιστήμη των Υπολογιστών.
3. Η θεωρία υπολογισμού αποτελεί _____ τη βάση της επιστήμης των υπολογιστών.
4. Η _____ ερευνά τις δυνατότητες εφαρμογής των ηλεκτρονικών υπολογιστών στην Ιατρική.

Γ. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με την κατάλληλη έννοια της στήλης Β

Στήλη Α	**Στήλη Β**
1. Υλικό Υπολογιστή	Α. Θεωρητική Επιστήμη των Υπολογιστών
2. Ανάλυση Αλγορίθμων	
3. Δομές Δεδομένων	
4. Δίκτυα υπολογιστών	
5. Οργάνωση και αρχιτεκτονική υπολογιστών	Β. Εφαρμοσμένη Επιστήμη των Υπολογιστών
6. Γλώσσες Προγραμματισμού	
7. Πληροφοριακά συστήματα	
8. Τεχνητή νοημοσύνη	
9. Θεωρία Υπολογισμών	
10. Αλγόριθμοι	
11. Τεχνολογία Λογισμικού	
12. Λειτουργικά Συστήματα	

κεφάλαιο 2

Θέματα Θεωρητικής Επιστήμης των Υπολογιστών

Η έννοια του προβλήματος

Στόχοι

Στόχοι του κεφαλαίου είναι οι μαθητές:

- να έρθουν σε επαφή με την έννοια του προβλήματος.
- να μπορούν να διακρίνουν τις κατηγορίες των προβλημάτων με κριτήριο τη δυνατότητα επίλυσής τους.
- να ακολουθούν τα στάδια επίλυσης ενός προβλήματος στην αντιμετώπιση υπολογιστικών προβλημάτων.
- να αναγνωρίζουν τις βασικές μεθοδολογίες επίλυσης προβλημάτων και να τις εφαρμόζουν σε δοσμένα προβλήματα.

Λέξεις κλειδιά

Πρόβλημα	Επιλύσιμα - Μη επιλύσιμα - Ανοικτά προβλήματα, Υπολογιστικά προβλήματα, Προβλήματα Απόφασης, Προβλήματα Βελτιστοποίησης, Κατανόηση - Επίλυση προβλήματος, Δεδομένα, Ζητούμενα, Ορθότητα-Πληρότητα-Σαφήνεια προβλήματος, Μεθοδολογίες επίλυσης προβλημάτων

Εισαγωγικές ερωτήσεις

1. Γνωρίζετε όχι ο “τετραγωνισμός του κύκλου” είναι μη επιλύσιμο πρόβλημα;
2. Τι σας δυσκολεύει περισσότερο στην εκφώνηση ενός προβλήματος;
3. Αλλάζετε τρόπο αντιμετώπισης ενός προβλήματος της καθημερινής σας ζωής ανάλογα με τις συνθήκες που το περιβάλλουν;

2. Η έννοια του προβλήματος

2.1 Το πρόβλημα στην επιστήμη των Η/Υ

Τη λέξη πρόβλημα την έχουμε συναντήσει πολλές φορές. Έχουμε λύσει πολλά προβλήματα από την αρχή της σχολικής μας ζωής σε διάφορα μαθήματα όπως τα Μαθηματικά τη Φυσική, τη Χημεία κ.ά.. Όμως προβλήματα αντιμετωπίζουμε και καθημερινά στη ζωή μας όπως, για παράδειγμα, πώς θα πάω στο σχολείο, τί θα φάω σήμερα, πού θα πάω βόλτα με τους φίλους μου, πώς θα τακτοποιήσω το δωμάτιο μου. Τα προβλήματα αυτά θεωρούνται απλά, υπάρχουν όμως και πιο σύνθετα όπως η ρύπανση του περιβάλλοντος, η αντιμετώπιση φυσικών φαινομένων, η αποκωδικοποίηση του DNA, η επαγγελματική σταδιοδρομία, η ανεργία, η θεραπεία ασθενειών κ. ά.

Όλα τα προβλήματα είναι καταστάσεις που πρέπει να αντιμετωπιστούν αλλά δε γνωρίζουμε το πώς. Προβλήματα προκύπτουν όταν ένα εμπόδιο δυσκολεύει την επίτευξη ενός στόχου. Για παράδειγμα, μια βλάβη (το εμπόδιο) σε ένα εργοστάσιο παραγωγής εμποδίζει την εκπλήρωση των παραγγελιών (ο στόχος). Μόλις το εμπόδιο παρακαμφθεί, έχουμε φτάσει στη λύση του προβλήματος.

Στόχος είναι αυτό που έχουμε αποφασίσει ότι πρέπει να επιτύχουμε.

Εμπόδιο είναι αυτό που μας δυσκολεύει στην επίτευξη ενός στόχου.

Ο Jackson (1985) συνοψίζει:

Στόχος + Εμπόδιο = ΠΡΟΒΛΗΜΑ

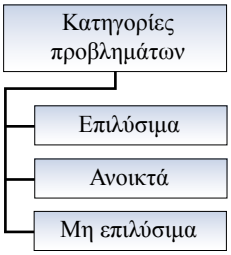
Με άλλα λόγια, **πρόβλημα** είναι μια κατάσταση που απαιτεί λύση αλλά η λύση της δεν είναι γνωστή ούτε προφανής.

2.2 Κατηγορίες προβλημάτων

Αν παρατηρήσουμε τα προβλήματα που αναφέρθηκαν παραπάνω, διαπιστώνουμε τη διαφορετική τους φύση. Η κατηγοριοποίησή τους μπορεί να γίνει με βάση διάφορα κριτήρια. Αν ως κριτήριο θέσουμε τη δυνατότητα επίλυσής τους, τα προβλήματα διακρίνονται σε:

- **Επιλύσιμα**. Χαρακτηρίζονται τα προβλήματα των οποίων η λύση έχει διατυπωθεί ή η συνάφειά τους με άλλα, ήδη λυμένα, μας επιτρέπει να θεωρούμε βέβαιη την δυνατότητα επίλυσής τους. Παραδείγματα τέτοιων προβλημάτων είναι η επίλυση δευτεροβάθμιων εξισώσεων, ο υπολογισμός του ρεύματος σε ένα κύκλωμα, η κατασκευή μιας γέφυρας, η εξοικονόμηση ενέργειας, η οργάνωση μιας βιβλιοθήκης.
- **Ανοικτά**. Χαρακτηρίζονται τα προβλήματα των οποίων η λύση δεν έχει ακόμα βρεθεί, ούτε έχει αποδειχθεί ότι δεν επιδέχονται λύση. Ως ανοικτά προβλήματα μπορούμε να αναφέρουμε την ανακάλυψη ζωής σε άλλους πλανήτες, τη θεραπεία του καρκίνου, την πρόβλεψη των σεισμών.
- **Μη επιλύσιμα**. Χαρακτηρίζονται τα προβλήματα για τα οποία έχουμε καταλήξει στην παραδοχή ότι δεν μπορούν να λυθούν. Τέτοια προβλήματα είναι ο τετραγωνισμός του κύκλου, το ταξίδι στο παρελθόν, η γήρανση του ανθρώπου.

Τα προβλήματα που δίνονται προς επίλυση στον υπολογιστή έχουν να κάνουν με τη διενέργεια υπολογισμών οι οποίοι απαιτούν μια σειρά από λογικές και αριθμητικές πράξεις. Τα προβλήματα αυτά λέγονται υπολογιστικά και είναι αυτά με τα οποία θα ασχοληθούμε κυρίως σε αυτό το βιβλίο.



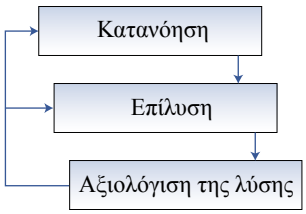
Εικόνα 2-1. Κατηγορίες προβλημάτων με κριτήριο την επιλυσιμότητα

Η εικασία του Goldbach ότι όλοι οι θετικοί άρτιοι αριθμοί μπορούν να γραφούν ως άθροισμα δύο πρώτων αριθμών, παραμένει ανοικτό πρόβλημα

Στα υπολογιστικά προβλήματα ζητούμε να βρούμε την απάντηση που ικανοποιεί τα δεδομένα του προβλήματος. Η πλειοψηφία των προβλημάτων απαιτούν τη διενέργεια πράξεων υπάρχουν όμως, και υπολογιστικά προβλήματα απόφασης ή βελτιστοποιήσης. **Απόφασης** είναι ένα πρόβλημα στο οποίο η απάντηση είναι ένα ναι ή ένα όχι. Για παράδειγμα, δεδομένου ότι ο καιρός είναι άστατος, να πάω στο σχολείο με το ποδήλατο; **Βελτιστοποίησης** είναι ένα πρόβλημα στο οποίο αναζητούμε την απάντηση που ικανοποιεί κατά τον καλύτερο τρόπο τα δεδομένα του. Για παράδειγμα, δεδομένου ότι διαθέτω 100 ευρώ, ποιο είναι το καλύτερο κινητό που μπορώ να αγοράσω;

2.3 Διαδικασία επίλυσης προβλημάτων

Η αντιμετώπιση υπολογιστικών προβλημάτων γίνεται με συγκεκριμένη διαδικασία που περιέχει σαφή βήματα.



Εικόνα 2-2. Στάδια επίλυσης προβλήματος

1. Κατανόηση και Παρουσίαση προβλήματος

Σημαντικός παράγοντας για την **κατανόηση** του προβλήματος αλλά και τον καθορισμό της λύσης του είναι ο προσδιορισμός του χώρου του προβλήματος. Ο **χώρος του προβλήματος** καθορίζεται με τη βοήθεια ερωτήσεων («πώς», «πού», «τί», «γιατί», «πότε»). Απαντώντας στα ερωτήματα προσδιορίζουμε τα δεδομένα και τα ζητούμενα του προβλήματος. (Είσοδος - Έξοδος)

Δεδομένο προβλήματος ονομάζεται ένα γνωστό ή αποδεκτό στοιχείο το οποίο χρησιμοποιείται ως βάση ή προϋπόθεση για την επίλυση του προβλήματος.

Ζητούμενο προβλήματος είναι αυτό που ψάχνουμε για να βγούμε από τη δύσκολη κατάσταση στην οποία βρισκόμαστε.

Ανάλογα με τη φύση του προβλήματος τα δεδομένα και τα ζητούμενα μπορεί να είναι αριθμητικά, οικονομικά, λογικά κ.ά. Ανεξάρτητα από το είδος τους, τα δεδομένα και τα ζητούμενα πρέπει να έχουν τρεις (3) σημαντικές ιδιότητες: ορθότητα, πληρότητα και σαφήνεια.

Η **ορθότητα** είναι αναγκαίο να ελέγχεται κάθε φορά που επιδιώκεται η επίλυση ενός προβλήματος. Για παράδειγμα, έστω ότι σας ζητείται να ταξινομήσετε σε αλφαβητική σειρά τα επίθετα των συμμαθητών σας. Αν σας τα έχουν δώσει με ορθογραφικά λάθη, η ταξινόμηση που θα προκύψει θα είναι λανθασμένη. Πριν ξεκινήσουμε την προσπάθεια επίλυσης ενός προβλήματος πρέπει να ελέγξουμε την ορθότητα των δεδομένων.

Η **πληρότητα** πρέπει κι αυτή να ελέγχεται κάθε φορά που επιδιώκεται η επίλυση ενός προβλήματος. Έστω ότι στο παραπάνω παράδειγμα μας ζητάνε να ταξινομηθούν οι συμμαθητές μας με βάση την ημερομηνία γέννησης και μας δίνουν τα στοιχεία στην Εικόνα 2-3. Παρατηρούμε ότι τα δεδομένα είναι ελλιπή (εφόσον στην Παπαδάκη Μιχαέλα υπάρχει ημερομηνία γέννησης) και δεν μπορούμε να επιλύσουμε το πρόβλημα.

Η **σαφήνεια** είναι ενας σημαντικός παράγοντας για την ορθή επίλυση ενός προβλήματος. Τα δεδομένα αλλά και τα ζητούμενα, πρέπει να είναι **σαφή**, δηλαδή δεν υπάρχει ημερομηνία γέννησης) και δεν μπορούμε να επιλύσουμε το πρόβλημα.

Επίθετο	Όνομα	Ημερ. Γέννησης
Παπαδοπούλου	Κυριακή	3/22/98
Παπαδάκης	Κωνσταντίνος	2/25/98
Παπαδάκη	Μιχαέλα	...
Παπαδόπουλος	Μιχάλης	5/20/98

Εικόνα 2-3. Δεδομένα χωρίς πληρότητα

Η σαφήνεια είναι ενας σημαντικός παράγοντας για την ορθή επίλυση ενός προβλήματος. Τα δεδομένα αλλά και τα ζητούμενα, πρέπει να είναι σαφή, δηλαδή δεν απαιτούνται διευκρινιστικές ερωτήσεις από το πρόσωπο που καλείται να λύσει το πρόβλημα.

Σαφήνεια δεδομένων

Για παράδειγμα, έστω ότι μας ζητάει η μητέρα μας να τη βοηθήσουμε στην παρασκευή ενός κέικ. Το επόμενο πρωί, έτοιμοι να ξεκινήσουμε τη διαδικασία παίρνουμε τη συνταγή και διαπιστώνουμε ότι δεν υπάρχουν τα υλικά που χρειάζονται. Το πρόβλημα που αρχικά είχαμε αντιληφθεί είναι διαφορετικό από αυτό που προέκυψε.

Σαφήνεια ζητουμένων

Έστω ότι μας ζητείται από μια φίλη μας να αγοράσουμε μια σοκολάτα. Η προθυμία να την εξυπηρετήσουμε μας οδηγεί στο Super Market. Εκεί έχουμε να επιλέξουμε ανάμεσα σε πάρα πολλές μάρκες, ποιότητες αλλά και μεγέθη. Στη συγκεκριμένη περίπτωση αυτό που κάνουμε είναι ή να επιλέξουμε κάποια στην τύχη, ή να ζητήσουμε νέες, πιο σαφείς, οδηγίες, τηλεφωνικά.

Η παρουσίαση του προβλήματος καθορίζει σε γενικές γραμμές την επιθυμητή σχέση εισόδου-εξόδου.

Ένας ακόμα σημαντικός παράγοντας για την ορθή επίλυση ενός προβλήματος είναι ο τρόπος παρουσίασής του. Η παρουσίαση του προβλήματος μπορεί να γίνει φραστικά (με λέξεις) ή αλγεβρικά (με μαθηματικά σύμβολα).

Παράδειγμα 2-1. Βρείτε δύο αριθμούς που το άθροισμα τους είναι 78 και το γινόμενο τους 1296.

2. Επίλυση του προβλήματος

Στην προηγούμενη παράγραφο περιγράψαμε τη διαδικασία εύρεσης των δεδομένων και των ζητουμένων μέσα από την παρουσίαση του προβλήματος.

Φυσική Γλώσσα	Γλώσσα της Άλγεβρας
Βρείτε δύο αριθμούς	x,y
που το άθροισμα τους είναι 78	x+y=78
και το γινόμενο τους 1296	x.y= 1296

Εικόνα 2-4. Παρουσίαση του προβλήματος σε φυσική και αλγεβρική γλώσσα

Προχωρώντας προς το στάδιο της επίλυσης (problem solving) καταλαβαίνουμε ότι σε πολλά προβλήματα η λύση τους δεν είναι άμεσα γνωστή. Η επίλυση περιλαμβάνει την ανάλυση, μοντελοποίηση, σχεδίαση, και υλοποίηση μίας κατάλληλης λύσης του προβλήματος με τον υπολογιστή.

Βασικές έννοιες στην επίλυση προβλημάτων είναι η δομή, η ανάλυση και η σύνθεση. Με τον όρο **δομή**, εννοούμε τον τρόπο με τον οποίο επιμέρους στοιχεία σχετίζονται και συνδέονται μεταξύ τους ώστε να σχηματίζουν ενιαίο σύνολο. **Ανάλυση** είναι ο διαχωρισμός ενός συνόλου στα συστατικά του στοιχεία. Σύνθεση είναι η τοποθέτηση στοιχείων σε συσχετισμό μεταξύ τους έτσι ώστε να δημιουργείται ένα σύνολο.

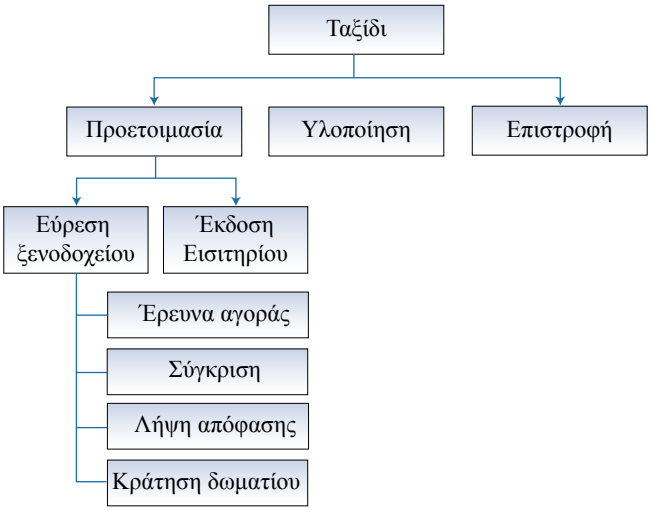
Όταν το ενιαίο σύνολο είναι το πρόβλημα, τα επιμέρους στοιχεία είναι τα μικρότερα προβλήματα στα οποία χωρίζεται, τα υποπροβλήματα. Εφαρμόζοντας την ανάλυση και τη σύνθεση στην επίλυση προβλημάτων δημιουργούνται τρεις βασικές μεθοδολογίες η Αναλυτική (Top Down), η Συνθετική (Bottom Up), και η Μικτή (Mixed).

Αναλυτική (Top Down problem solving) είναι η μεθοδολογία επίλυσης προβλημάτων που βασίζεται στη σχεδίαση από το γενικό στο ειδικό. Η γενική αρχή της είναι ότι για να λύσουμε κάποιο σύνθετο πρόβλημα πρέπει:

1. Να ορίσουμε τα υποπροβλήματα (sub-problems).
2. Να επαναλάβουμε την διαδικασία αυτή για κάθε ένα από τα υποπροβλήματα, όσο αυτό είναι αναγκαίο.
3. Όταν φτάσουμε σε υποπροβλήματα που δεν απαιτούν επιπλέον διάσπαση, προχωράμε στην άμεση επίλυσή τους, τότε έχει επιλυθεί και το αρχικό πρόβλημα.

Παράδειγμα 2-2. Ας δούμε ένα παράδειγμα εφαρμογής της αναλυτικής μεθοδολογίας: Θέλουμε να υλοποιήσουμε ένα ταξίδι στο εξωτερικό.

Εικόνα 2-5. Προετοιμασία ταξιδιού



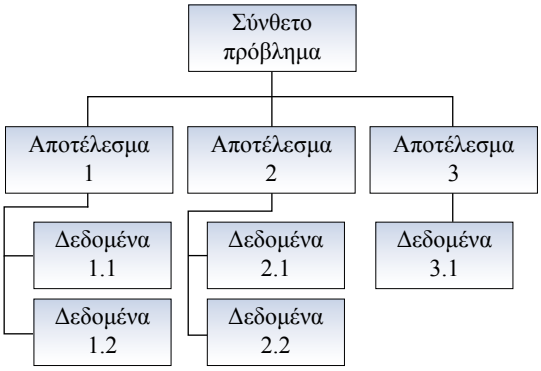
Υπάρχουν πολλοί τρόποι με τους οποίους μπορούμε να αναλύσουμε το πρόβλημα σε υποπροβλήματα. Ένας από όλους είναι να το αναλύσουμε αρχικά σε τρία υποπροβλήματα: προετοιμασία ταξιδιού, υλοποίηση και επιστροφή. Η προετοιμασία του ταξιδιού μπορεί να περιλαμβάνει έκδοση εισιτηρίων και εύρεση ξενοδοχείου. Το πρόβλημα της εύρεσης ξενοδοχείου με τη σειρά του μπορεί να αναλυθεί στα υποπροβλήματα της έρευνας αγοράς, σύγκρισης, λήψης απόφασης και κράτησης δωματίου.

Με τον ίδιο τρόπο θα μπορούσαμε να αναλύσουμε και τα υπόλοιπα υποπροβλήματα προχωρώντας την ανάλυση σε ακόμα χαμηλότερα επίπεδα. Η διαγραμματική αναπαράσταση της ανάλυσης του προβλήματος δίνεται στην Εικόνα 2.5.

Συνθετική (bottom up problem solving) είναι η μεθοδολογία επίλυσης προβλημάτων που βασίζεται στη σχεδίαση από το ειδικό στο γενικό με τη σύνδεση των δεδομένων. Για το σκοπό αυτό:

1. Αρχίζουμε από δεδομένα που είναι ορθά.
2. Στη συνέχεια, με μια λογικά ορθή διαδικασία, παράγουμε νέα δεδομένα και αποτελέσματα
3. Η διαδικασία τερματίζει, όταν παραχθεί το ζητούμενο του προβλήματος.

Μεικτή (mixed), είναι η μεθοδολογία επίλυσης προβλημάτων που συνδυάζει την αναλυτική και τη συνθετική μέθοδο. Σύμφωνα με τη μεικτή μεθοδολογία, κάποια από τα υποπροβλήματα επιλύονται με την αναλυτική και κάποια με τη συνθετική μέθοδο.



Εικόνα 2-6. Συνθετική μεθοδολογία

Η ανάλυση του προβλήματος μπορεί να αναπαρασταθεί είτε φραστικά είτε διαγραμματικά. Στη διαγραμματική αναπαράσταση, η περιγραφή έχει τη μορφή γενεολογικού δέντρου, στο οποίο κάθε πρόβλημα έχει «παιδιά» τα υποπροβλήματα στα οποία αναλύεται.

Μπορείτε να σκεφτείτε ένα πρόβλημα που να επιλύεται με τη συνθετική ή τη μεικτή μεθοδολογία;

3. Αξιολόγηση της λύσης

Το στάδιο αξιολόγησης της λύσης είναι πολύ σημαντικό. Για να ελέγξουμε εάν έχουμε επιλύσει σωστά ένα πρόβλημα, αρχικά καταγράφουμε υποθετικά δεδομένα σύμφωνα με τις απαιτήσεις του προβλήματος. Στη συνέχεια, εφαρμόζουμε τα βήματα επίλυσης για να εξάγουμε τα αποτελέσματα. Τέλος, συγκρίνουμε τα αποτελέσματα που πήραμε με τα ζητούμενα του προβλήματος. Εάν διαπιστώσουμε λάθος, εντοπίζουμε το τμήμα της λύσης που εκτελεί τη λανθασμένη λειτουργία, το διορθώνουμε και επαναλαμβάνουμε τη διαδικασία ελέγχου, έως ότου τα αποτελέσματα να μη διαφέρουν από τα ζητούμενα. Πολλές φορές το λάθος μπορεί να οφείλεται και στην κατανόηση του προβλήματος. Στην περίπτωση αυτή η διαδικασία επίλυσης επαναλαμβάνεται από την αρχή.

Ερωτήσεις - Θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Προβλήματα είναι καταστάσεις που πρέπει να αντιμετωπιστούν αλλά δε γνωρίζουμε το πώς.
2. Επιλύσιμα χαρακτηρίζονται τα προβλήματα των οποίων η συνάφειά της λύσης τους με άλλα, ήδη λυμένα, μας επιτρέπει να θεωρούμε βέβαιη την δυνατότητα επίλυσής τους.
3. Ανοικτά χαρακτηρίζονται τα προβλήματα για τα οποία έχουμε καταλήξει στην παραδοχή ότι δεν μπορούν να λυθούν.
4. Όταν τα δεδομένα ενός προβλήματος είναι ελλιπή ή ασαφή, τότε το πρόβλημα χαρακτηρίζεται ως μη επιλύσιμο.
5. Ο έλεγχος των δεδομένων μπορεί να οδηγήσει και πάλι στην είσοδο.
6. Δομή ενός προβλήματος είναι η εύρεση του συνόλου των μερών που το απαρτίζουν.
7. Η ανάλυση των προβλημάτων σε υποπροβλήματα πρέπει να αποφεύγεται γιατί με αυτόν τον τρόπο αντί να λύσουμε ένα πρόβλημα πρέπει να λύσουμε πολλά προβλήματα.
8. Η λύση της εξίσωσης 3χ+7=16 είναι πρόβλημα υπολογιστικό.

Β. Σημειώστε την κατηγορία στην οποία ανήκει το κάθε πρόβλημα.

	Επιλύσιμο	**Ανοικτό**	**Μη επιλύσιμο**
1. Ο υπολογισμός της υποτείνουσας ενός ορθογωνίου τριγώνου όταν γνωρίζω τις 2 πλευρές του.			
2. Ακριβής πρόγνωση σεισμικών δονήσεων.			
3. Θεραπεία του aids.			
4. Η οργάνωση μιας βιβλιοθήκης			
5. Κατασκευή αυτοκινήτου με αυτόματο «πιλότο».			
6. Η γήρανση του ανθρώπου			
7. Παρασκευή εμβολίου για τον καρκίνο.			
8. Η εικασία του Γκόλντμπαχ (Goldbach, κάθε άρτιος μπορεί να γραφτεί ως άθροισμα δύο πρώτων αριθμών)			
9. Το πρόβλημα του τετραγωνισμού του κύκλου με κανόνα και διαβήτη			
10. Η επίλυση της δευτεροβάθμιας εξίσωσης			

Γ. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Δεδομένο ενός προβλήματος ονομάζεται ένα _____ ή αποδεκτό στοιχείο το οποίο χρησιμοποιείται ως _____ ή ή προϋπόθεση στην επίλυση προβλημάτων.
2. Πριν την επίλυση ενός προβλήματος, ελέγχεται η _____ των δεδομένων.
3. Οι τρεις βασικές μεθοδολογίες επίλυσης προβλημάτων που χρησιμοποιούνται είναι η _____, η _____, και η _____ .

Ασκήσεις - Προβλήματα

1. Να διατυπώσετε με αλγεβρικό τρόπο τα παρακάτω υπολογιστικά :
   α. Να υπολογιστεί το εμβαδό ενός τριγώνου, αν σας δίνεται ότι το εμβαδό του είναι το μισό του γινομένου της βάσης του επί το ύψος του. β. Η Κασσιανή είναι μεγαλύτερη της Άννα-Μαρίας κατά 10 χρόνια. Η Άννα-Μαρία είναι μικρότερη του Γιώργου κατά 3 χρόνια. Ο Κωνσταντίνος είναι 20 ετών και κατά 1 χρόνο μεγαλύτερος του Γιώργου. Πόσο χρονών είναι η Κασσιανή;
2. «Η Μάρθα θέλει να αγοράσει έναν ηλεκτρονικό υπολογιστή». Ποια είναι τα πιθανά ερωτήματα που θα πρέπει να απαντηθούν προκειμένου να προσδιοριστεί καλύτερα το ζητούμενο στο πρόβλημα της Μάρθας;»
3. Να λυθεί η εξίσωση: $ax^2+bx+c=0$. Αναλύστε σε υποπροβλήματα και παρουσιάστε σχηματικά το πρόβλημα.
4. «Ο Δημήτρης βρίσκεται στο λιμάνι του Πειραιά και θέλει να πάει στη Θεσσαλονίκη». Αναλύστε σε υποπροβλήματα και παρουσιάστε σχηματικά το πρόβλημα. Ποια μέθοδο χρησιμοποιήσατε και γιατί; Με βάση το είδος επίλυσης σε ποια ή ποιες κατηγορίες προβλημάτων θα το εντάσσατε;

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να περιγράφετε την έννοια του προβλήματος.
- να αναγνωρίζετε και να απαριθμείτε τις κατηγορίες προβλημάτων.
- να διακρίνετε την ύπαρξη υπολογιστικών προβλημάτων και να αναφέρετε τις φάσεις επίλυσής τους.
- να προσδιορίζετε τα δεδομένα και τα ζητούμενα ενός προβλήματος
- να ελέγχετε την πληρότητα, την ορθότητα και την σαφήνεια των δεδομένων και των ζητούμενων.
- να αναλύετε ένα πρόβλημα σε απλούστερα και να διατυπώνετε τα αποτελέσματα της ανάλυσης με τρόπο σαφή.

κεφάλαιο 3

Θέματα Θεωρητικής Επιστήμης των Υπολογιστών

Αλγόριθμοι

Στόχοι

Στόχοι του κεφαλαίου είναι οι μαθητές:

- να αναγνωρίζουν τη χρησιμότητα των αλγορίθμων στην επίλυση προβλημάτων.
- να μπορούν να περιγράφουν τα βασικά χαρακτηριστικά κάθε αλγόριθμου.
- να διακρίνουν τους διαφορετικούς τύπους αλγορίθμων.
- να συγκρίνουν τις διαφορετικές αναπαραστάσεις αλγορίθμων και να ξεχωρίζουν τα πλεονεκτήματα τους.

Λέξεις κλειδιά

Αλγόριθμος	Χαρακτηριστικά αλγορίθμου (Καθοριστικότητα, Περατότητα, Αποτελεσματικότητα, Είσοδος, Έξοδος), Ανάλυση Αλγορίθμων, Θεωρία Υπολογισμού, Υπολογισιμότητα, Πολυπλοκότητα αλγορίθμων, Σειριακή επεξεργασία, Παράλληλη επεξεργασία, Επαναληπτικοί και Αναδρομικοί αλγόριθμοι, Φυσική Γλώσσα, Διάγραμμα Ροής, Γλώσσες Περιγραφής Αλγορίθμων, Ψευδογλώσσα, Δομές δεδομένων, Εντολές αλγορίθμου, Δομές αλγορίθμων, Εκσφαλμάτωση, Λογικά λάθη, Συντακτικά λάθη, Τεκμηρίωση

Εισαγωγικές ερωτήσεις

1. Ποιος είναι ο αλγόριθμος που χρησιμοποιείτε καθημερινά στην προετοιμασία σας για το σχολείο;
2. Έχετε αναρωτηθεί από τι εξαρτάται η ταχύτητα εντοπισμού των στοιχείων ενός μαθητή σε ένα πρόγραμμα υπολογιστή;
3. Πώς περιγράφετε σε έναν φίλο σας τη λύση που δώσατε σε ένα πρόβλημα της καθημερινότητάς σας;

3. Αλγόριθμοι

3.1 Η έννοια του αλγορίθμου

Στο προηγούμενο κεφάλαιο είδαμε ότι για την επίλυση ενός προβλήματος μπορούν να ακολουθηθούν διάφορες τεχνικές οι οποίες έχουν κατηγοριοποιηθεί και συστηματοποιηθεί. Παρόλα αυτά, η διεργασία αντιμετώπισης προβλημάτων παραμένει μια υψηλά διαισθητική λειτουργία του ανθρώπινου εγκεφάλου. Αναρωτηθήκατε το πώς εκτελούνται αριθμητικές πράξεις στο μυαλό μας με τόση ευκολία; Το πώς είμαστε σίγουροι ότι το αποτέλεσμά τους είναι σωστό; Στο μυαλό μας, με θαυμαστό τρόπο, μέσα από τη γνώση, δημιουργούνται διαδικασίες επίλυσης προβλημάτων που καταγράφονται σε αυτό ως σωστές. Με την εμφάνιση των ηλεκτρονικών υπολογιστών, το στοίχημα του ανθρώπου ήταν να μεταδώσει αυτή τη θαυμαστή ιδιότητα του εγκεφάλου του στον υπολογιστή. Η διαδικασία επίλυσης ενός προβλήματος οδηγεί στη δημιουργία ενός αλγορίθμου.

Οι αλγόριθμοι στην καθημερινή ζωή της Ματίνας

- Βάλε ελαφριά ρούχα. Πήγαινε στο μπαλκόνι που έχει ησυχία. Κατέβασε την τέντα. Πάρε ένα χυμό να δροσιστείς και φυσικά το βιβλίο σου, είπε η μητέρα μου.

Όχι πάλι, σκέφτηκα.... Εντολές, εντολές, εντολές.... Δε μου φτάνει το πρόβλημα μου, έχω και τους αλγόριθμους της μητέρας μου! Ξεκίνησα το διάβασμα. Ορισμός: $\sqrt{x}$ είναι ο αριθμός y για τον οποίο ισχύει $y^2 = x$. Ωραία σκέφτηκα. Η ρίζα του 9 είναι το 3. Η ρίζα του 49 είναι το 7. Ναι, αλλά πώς θα υπολογίσω τη ρίζα του 2 ή τη ρίζα του 2371; Για τη λύση θα πρέπει να σκέφτομαι και να συνδυάζω ιδιότητες. Θα έχω όμως και τότε αποτέλεσμα; Αποτελεσματικότερο το κομπιουτεράκι της μητέρας μου. Εύκολο και σίγουρο. Πληκτρολόγησα το 2, πάτησα το σύμβολο $\sqrt{x}$ και να το αποτέλεσμα 1.41421356237309504880168872420 97 !!!!

Πώς το βρήκε τόσο γρήγορα; Πώς το σκέφτηκε; Γιατί εγώ πρέπει να διαβάζω, να συμπληρώνω φυλλάδια, να κάνω τόσο πολύπλοκους συνδυασμούς στο μυαλό μου και να μην μπορώ να υπολογίσω ΟΛΕΣ τις ρίζες, όπως το κομπιουτεράκι; Θα πρέπει, σκέφτηκα, να υπάρχει μια συγκεκριμένη γενική διαδικασία που αν την ακολουθήσω, θα μπορώ να υπολογίσω οποιαδήποτε ρίζα. Πρέπει να υπάρχει ένας αλγόριθμος!

Για επιβεβαίωση των σκέψεών μου έκανα αναζήτηση στον παγκόσμιο ιστό, στη μηχανή που λειτουργεί με τον εξυπνότερο και γρηγορότερο αλγόριθμο αναζήτησης. Φράση αναζήτησης: «Αλγόριθμος για τον υπολογισμό της τετραγωνικής ρίζας ενός αριθμού» και πρώτο αποτέλεσμα «Η προσεγγιστική μέθοδος του Newton για τον υπολογισμό της τετραγωνικής ρίζας αριθμού».

Αν y μια τιμή κατά προσέγγιση της $\sqrt{x}$, τότε το αποτέλεσμα της πράξης $\frac{y+\frac{x}{y}}{2}$ δίνει μια καλύτερη προσέγγιση, πιο κοντά στην πραγματική τιμή της ρίζας.

Άρχισα να το εφαρμόζω, ελπίζοντας να βρω την τιμή που μου έβγαλε το κομπιουτεράκι.

Πρώτη φορά $\frac{2+\frac{2}{2}}{2}=1.5$, δεύτερη φορά $\frac{1.5+\frac{2}{1.5}}{2}=1.4166$, τρίτη φορά... $\frac{1.4166+\frac{2}{1.4166}}{2}=1.41421568$

Το εφάρμοσα μόλις τρεις φορές και παρατήρησα ότι το αποτέλεσμα σύγκλινε με την τιμή που είχε βγάλει το κομπιουτεράκι. Είχα ανακαλύψει μέσα από την παρατήρηση και την έρευνα την έννοια του αλγορίθμου, μια μηχανικά εκτελέσιμη υπολογιστική διαδικασία, μέσω της οποίας, αν ξέρεις πράξεις και την ακολουθήσεις πιστά, υπολογίζεις μια ποσότητα. Με αυτό τον τρόπο θα μπορούσα να υπολογίσω οποιανδήποτε τετραγωνική ρίζα.

Η έννοια του αλγορίθμου είναι θεμελιώδης για την επιστήμη της Πληροφορικής. Στα Μαθηματικά δίνουμε ορισμούς που περιγράφουν το «τι είναι». Διατυπώνουμε γενικές ιδιότητες και θεωρήματα. Συνδυάζοντάς τα, λύνουμε ειδικά προβλήματα (π.χ. τον υπολογισμό της ρίζας του 2). Στην Πληροφορική κατασκευάζουμε αλγορίθμους που περιγράφουν το «πώς υπολογίζεται», δηλαδή δημιουργούμε συγκεκριμένες υπολογιστικές διαδικασίες οι οποίες εκτελούνται μηχανικά. Στη συνέχεια, τις εφαρμόζουμε για να λύνουμε γενικά προβλήματα (π.χ. υπολογισμός της ρίζας οποιουδήποτε αριθμού).

Αλγόριθμος είναι η ακριβής περιγραφή μιας σειράς βημάτων που απαιτούνται για την επίλυση ενός προβλήματος.

Ο αλγόριθμος αποτελεί τη βάση για τη δημιουργία ενός προγράμματος που θα εκτελεστεί από έναν ηλεκτρονικό υπολογιστή. Η έννοιά του είναι γενικότερη από εκείνη του προγράμματος και δε συνδέεται αποκλειστικά με την Πληροφορική. Μπορεί να εκφραστεί ακόμα και σε φυσική γλώσσα, Ελληνικά ή Αγγλικά.

Επίλυση προβλήματος με ΗΥ

Λύτης = Προγραμματιστής

Εκτελεστής = Υπολογιστής

Χρήστης = Χρήστης Η/Υ

Μια συνταγή μαγειρικής είναι ένας αλγόριθμος. Οι οδηγίες του καθηγητή για την επίλυση της δευτεροβάθμιας εξίσωσης είναι, επίσης, αλγόριθμος. Στην επίλυση προβλημάτων μέσω αλγορίθμων, υπάρχουν τρεις (3) διαφορετικοί διακριτοί ρόλοι:

- Ο λύτης, αυτός που καλείται να αντιμετωπίσει το πρόβλημα σχεδιάζοντας τον αλγόριθμο.
- Ο εκτελεστής, αυτός που εφαρμόζει πιστά τις εντολές του αλγορίθμου που έφτιαξε ο λύτης.
- Ο χρήστης, αυτός που ενεργοποιεί τον αλγόριθμο, καλώντας τον εκτελεστή να λύσει, όποτε θέλει, το πρόβλημα.

Ερωτήσεις - Θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Η έννοια του αλγόριθμου είναι θεμελιώδης για την επιστήμη της Πληροφορικής.
2. Η έννοια του αλγόριθμου είναι γενικότερη από εκείνη του προγράμματος.
3. Η έννοια του αλγόριθμου συνδέεται αποκλειστικά με την Πληροφορική.
4. Μια συνταγή μαγειρικής είναι ένας αλγόριθμος
5. Στην επίλυση προβλημάτων μέσω αλγορίθμων, υπάρχουν δύο διαφορετικοί διακριτοί ρόλοι

Β. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Ο αλγόριθμος αποτελεί τη βάση για τη δημιουργία ενός

_____ .

2. Οι διακριτοί ρόλοι στην επίλυση προβλημάτων μέσω αλγορίθμων είναι ο _____, ο _____ και ο_____ .

Γ. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με την κατάλληλη έννοια της στήλης Β

1.

Στήλη Α	Στήλη Β
α. Μαθηματικά	1. ορισμοί
	2. ιδιότητες
	3. θεωρήματα
β. Πληροφορική	4. υπολογιστικές διαδικασίες
	5. αλγόριθμοι

2. Επίλυση προβλήματος με υπολογιστή

Στήλη Α	Στήλη Β
α. Λύτης	1. Χρήστης Η/Υ
β. Εκτελεστής	2. Προγραμματιστής
γ. Χρήστης	3. Υπολογιστής

Ασκήσεις - Προβλήματα

Αναζητήσετε τον αλγόριθμο του Ευκλείδη για τον υπολογισμό του Μέγιστου Κοινού Διαιρέτη. Μελετήστε τη λειτουργία του και εφαρμόστε τον για τον υπολογισμό του ΜΚΔ των αριθμών 32 και 40.

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να ορίζετε και να περιγράφετε την έννοια του αλγόριθμου
- να προσδιορίζετε τη θέση του αλγόριθμου στην καθημερινή ζωή
- να αναγνωρίζετε τη σημασία της αλγοριθμικής σκέψης στην επίλυση προβλημάτων και τη λήψη αποφάσεων
- να δίνετε παραδείγματα αλγορίθμων είτε για απλά καθημερινά προβλήματα είτε για απλά υπολογιστικά προβλήματα.
- να αναγνωρίζετε τους ρόλους του λύτη, του εκτελεστή και του χρήστη στη δημιουργία ενός αλγορίθμου.

3.2 Χαρακτηριστικά αλγορίθμου

Ο αλγόριθμος αποτελείται από μια σειρά εντολών που, αν εφαρμοσθούν πιστά, οδηγούν στην επίλυση ενός προβλήματος. Είναι, όμως, οποιαδήποτε σειρά από εντολές ένας αλγόριθμος; Η απάντηση είναι όχι. Οι εντολές ενός αλγορίθμου, αλλά και ο αλγόριθμος ως ολότητα, θα πρέπει να ικανοποιούν κάποια κριτήρια:

- **Είσοδος**: είναι τα στοιχεία που χρειάζεται ο αλγόριθμος για να εκτελεσθεί.
- **Έξοδος**: είναι τα στοιχεία που παράγει ο αλγόριθμος, τα αποτελέσματά του.
- **Καθοριστικότητα**: κάθε εντολή θα πρέπει να είναι μονοσήμαντη, δηλαδή να καθορίζει με απόλυτη σαφήνεια και ακρίβεια τον τρόπο εκτέλεσής της σε κάθε δυνατή περίπτωση.
- **Περατότητα**: εκτελώντας τα βήματα του αλγορίθμου, θα πρέπει να φθάνουμε σε πέρας (τέλος) σε κάθε δυνατή περίπτωση.
- **Αποτελεσματικότητα**: κάθε εντολή θα πρέπει να είναι διατυπωμένη με απλό τρόπο, ώστε να μπορεί να εκτελεσθεί.

Ας εξετάσουμε τα αλγοριθμικά κριτήρια μέσα από ένα πρόβλημα.

Πρόβλημα: πώς φτιάχνουμε ένα κέικ σοκολάτας;

Αλγόριθμος: Αρκεί να ανοίξουμε ένα βιβλίο ζαχαροπλαστικής και να ακολουθήσουμε πιστά οποιαδήποτε συνταγή.

Είσοδος: τα υλικά **Έξοδος**: το ζεστό ζουμερό κέικ

Καθοριστικότητα: η εντολή «Ψήσε σε φούρνο ήδη προθερμασμένο σε θερμοκρασία 175 βαθμούς Κελσίου για ακριβώς 45 λεπτά» είναι απόλυτα σαφής, ενώ η εντολή «Ψήσε σε φούρνο μέχρι να φουσκώσει» σίγουρα δεν είναι!

Περατότητα: Μια εντολή επανάληψης «Ψήσε στο φούρνο μέχρι να σταματήσει η γη να κινείται» θα ήταν καταστροφική για τον αλγόριθμο και το κέικ. Ο αλγόριθμος δε θα τελειώσει ποτέ και το κέικ θα καιγόταν.

Αποτελεσματικότητα: Η εντολή «Άνοιξε το φούρνο και πιάσε το καυτό κέικ με τα δύο σου γυμνά χέρια», αν και απόλυτα σαφής, δε θα ήταν καταστροφική για το κέικ αλλά θα ήταν για τα χέρια μας και κατά συνέπεια για τον αλγόριθμο!

Είναι φανερό από το παραπάνω παράδειγμα ότι τα κριτήρια της καθοριστικότητας και αποτελεσματικότητας είναι σχετικά με αυτόν που θα κληθεί να εφαρμόσει τον αλγόριθμο. Κάθε εκτελεστής έχει διαφορετικές δυνατότητες, διαφορετικά δυνατά και αδύνατα σημεία. Ο άνθρωπος, για παράδειγμα, μπορεί να ακούσει, να διαβάσει, μπορεί να μιλήσει αλλά ακόμα και να εφαρμόσει τύπους, να σκεφθεί, να αυτοσχεδιάσει. Ο υπολογιστής, από την άλλη, μπορεί να εισάγει από το πληκτρολόγιο, να εμφανίσει, να κάνει αριθμητικές και λογικές πράξεις. Κάθε εκτελεστής έχει διαφορετικό σύνολο κατανοητών και εφικτών εντολών. Ο άνθρωπος έχει ένα ευρύ σύνολο εκτελέσιμων εντολών. Ο υπολογιστής μπορεί να εκτελεί μόνο βασικές πράξεις. Σε κάθε περίπτωση, μια σειρά εντολών θα πρέπει να λαμβάνει υπόψη το μελλοντικό εκτελεστή της, προκειμένου να ικανοποιεί τα αλγοριθμικά κριτήρια.

Ερωτήσεις - Θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Μια σειρά από εντολές είναι ένας αλγόριθμος.
2. Οι εντολές ενός αλγορίθμου πρέπει να ικανοποιούν κάποια κριτήρια.
3. Ένας αλγόριθμος είναι μια σειρά εντολών.
4. Ο υπολογιστής μπορεί να εκτελεί μόνο βασικές πράξεις.
5. Μια σειρά εντολών πρέπει να λαμβάνει υπόψη το μελλοντικό εκτελεστή της, προκειμένου να ικανοποιεί τα αλγοριθμικά κριτήρια.
6. Ο άνθρωπος έχει ευρύτερο σύνολο εκτελέσιμων εντολών από τον υπολογιστή.

Β. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με την κατάλληλη έννοια της στήλης Β:

Στήλη Α	Στήλη Β
Είσοδος	κάθε εντολή πρέπει να είναι απλά διατυπωμένη και εκτελέσιμη
Έξοδος	δεδομένα
Καθοριστικότητα	ο αλγόριθμος φθάνει πάντα σε τέλος
Περατότητα	κάθε εντολή πρέπει να είναι μονοσήμαντη
Αποτελεσματικότητα	αποτελέσματα

Γ. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Ο αλγόριθμος αποτελείται από μια σειρά _____ που, αν εφαρμοσθούν πιστά, οδηγούν στην _____ ενός προβλήματος.
2. Τα κριτήρια της _____ και _____ είναι σχετικά με αυτόν που θα κληθεί να εφαρμόσει τον αλγόριθμο.

Δ. Ερωτήσεις

1. Δίνεται η εντολή «Πολλαπλασίασε την ακτίνα επί 2 και επί 3.14» σε φυσική γλώσσα. Έχει καθοριστικότητα; Έχει αποτελεσματικότητα;
2. Ποιος έχει ευρύτερο σύνολο εκτελέσιμων εντολών, ο άνθρωπος ή ο υπολογιστής;
3. Συζητήσετε στην τάξη το πώς επηρεάζει ο εκτελεστής (άνθρωπος ή υπολογιστής) την αποτελεσματικότητα ενός αλγορίθμου.

Ασκήσεις - Προβλήματα

1. Περιγράψτε με φυσική γλώσσα αλγόριθμο υπολογισμού της περιφέρειας και του εμβαδού ενός κύκλου. Δώστε τον σε συμμαθητή σας να τον εκτελέσει και συζητήστε στην τάξη αν ικανοποιεί τα αλγοριθμικά κριτήρια.

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να αναγνωρίζετε πότε μια σειρά από εντολές αποτελεί αλγόριθμο
- να αναγνωρίζετε τη σημασία του να τερματίζει και να δίνει αποτελέσματα ένας αλγόριθμος.
- να αναγνωρίζετε το ρόλο του λύτη και του εκτελεστή στην αποτελεσματικότητα ενός αλγορίθμου.

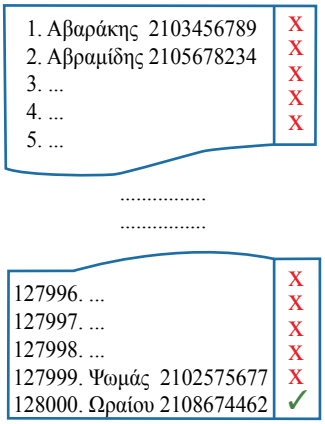
3.3 Ανάλυση αλγορίθμων

Οι αλγόριθμοι προκύπτουν από την ανάγκη εύρεσης λύσεων σε προβλήματα που μας απασχολούν και φυσικά μπορούμε να έχουμε πολλές λύσεις για το ίδιο πρόβλημα άρα και πολλούς αλγόριθμους. Πώς όμως μπορούμε να αξιολογήσουμε τους διαφορετικούς αλγόριθμους και να εντοπίσουμε ποιος είναι καλύτερος έναντι κάποιου άλλου; Υπάρχουν στοιχεία σε έναν αλγόριθμο που να μπορούν να μετρηθούν και να υπολογιστεί πόσο αποδοτικός είναι; Στα ερωτήματα αυτά προσπαθεί να απαντήσει ο τομέας της επιστήμης των υπολογιστών που ασχολείται με την **ανάλυση αλγορίθμων**.

Τα δύο στοιχεία ενός αλγορίθμου που μπορούν να μετρηθούν και να μας δώσουν μια εκτίμηση της αποδοτικότητάς του είναι ο χρόνος που χρειάζεται για να εκτελεστεί και ο χώρος, δηλαδή η μνήμη, που απαιτείται για να λειτουργήσει σωστά. Μεγαλύτερη έμφαση δίνουμε στη σωστή χρήση του χρόνου και όχι του χώρου, καθώς το κόστος των κυκλωμάτων μνήμης των υπολογιστών μειώνεται συνεχώς με την εξέλιξη της τεχνολογίας.

Η εκτίμηση του χρόνου εκτέλεσης ενός αλγορίθμου μπορεί να γίνει με δύο τρόπους: α) πειραματικά, χρονομετρώντας το διάστημα από τη στιγμή που ξεκινά να εκτελείται ένας αλγόριθμος μέχρι τη στιγμή που θα τερματίσει, β) θεωρητικά, προσπαθώντας να βρούμε μια μαθηματική σχέση που να συνδέει το χρόνο εκτέλεσης με το πλήθος των δεδομένων εισόδου. Επειδή οι αλγόριθμοι μπορούν να εφαρμόζονται σε διαφορετικό πλήθος δεδομένων κάθε φορά και φυσικά σε μηχανήματα διαφορετικών δυνατοτήτων, η πειραματική προσέγγιση δε μας δίνει πολλά στοιχεία για την αποδοτικότητα του αλγορίθμου. Από την άλλη πλευρά, η θεωρητική προσέγγιση της εύρεσης μιας μαθηματικής σχέσης μας καλύπτει καλύτερα διότι έτσι μπορούμε να εκτιμήσουμε την αποδοτικότητα ενός αλγορίθμου ανεξάρτητα από το πλήθος των δεδομένων εισόδου ή της υπολογιστικής μηχανής που χρησιμοποιούμε.

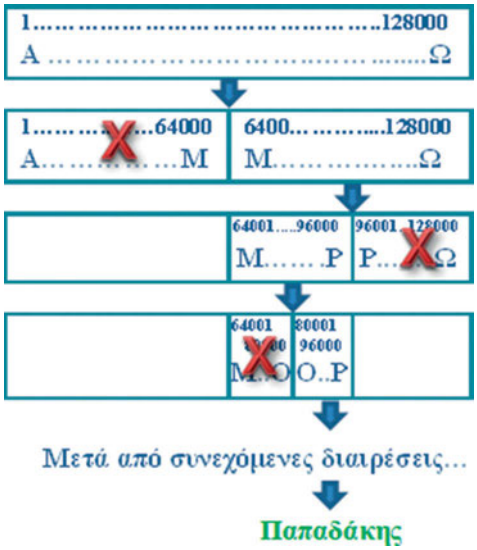
Ας δούμε όμως καλύτερα ένα παράδειγμα. Έστω ότι έχουμε έναν τηλεφωνικό κατάλογο που περιλαμβάνει 128000 ονόματα και τηλέφωνα. Θέλουμε να φτιάξουμε έναν αλγόριθμο που, εάν του δίνουμε ένα όνομα ως είσοδο, αυτός να μας επιστρέφει το τηλέφωνο που αντιστοιχεί σε αυτό. Μια πρώτη λύση στο πρόβλημά μας είναι να αρχίσουμε να διαβάζουμε από την αρχή όλα τα ονόματα του τηλεφωνικού καταλόγου και μόλις βρούμε το όνομα που μας ενδιαφέρει να επιστραφεί το τηλέφωνό του. Η λύση αυτή είναι ιδιαίτερα χρονοβόρα, χρησιμοποιείται όμως στους υπολογιστές και ονομάζεται **σειριακή αναζήτηση**.



Εικόνα 3-1. Σειριακή αναζήτηση του ονόματος «Ωραίου»

Προσπαθώντας να αναλύσουμε τον αλγόριθμό μας παρατηρούμε ότι εάν το όνομα που δίνουμε προς αναζήτηση βρίσκεται στις πρώτες σελίδες του τηλεφωνικού καταλόγου, ο αλγόριθμός μας βρίσκει το αντίστοιχο τηλέφωνο γρήγορα, ενώ, εάν το όνομα που δίνουμε βρίσκεται στις τελευταίες σελίδες καθυστερεί σημαντικά. Εάν συμφωνήσουμε ότι ο χρόνος ανάγνωσης και σύγκρισης ενός ονόματος με αυτό που ψάχνουμε είναι ίσος με ένα δευτερόλεπτο, τότε ο χρόνος εκτέλεσης του αλορίθμου θα ισούται με τη θέση του ονόματος στον τηλεφωνικό κατά λογο. Έτσι, εάν αναζητούμε το 10ο στη σειρά

όνομα, ο χρόνος εκτέλεσης θα είναι 10 δευτερόλεπτα. Αυτό που μας ενδιαφέρει συνήθως σε έναν αλγόριθμο είναι ο χρόνος της χειρότερης περίπτωσης που στο συγκεκριμένο παράδειγμα είναι 128000 δευτερόλεπτα και αντιστοιχεί στην αναζήτηση του τελευταίου ονόματος του τηλεφωνικού καταλόγου. Γενικεύοντας, αν ο τηλεφωνικός μας κατάλογος είχε n ονόματα, τότε ο χρόνος χειρότερης περίπτωσης θα συμβολίζονταν με O(n) θέλοντας να δείξουμε τη γραμμική σχέση ανάμεσα στο πλήθος των δεδομένων εισόδου και του χρόνου εκτέλεσης του αλγορίθμου. Ο συμβολισμός O ( ) λοιπόν μας δίνει ένα πάνω όριο για τον χρόνο εκτέλεσης ενός αλγορίθμου και αποτελεί ένα πολύ καλό μέτρο της αποδοτικότητάς του.



Εικόνα 3-2. Δυαδική Αναζήτηση του ονόματος «Παπαδάκης»

Οι περισσότεροι αλγόριθμοι έχουν χρονική πολυπλοκότητα που ανήκει σε μια από τις κατηγορίες: O(1), O(logn), O(n), O(nlogn), $O(n^2)$, $O(2^n)$

Μια δεύτερη λύση στο ίδιο πρόβλημα εκμεταλλεύεται την ιδιότητα της αλφαβητικής ταξινόμησης των ονομάτων του τηλεφωνικού καταλόγου. Όταν μας δίνεται ένα όνομα προς αναζήτηση, τότε το συγκρίνουμε με το όνομα που βρίσκεται στη μέση του τηλεφωνικού καταλόγου. Εάν το όνομα που ψάχνουμε προηγείται αλφαβητικά του μεσαίου ονόματος, τότε απορρίπτουμε αμέσως όλα τα ονόματα που βρίσκονται μετά τη μέση του τηλεφωνικού καταλόγου, ενώ, εάν το όνομα που ψάχνουμε έπεται αλφαβητικά του μεσαίου ονόματος, τότε απορρίπτουμε όλα τα ονόματα που βρίσκονται πριν τη μέση του τηλεφωνικού καταλόγου. Με μία και μόνη σύγκριση έχουμε «ελέγξει» τα μισά ονόματα! Στη συνέχεια εφαρμόζουμε την τακτική μας στα 64000 πρώτα ή στα 64000 τελευταία ονόματα που έμειναν, συγκρίνοντας πάλι το μεσαίο όνομα με αυτό που αναζητούμε. Μετά τη νέα σύγκριση θα μας μείνουν προς έλεγχο 32000 ονόματα και συνεχίζουμε με την ίδια λογική μέχρι να βρούμε τελικά το όνομα που ψάχνουμε. Τα βήματα που θα εκτελέσει ο αλγόριθμός μας, στη χειρότερη περίπτωση, είναι μόλις 17! Ο αριθμός αυτός μπορεί να προκύψει μαθηματικά υπολογίζοντας τον δυαδικό λογάριθμο του 128000 ($\log_2 128000$) καθώς ο αλγόριθμός μας βασίστηκε σε συνεχόμενες διαιρέσεις του πλήθους των δεδομένων

του προβλήματός μας με το 2. Στη γενική περίπτωση λοιπόν που ο αριθμός των ονομάτων του τηλεφωνικού καταλόγου ισούται με n, ο χρόνος χειρότερης περίπτωσης θα συμβολιζόταν με O(log n) (με log n συμβολίζουμε χάριν συντομίας τον δυαδικό λογάριθμο του n) . Ο αλγόριθμος που μόλις περιγράψαμε ονομάζεται δυαδική αναζήτηση και αποτελεί τον γρηγορότερο αλγόριθμο αναζήτησης, όταν τα δεδομένα μας είναι ταξινομημένα.

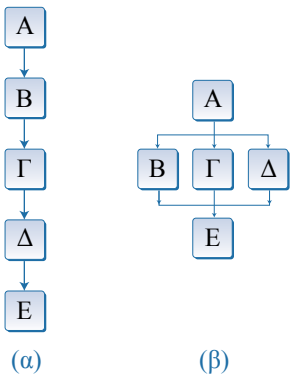
	Πλήθος δεδομένων προβλήματος				
Πολυπλοκότητα αλγορίθμου	**10**	**100**	**1000**	**10000**	**100000**
O(logn)	3,3 ns	6,6 ns	10 ns	13,3 ns	16,6 ns
O(n)	10 ns	100 ns	1000 ns	10000 ns	100000 ns
O(n^2)	100 ns	10000 ns	0,001 sec	0,1 sec	10 sec
O(n^3)	1000 ns	0,001 sec	1 sec	100 sec	278 ώρες
O(2^n)	1024 ns	401969368413 αιώνες			

Εικόνα 3-3. Χρόνοι εκτέλεσης αλγορίθμων με χρόνο εκτέλεσης εντολής 1 ns

Η **πολυπλοκότητα αλγορίθμων** είναι ο ένας από τους δύο κλάδους του τομέα της επιστήμης των υπολογιστών που ονομάζεται **θεωρία υπολογισμού**. Ο τομέας αυτός εξετάζει αρχικά εάν μπορεί να λυθεί ένα πρόβλημα χρησιμοποιώντας κάποιο αλγόριθμο και, σε δεύτερη φάση, πόσο αποδοτικά μπορεί να λυθεί το συγκεκριμένο πρόβλημα. Ο κλάδος που αναζητά τι μπορεί να υπολογισθεί και τι όχι ονομάζεται **υπολογισιμότητα**. Για την απόδειξη της υπολογισιμότητας ή όχι κάποιου προβλήματος οι επιστήμονες της πληροφορικής χρησιμοποιούν διάφορα υπολογιστικά μοντέλα με το ισχυρότερο από αυτά να είναι η λεγόμενη **Μηχανή Τιούρινγκ** από το όνομα του μεγάλου βρετανού μαθηματικού Άλαν Τιούρινγκ που θεωρείται ο πατέρας της επιστήμης των υπολογιστών.

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να περιγράφετε τι είναι η αποδοτικότητα ενός αλγορίθμου
- να αναγνωρίζετε και να συσχετίζετε τις έννοιες «πολυπλοκότητα» και «υπολογισιμότητα» αλγορίθμου στην επίλυση ενός προβλήματος



Εικόνα 3-4. Σειριακή και παράλληλη εκτέλεση αλγορίθμου

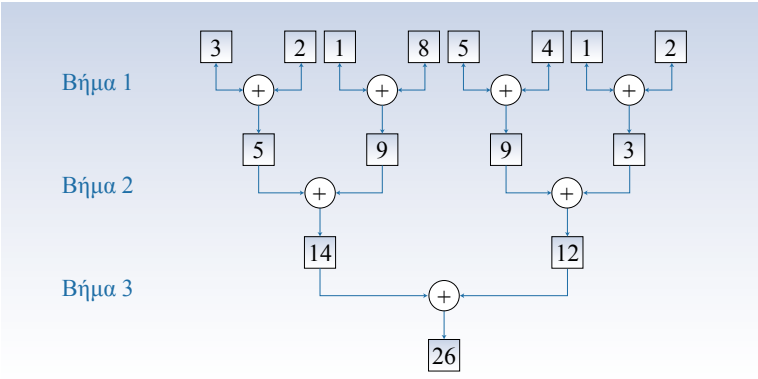
3.4 Βασικοί τύποι αλγορίθμων

Αλγόριθμοι σειριακής και παράλληλης επεξεργασίας

Οι εντολές ενός αλγορίθμου γράφονται η μια κάτω από την άλλη και εκτελούνται ακολουθιακά. Σε κάποιους αλγόριθμους, τμήματα εντολών μπορούν να εκτελεσθούν ταυτόχρονα. Ας θεωρήσουμε έναν αλγόριθμο που αποτελείται από 5 τμήματα, Α έως Ε. Στην Εικόνα 3-4 (α), βλέπουμε την εκτέλεση των τμημάτων του αλγορίθμου με σειριακή επεξεργασία. Αν τα τμήματα Β, Γ, Δ δεν εξαρτώνται το ένα από το άλλο, τότε μπορούν να εκτελεσθούν ταυτόχρονα, μειώνοντας δραστικά το χρόνο εκτέλεσης. Στην Εικόνα 3-4 (β), βλέπουμε την εκτέλεση των τμημάτων του αλγορίθμου με παράλληλη επεξεργασία.

Για παράδειγμα, μπορούμε να εφαρμόσουμε τη σειριακή και παράλληλη επεξεργασία στον υπολογισμό του αθροίσματος 8 τυχαίων αριθμών. Ο αλγόριθμος σειριακής επεξεργασίας απαιτεί 8 βήματα εκτέλεσης: Στην Εικόνα 3-5 βλέπουμε την εφαρμογή ενός αλγορίθμου παράλληλης επεξεργασίας για την επίλυση του ίδιου προβλήματος. Η πρόσθεση του πρώτου με τον δεύτερο, του τρίτου με τον τέταρτο, του πέμπτου με τον έκτο και του έβδομου με τον όγδοο γίνονται ταυτόχρονα σε ένα βήμα.

Αν θέλουμε να προσθέσουμε, χίλιους (1000) αριθμούς πόσα βήματα απαιτούνται με παράλληλη επεξεργασία; Πώς επηρεάζεται η απάντηση από το πλήθος των διαθέσιμων επεξεργαστών.



Εικόνα 3-5. Παράλληλη εκτέλεση αλγόριθμου

Ο αλγόριθμος σειριακής επεξεργασίας απαιτεί επτά (7) βήματα, ενώ ο αλγόριθμος παράλληλης επεξεργασίας, μόλις τρία (3).

Βήμα 1:	Πρόσθεσε τον πρώτο με το δεύτερο αριθμό.
Βήμα 2:	Πρόσθεσε το αποτέλεσμα με τον τρίτο αριθμό.
	...
Βήμα 7:	Πρόσθεσε το αποτέλεσμα με τον όγδοο αριθμό.

Αλγόριθμος σειριακής επεξεργασίας είναι ένας αλγόριθμος του οποίου τα βήματα εκτελούνται ακολουθιακά το ένα μετά το άλλο, από έναν επεξεργαστή.

Αλγόριθμος παράλληλης επεξεργασίας είναι ένας αλγόριθμος του οποίου τα βήματα μπορούν να εκτελούνται ταυτόχρονα από διαφορετικούς επεξεργαστές, μειώνοντας το χρόνο εκτέλεσης.

Επαναληπτικοί αλγόριθμοι

Ένα από τα μεγάλα πλεονεκτήματα της χρήσης των υπολογιστών για την επίλυση προβλημάτων είναι η δυνατότητα επανάληψης μιας διαδικασίας πολλές φορές. Ο άνθρωπος, έστω και αν μπορεί να εκτελέσει μια ενέργεια, κουράζεται με την επανάληψή της, κάνει λάθη. Δεν ισχύει το ίδιο για τον υπολογιστή. Γι' αυτό και οι επαναληπτικοί αλγόριθμοι είναι πολύ διαδεδομένοι στους υπολογιστές.

Ο υπολογισμός του αθροίσματος δύο αριθμών είναι μια απλή πρόσθεση. Όταν, όμως, έχουμε χίλιους αριθμούς, τότε η πράξη της πρόσθεσης πρέπει να εκτελεστεί χίλιες φορές. Αυτή είναι μια επαναληπτική διαδικασία, η οποία μπορεί να εκφρασθεί ως εξής:

Σε τρεις γραμμές ο αλγόριθμος έχει ολοκληρωθεί. Ο βρόχος μείωσε δραστικά το πλήθος των απαιτούμενων εντολών χωρίς να επηρεάσει την αποτελεσματικότητα του αλγορίθμου.

Μια επαναληπτική διαδικασία λέγεται **βρόχος**.

Ένας βρόχος που δεν σταματά ποτέ λέγεται **ατέρμων**.

Αναδρομικοί αλγόριθμοι

Η αναδρομή είναι μια ιδιαίτερα διαισθητική αλγοριθμική μέθοδος. Η κατανόηση των αναδρομικών αλγορίθμων δεν είναι εύκολη υπόθεση. Απαιτεί ιδιαίτερο τρόπο σκέψης που κατακτάται μόνο με έντονη εξάσκηση. Ένας αναδρομικός αλγόριθμος χρησιμοποιεί τον ίδιο του τον εαυτό.

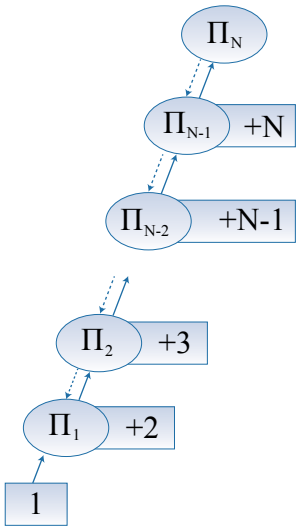
Αναδρομικός είναι ο αλγόριθμος που καλεί άμεσα ή έμμεσα τον εαυτό του μία ή περισσότερες φορές, επιλύοντας κάθε φορά ένα πρόβλημα της ίδιας φύσης με το αρχικό, αλλά μικρότερου μεγέθους.

Αν συμβολίσω το γινόμενο 1*2*...*N με $Π_N$, τότε προκύπτει ο εξής μαθηματικός αναδρομικός ορισμός:

ΠΝ = 1 , αν N=1

ΠΝ = ΠΝ-1 * N, αν N>1

με τη χρήση του οποίου μπορώ να υπολογίσω αναδρομικά οποιοδήποτε γινόμενο $Π_N$.



Εικόνα 3-6. Αναδρομικός αλγόριθμος

Για παράδειγμα, ο υπολογισμός του γινομένου Π = 1*2*3*.. .*N μπορεί να γίνει αναδρομικά. Ας προσπαθήσουμε να λύσουμε ένα στιγμιότυπο του προβλήματος, τον υπολογισμό του 1*2*...*6 και μετά να γενικεύσουμε τη λύση, αντικαθιστώντας το 6 με N. Αναπτύσσουμε το εξής σκεπτικό: Για να υπολογίσω το γινόμενο 1*2*3*...*6, αρκεί να πολλαπλασιάσω τον αριθμό 6 με το γινόμενο 1 *2* .*5. Για να υπολογίσω το γινόμενο 1 *2*3*.*5, αρκεί να πολλαπλασιάσω τον αριθμό 5 με το γινόμενο 1 *2* ...*4. Συνεχίζοντας την απλούστευση του προβλήματος, θα φθάσω στο πρόβλημα υπολογισμού του γινομένου 1*2 που είναι τετριμμένο. Ακολουθώντας, τώρα, την αντίθετη φορά, έχοντας υπολογίσει το 1 *2, υπολογίζω το (1*2)*3, μετά το ((1*2)*3)*4, κ.ο.κ., μέχρι να φθάσω στο ((1*2)*.. ,)*6. Μπορώ να γενικεύσω το παραπάνω σκεπτικό για οποιοδήποτε ακέραιο N.

Η δημιουργία και ο έλεγχος ορθότητας αναδρομικών αλγορίθμων είναι ακόμα πιο δύσκολες διεργασίες που απαιτούν προγραμματιστική εμπειρία. Παρόλ' αυτά, η αναδρομή είναι πολύ ισχυρή μέθοδος, που δίνει τη δυνατότητα να περιγράψουμε με σύντομο τρόπο τη λύση δύσκολων προβλημάτων.

Ερωτήσεις - Θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Σε κάποιους αλγόριθμους, τμήματα εντολών μπορούν να εκτελεσθούν ταυτόχρονα.
2. Η εκτέλεση τμημάτων ενός αλγόριθμου με παράλληλη επεξεργασία, μειώνει δραστικά το χρόνο εκτέλεσης.
3. Η αξιοποίηση της παραλληλίας μπορεί να εφαρμοσθεί σε προβλήματα που απαιτούν ομοιόμορφη επεξεργασία πολλών δεδομένων.
4. Η παράλληλη επεξεργασία μπορεί να εφαρμοσθεί σε όλα τα προβλήματα.

Β. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Αλγόριθμος σειριακής επεξεργασίας είναι ένας αλγόριθμος του οποίου τα βήματα εκτελούνται _____ το ένα μετά το άλλο, από έναν _____ .
2. Αλγόριθμος παράλληλης επεξεργασίας είναι ένας αλγόριθμος του οποίου τα βήματα μπορούν να εκτελούνται _____ από διαφορετικούς _____ , μειώνοντας το χρόνο εκτέλεσης.
3. Για τον υπολογισμό του αθροίσματος 8 τυχαίων αριθμών, ο αλγόριθμος σειριακής επεξεργασίας απαιτεί _____ βήματα, ενώ ο αλγόριθμος παράλληλης επεξεργασίας, μόλις _____.

Ασκήσεις - Προβλήματα

1. Αναλύσετε τον υπολογισμό της αριθμητικής παράστασης ((x-y)*(x+y))/((2+x)*y) σε απλές εντολές μιας πράξης. Ποιες εντολές μπορούν να εκτελεστούν ταυτόχρονα; Πόσα βήματα απαιτούνται για τη σειριακή εκτέλεση από έναν επεξεργαστή; Πόσα βήματα απαιτούνται κατ' ελάχιστο για την παράλληλη επεξεργασία από τρεις επεξεργαστές;
2. Αναζητήσετε το «Πρόβλημα των πύργων του Ανόι». Επιλύσετε το πρόβλημα για 3 δίσκους και εκφράστε τη λύση του με φυσική γλώσσα. Χρησιμοποιήστε τη λύση που κατασκευάσατε, για να λύσετε το πρόβλημα για 4 δίσκους. Γενικεύσετε τη λύση για οποιοδήποτε πλήθος δίσκων. Εκφράσετε τη λύση σας αναδρομικά.

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να αναγνωρίζετε την ύπαρξη συγκεκριμένων χαρακτηριστικών και τύπων αλγορίθμων
- να διακρίνετε έναν αλγόριθμο σειριακής επεξεργασίας από έναν αλγόριθμο παράλληλης επεξεργασίας
- να αναγνωρίζετε τις αλληλοεξαρτήσεις μεταξύ των εντολών ενός σειριακού αλγορίθμου και να εντοπίζετε ποια βήματα ενός αλγόριθμου που μπορούν να εκτελεσθούν παράλληλα.
- να προσδιορίζετε την έννοια της επαναληπτικής διαδικασίας και του βρόχου
- να αναγνωρίζετε πότε ένα πρόβλημα απαιτεί επαναληπτικό αλγόριθμο για την επίλυσή του.
- να διατυπώνετε σε φυσική γλώσσα έναν επαναληπτικό αλγόριθμο.
- να αναγνωρίζετε πότε ένας αλγόριθμος είναι αναδρομικός.

3.5 Αναπαράσταση αλγορίθμου

Η αναπαράσταση ενός αλγορίθμου μπορεί να γίνει με διάφορους τρόπους. Ο τρόπος περιγραφής σε συνδυασμό με αυτόν που θα κληθεί να τον εκτελέσει, επηρεάζουν την ικανοποίηση ή μη των αλγοριθμικών κριτηρίων. Επιπλέον, ο ίδιος αλγόριθμος, αν εκφρασθεί με διαφορετικούς τρόπους, μπορεί να ικανοποιεί ή όχι κάποια αλγοριθμικά κριτήρια. Για την αναπαράσταση των αλγορίθμων χρησιμοποιούνται διάφοροι τρόποι όπως η φυσική γλώσσα, το διάγραμμα ροής, οι γλώσσες περιγραφής αλγορίθμων και οι γλώσσες προγραμματισμού.

Η **φυσική γλώσσα** αποτελεί τον πιο απλό και ανεπεξέργαστο τρόπο παρουσίασης ενός αλγορίθμου, που με απλά λόγια και ελεύθερες εκφράσεις περιγράφουμε τα βήματα. Ωστόσο, ο συγκεκριμένος τρόπος έκφρασης ενέχει αυξημένη πιθανότητα λάθους ή ασάφειας. Η φυσική γλώσσα ενδείκνυται μόνο σε καθημερινά προβλήματα στα οποία η ακρίβεια και η σαφήνεια δεν παίζουν σημαντικό ρόλο στην καθοριστικότητα και αποτελεσματικότητα του τρόπου επίλυσής τους.

Το **διάγραμμα ροής** είναι η αναπαράσταση του αλγορίθμου με τη χρήση γεωμετρικών σχημάτων. Είναι ο πλέον εποπτικός τρόπος παρουσίασης. Σε ένα διάγραμμα ροής μπορούμε, με μια ματιά, να αναγνωρίσουμε τις λογικές δομές που περιλαμβάνει ο αλγόριθμος. Από την άλλη, η χρήση σχημάτων περιορίζει τη δυνατότητα έκφρασης σε ένα σχετικά μικρό χώρο, όπως είναι μια εκτυπώσιμη σελίδα. Αυτό καθιστά σχεδόν αδύνατη την αναπαράσταση ενός αλγορίθμου εκατοντάδων ή χιλιάδων εντολών με ένα διάγραμμα ροής. Ο περιορισμός αυτός δεν υπάρχει σε μια γλώσσα. Τα σχήματα που χρησιμοποιούνται στα διαγράμματα του παρόντος βιβλίου και η έννοιά τους, φαίνονται στον παρακάτω πίνακα:

Σχήμα		Έννοια
	Βέλος	Ροή (οδηγεί στην επόμενη εντολή)
	Έλλειψη	Αρχή - Τέλος αλγορίθμου
	Πλάγιο - παραλληλόγραμμο	Είσοδος - Έξοδος
	Ορθογώνιο παραλληλόγραμμο	Επεξεργασία - Εκτελεση πράξεων
	Ρόμβος	Συνθήκη

Εικόνα 3-7. Σύμβολα διαγραμμάτων ροής

Οι **γλώσσες περιγραφής αλγορίθμων** αποτελούν ενδιάμεσο στάδιο αναπαράστασης. Προσεγγίζουν τις γλώσσες προγραμματισμού, είναι όμως απλούστερες. Στόχος τους είναι η κωδικοποιημένη αποτύπωση αλγορίθμων σε κείμενο, χωρίς, όμως, να ενδιαφέρουν οι λεπτομέρειες υλοποίησής τους από έναν υπολογιστή. Γι' αυτό και αναφέρονται ως ψευδογλώσσες ή ψευδοκώδικες.

Οι γλώσσες προγραμματισμού αποτελούν τις γλώσσες επικοινωνίας του ανθρώπου με τον υπολογιστή. Η προσπάθεια του ανθρώπου να αναθέσει την επίλυση προβλημάτων σε υπολογιστές οδήγησε στη δημιουργία γλωσσών που έχουν περιορισμένο σύνολο κωδικοποιημένων εντολών με αυστηρούς κανόνες σύνταξης. Με τη χρήση τους, ο άνθρωπος δίνει εντολές στον υπολογιστή και αυτός τις εκτελεί.

Ας δούμε τον αλγόριθμο υπολογισμού περιφέρειας και εμβαδού ενός κύκλου. Στην Εικόνα 3-8, ο ίδιος αλγόριθμος έχει εκφρασθεί με τρεις διαφορετικούς τρόπους. Ανάμεσα στους τρεις τρόπους υπάρχει αντιστοιχία των εντολών. Γενικά, είναι εύκολο να μετατρέψουμε έναν αλγόριθμο από μια μορφή αναπαράστασης σε μια άλλη. Αν γνωρίζουμε καλά μια γλώσσα προγραμματισμού, τότε είναι εύκολο να μάθουμε και μια δεύτερη, αρκεί να έχουμε κατανοήσει την αντιστοιχία των εντολών τους.

Φυσική Γλώσσα	Ψευδογλώσσα	Διάγραμμα ροής
Αρχή αλγορίθμου	**Αλγόριθμος** Κύκλος	Αρχή
Ρώτα να μάθεις πόσο είναι η ακτίνα.	**Γράψε** 'Δώσε ακτίνα'	Διάβασε ακτίνα
Πάρε την ακτίνα.	**Διάβασε** ακ	περιφέρεια ← ακτίνα*2*3.14
Πολλαπλασίασε την ακτίνα με το 2 και ότι βρεις με το 3.14. Το αποτέλεσμα είναι η περιφέρεια του κύκλου.	περ ← ακ*2*3.14	εμβαδό ← ακτίνα^2*3.14
Ύψωσε την ακτίνα στο 2 και πολλαπλασίασε ότι βρεις με το 3.14. Το αποτέλεσμα είναι το εμβαδό του κύκλου.	εμβ ← ακ^2*3.14	Γράψε περιφέρεια, εμβαδό
Πες την περιφέρεια και το εμβαδό που υπολόγισες.	**Γράψε** περ, εμβ	Τέλος
Τέλος αλγορίθμου	**Τέλος** Κύκλος	

Εικόνα 3-8. Τρεις αναπαραστάσεις του ίδιου αλγορίθμου

Ποιος από τους τρόπους αναπαράστασης είναι ο καλύτερος; Πόσο εύκολη είναι η δημιουργία διαγραμμάτων στον υπολογιστή; Ένα πρόγραμμα σε γλώσσα προγραμματισμού μπορεί να κατανοηθεί και εκτελεστεί από οποιονδήποτε άνθρωπο; Ένα κείμενο σε φυσική γλώσσα μπορεί να εκτελεστεί άμεσα από τον υπολογιστή; Οι απαντήσεις δεν είναι άμεσες ούτε εύκολες. Η φύση και το μέγεθος του προβλήματος, καθορίζουν τον καταλληλότερο τρόπο αναπαράστασης.

Συνοψίζοντας τους τρόπους αναπαράστασης αλγορίθμων, παρατηρούμε ότι ο βαθμός δόμησης του τρόπου αναπαράστασης αυξάνεται καθώς πηγαίνουμε από τον πλούτο και την ελευθερία της φυσικής γλώσσας προς τους αυστηρούς λεξικολογικούς και συντακτικούς κανόνες των γλωσσών προγραμματισμού. Η περιγραφή αλγορίθμου σε φυσική γλώσσα είναι ο λιγότερο δομημένος, ενώ οι γλώσσες προγραμματισμού ο πιο δομημένος τρόπος αναπαράστασης αλγορίθμων.

Ερωτήσεις - Θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Ένας αλγόριθμος μπορεί να εκφρασθεί με πολλούς διαφορετικούς τρόπους.
2. Η φυσική γλώσσα αποτελεί τον πιο απλό και ανεπεξέργαστο τρόπο παρουσίασης ενός αλγορίθμου.
3. Ένα κείμενο σε φυσική γλώσσα μπορεί να εκτελεστεί άμεσα από τον υπολογιστή.
4. Η περιγραφή του αλγορίθμου σε φυσική γλώσσα περιέχει απλά λόγια και ελεύθερες εκφράσεις.
5. Η περιγραφή του αλγορίθμου σε διάγραμμα ροής περιέχει σχήματα.
6. Η αναπαράσταση του αλγορίθμου με σχήματα είναι ο λιγότερο εποπτικός τρόπος παρουσίασης.
7. Οι γλώσσες προγραμματισμού είναι οι γλώσσες επικοινωνίας του ανθρώπου με τον υπολογιστή.
8. Ο καλύτερος τρόπους αναπαράστασης είναι η γλώσσα προγραμματισμού.
9. Ένα πρόγραμμα σε γλώσσα προγραμματισμού C μπορεί να κατανοηθεί και εκτελεστεί από οποιονδήποτε άνθρωπο.

Β. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με την κατάλληλη έννοια της στήλης Β

1.

Στήλη Α	Στήλη Β
Φυσική γλώσσα	Πιο δομημένος
Διάγραμμα ροής	Πιο αδόμητος
Γλώσσες περιγραφής αλγορίθμων	Πιο εποπτικός

2.

Στήλη Α	Στήλη Β
Βέλος	Επεξεργασία
Έλλειψη	Είσοδος - Έξοδος
Πλάγιο παραλληλόγραμμο	Συνθήκη
Ορθογώνιο παραλληλόγραμμο	Ροή (επόμενη εντολή)
Ρόμβος	Αρχή - Τέλος αλγορίθμου

Γ. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Τα Κινέζικα είναι μια _____ γλώσσα.
2. Η _____ _____ αποτελεί τον πιο απλό και ανεπεξέργαστο τρόπο παρουσίασης ενός αλγορίθμου.
3. Το _____ _____λεί μια εύληπτη και παραστατική παρουσίαση αλγορίθμου σε δισδιάστατη μορφή.
4. Οι _____ _____ _____ αποτελούν κωδικοποιημένη αποτύπωση αλγορίθμου με κείμενο.
5. Οι _____ _____ αποτελούν αποτύπωση αλγορίθμου με περιορισμένο σύνολο κωδικοποιημένων εντολών, κατανοητών από τον υπολογιστή.
6. Οι γλώσσες περιγραφής αλγορίθμων αναφέρονται και ως _____

Ασκήσεις - Προβλήματα

1. Γράψτε αλγόριθμο που δέχεται τον προφορικό και το γραπτό βαθμό ενός μαθητή υπολογίζει το μέσο όρο του και εξάγει αποτέλεσμα προαγωγής. Ο μαθητής προάγεται αν έχει μέσο όρο πάνω από 9.5, διαφορετικά παραπέμπεται για επανεξέταση. Εκφράστε τον αλγόριθμο και με τους τρεις τρόπους αναπαράστασης, φυσική γλώσσα, διάγραμμα ροής και Ψευδογλώσσα.

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να αναγνωρίζετε τις διάφορες μορφές αναπαράστασης του αλγορίθμου.
- να συγκρίνετε και να επιλέγετε τον κατάλληλο τρόπο αναπαράστασης ανάλογα με το πρόβλημα και τον αλγόριθμο.
- να εντοπίζετε και να περιγράφετε τη λειτουργία σχολίων σε απλά παραδείγματα αλγορίθμων.

κεφάλαιο

4

Θέματα Εφαρμοσμένης Επιστήμης των Υπολογιστών

Γλώσσες αναπαράστασης αλγορίθμων

Στόχοι

Στόχοι του κεφαλαίου είναι οι μαθητές:

- να διακρίνουν τις βασικές εντολές και δομές που χρησιμοποιούνται σε έναν αλγόριθμο.
- να επιλύουν απλά προβλήματα διατυπώνοντας τη λύση σε μορφή αλγορίθμου.
- να γνωρίσουν διαφορετικούς αλγορίθμους για την επίλυση συγκεκριμένων προβλημάτων και να προβληματιστούν για τα χαρακτηριστικά τους.
- να εντοπίζουν και να διορθώνουν τα λογικά λάθη ενός αλγορίθμου.
- να γνωρίσουν τη σημασία της τεκμηρίωσης σε έναν αλγόριθμο.

Λέξεις κλειδιά

Γλώσσες περιγραφής αλγορίθμων	Σταθερά, Μεταβλητή, Τελεστής, Έκφραση, Εντολές αλγορίθμου, Δομές αλγορίθμων, Εκσφαλμάτωση, Λογικά λάθη, Συντακτικά λάθη, Τεκμηρίωση

Εισαγωγικές ερωτήσεις

Εισαγωγικές ερωτήσεις

1. Ποιες είναι οι “εντολές” που χρησιμοποιείτε καθημερινά στον αλγόριθμο της προετοιμασίας σας για το σχολείο;
2. Πόσες συνθήκες χρειάζονται για την επιτυχή σας φοίτηση;
3. Έχετε σκεφτεί τι σημαίνει “τερματίζω” μια επανάληψη;

4. Γλώσσες αναπαράστασης αλγορίθμων

4.1 Γενικά στοιχεία αναπαράστασης αλγορίθμων

Στο προηγούμενο κεφάλαιο γνωρίσαμε την έννοια του αλγορίθμου σε θεωρητικό επίπεδο, ως τη γενική διαδικασία επίλυσης ενός προβλήματος. Στη συνέχεια θα μπούμε στην πράξη, θα δούμε πως αντιμετωπίζουμε προβλήματα με τη δημιουργία αλγορίθμων. Για την αναπαράσταση των αλγορίθμων του κεφαλαίου θα χρησιμοποιήσουμε μια Ψευδογλώσσα με κωδικοποιημένες εντολές χωρίς να μας απασχολεί η αυστηρότητα που διέπει μια επίσημη γλώσσα προγραμματισμού.

Γενικά, η μορφή ενός αλγορίθμου σε Ψευδογλώσσα πρέπει να ακολουθεί την παρακάτω σύνταξη:

```
Αλγόριθμος Όνομα(ΔεδομέναΕισόδου↓, ΔεδομέναΕξόδου↑)

Εντολές

Τέλος Όνομα
```

Ο αλγόριθμος αποτελείται από την επικεφαλίδα και το κύριο σώμα. Στην επικεφαλίδα πρώτα γράφεται η δεσμευμένη λέξη Αλγόριθμος. Ακολουθεί ένα όνομα που προσδιορίζει τη λειτουργία του αλγορίθμου και μέσα σε παρενθέσεις προαιρετικά οι μεταβλητές εισόδου (↓) και εξόδου (↑). Το κύριο σώμα περιέχει τις εντολές του αλγορίθμου. Ο αλγόριθμος κλείνει με τη δεσμευμένη λέξη Τέλος ακολουθούμενη από το όνομα του αλγορίθμου.

Πριν δούμε αναλυτικά τις εντολές της Ψευδογλώσσας, θα αποσαφηνίσουμε κάποιες βασικές έννοιες ενός αλγορίθμου.

Σταθερά είναι μια ποσότητα που η τιμή της δεν αλλάζει κατά τη διάρκεια εκτέλεσης του αλγορίθμου.

Μεταβλητή είναι μια ποσότητα που αναπαριστά ένα στοιχείο που έχει νόημα στον πραγματικό κόσμο. Η τιμή της μεταβλητής μπορεί να αλλάζει κατά τη διάρκεια εκτέλεσης του αλγορίθμου. Μια μεταβλητή αντιστοιχεί σε μια θέση μνήμης στον υπολογιστή. Το περιεχόμενό της θέσης μνήμης είναι η τιμή της μεταβλητής.

Οι σταθερές και οι μεταβλητές αποτελούν τα δεδομένα που υφίστανται επεξεργασία από τις εντολές του αλγορίθμου. Τόσο οι σταθερές, όσο και οι μεταβλητές είναι κάποιου συγκεκριμένου είδους, που λέγεται τύπος του δεδομένου. Στην Ψευδογλώσσα υποστηρίζονται 4 βασικοί τύποι δεδομένων: Ακέραιος, Πραγματικός, Λογικός, Χαρακτήρες.

Ας θεωρήσουμε το παράδειγμα ενός κινητού για να αντιληφθούμε τον τύπο δεδομένων. Η τιμή του κινητού είμαι μια μεταβλητή πραγματικού τύπου αφού οι τιμές που μπορεί να πάρει είναι πραγματικοί αριθμοί (π.χ. 95.6). Αντίστοιχα, ο κωδικός μοντέλου είναι τύπου χαρακτήρες (π.χ. GE1742) και αν υποστηρίζει 4G είναι λογικού τύπου (ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ). Προφανώς, ο τύπος μιας

Κάθε μεταβλητή έχει όνομα που εκφράζει το περιεχόμενό της. Το όνομα πρέπει:
- να μην έχει κενά
- να μην έχει σύμβολα (+,-,*,$,%)
- να μην ξεκινά από αριθμό, αν και μπορεί να περιέχει.
- να μην είναι **δεσμευμένη λέξη** (λέξη που χρησιμοποιείται στη γλώσσα για ειδικό σκοπό)

μεταβλητής σχετίζεται με τις τιμές που αυτή μπορεί να πάρει κατά τη διάρκεια του αλγόριθμου. Η σχέση των δεδομένων και του τύπου τους φαίνονται στον παρακάτω πίνακα:

Ποσότητα	**Τύπος**	**Πιθανές τιμές (Σταθερές)**
Μέγεθος μνήμης	Ακέραιος	128 1024
Τιμή κινητού	Πραγματικός	95.6 137.20
Κωδικός μοντέλου	Χαρακτήρες	‘GE1742’ ‘S34-1G’
Υποστηρίζει 4G ;	Λογικός	ΑΛΗΘΗΣ ΨΕΥΔΗΣ

Οι τιμές που μπορεί να πάρει μια μεταβλητή, ουσιαστικά είναι σταθερές του ίδιου τύπου. Οι σταθερές τύπου Χαρακτήρες, εισάγονται σε εισαγωγικά για να ξεχωρίσουν στον αλγόριθμο από τα ονόματα των μεταβλητών.

Τελεστής είναι το σύμβολο που χρησιμοποιείται για την εκτέλεση μιας πράξης. Υποστηρίζονται τρία είδη πράξεων, οι αριθμητικές, οι συγκριτικές και οι λογικές. Ο τελεστής ^ είναι η ύψωση σε δύναμη. Οι τελεστές div και mod υλοποιούν το πηλίκο και το υπόλοιπο της διαίρεσης ακεραίων, αντίστοιχα. Ο συγκριτικός τελεστής <> είναι το διάφορο.

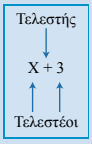
Αριθμητικοί	Συγκριτικοί	Λογικοί
^		ΟΧΙ (άρνηση)
* / div mod	>, < , >= , <= , = , <>	ΚΑΙ (σύζευξη)
+ -		Ή (διάζευξη)

Εικόνα 4-1. Οι τελεστές της Ψευδογλώσσας και η προτεραιότητά τους

Έκφραση είναι ο συνδυασμός σταθερών και μεταβλητών (τελεστέοι) με πράξεις (τελεστές). Κάθε έκφραση μπορεί να αποτιμηθεί. Η τελική τιμή μιας έκφρασης εξαρτάται από την ιεραρχία των πράξεων και τη χρήση των παρενθέσεων. Η τιμή μιας έκφρασης μπορεί να είναι είτε αριθμός, οπότε μιλάμε για **αριθμητική έκφραση**, είτε ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ οπότε μιλάμε για **λογική έκφραση**. Μια λογική έκφραση που περιέχει λογικούς τελεστές λέγεται **σύνθετη**, διαφορετικά **απλή**. Σε μια έκφραση εκτελούνται πρώτα οι αριθμητικοί, μετά οι συγκριτικοί και τέλος οι λογικοί τελεστές.

Η δημιουργία κάθε αλγορίθμου στηρίζεται στη χρήση τριών αλγοριθμικών δομών της δομής ακολουθίας, της δομής επιλογής και της δομής επανάληψης. Με τον συνδυασμό των τριών παραπάνω δομών μπορεί να υλοποιηθεί οποιοσδήποτε αλγόριθμος όσο δύσκολος κι αν είναι αυτός!

Μια έκφραση σχηματίζεται όταν εφαρμόζουμε πράξεις πάνω σε δεδομένα, π.χ.



Τελεστής: αυτός που τελεί, κάνει, πράττει = πράξη
Τελεστέος: αυτός που τελείται, παθαίνει την πράξη

Αν ο αριθμητικός τελεστής + αντικατασταθεί από το συγκριτικό > τότε η έκφραση από αριθμητική γίνεται λογική.

Ερωτήσεις - θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Σταθερά είναι μια ποσότητα που η τιμή της αλλάζει κατά τη διάρκεια εκτέλεσης του αλγορίθμου
2. Μια σταθερά μπορεί να είναι ακέραια, πραγματική, λογική ή χαρακτήρες.
3. Η τιμή μιας μεταβλητής δε μπορεί να αλλάξει κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου.
4. Ο τύπος μιας μεταβλητής μπορεί να αλλάξει κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου.
5. Μια έκφραση περιέχει μόνο σταθερές και μεταβλητές.
6. Η τιμή μιας λογικής έκφρασης μπορεί να είναι Αληθής ή Ψευδής

Β. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με το κατάλληλο στοιχείο της στήλης Β

Στήλη Α **όνομα μεταβλητής**	**Στήλη Β** **χαρακτηρισμός**
1. Φ.Π.Α.	α. αποδεκτή β. μη αποδεκτή
2. 2ΑΒ	
3. ΑΒ2	
4. ΔΙΑΒΑΣΕ	
5. ΒΑΘΜΟΣ	
6. "ΜΙΣΘΟΣ"	
7. ΕΜΦΑΝΙΣΕ_Χ	
8. ΑΚΕΡΑΙΟΣ	
9. Β*2	
10.S$	

Γ. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με το κατάλληλο στοιχείο της στήλης Β

Στήλη Α	**Στήλη Β**
1. ΦΠΑ	α. σταθερά β. μεταβλητή γ. τελεστής δ. έκφραση
2. 2	
3. ^	
4. Α+3	
5. Α>3	
6. ΜΙΣΘΟΣ	
7. 'ΜΙΣΘΟΣ'	
8. < >	
9. DIV	
10. Α DIV 3	

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να περιγράφετε τα βασικά συνθετικά μέρη ενός αλγόριθμου (σταθερές, μεταβλητές, τελεστές, εκφράσεις)

4.2 Δομή Ακολουθίας

Η δομή ακολουθίας στην καθημερινή ζωή της Ελένης

Έχω πάρτι το βράδυ! Έχω όμως και διάβασμα. Πρέπει να τελειώσω στην ώρα μου, για να έχω χρόνο μετά να ετοιμαστώ και να είμαι στις 9 στο πάρτι. Ας οργανωθώ: 7:00-7:30 να διαβάσω. 7:30-7:45 να κάνω μπάνιο. 7:45-8:00 να στεγνώσω τα μαλλιά μου. 8:00-8:15 να μακιγιαριστώ. 8:15-8:30 να ντυθώ. 8:45-9:00 να πάω στο πάρτι. Όλα πρέπει να εκτελεσθούν στη σειρά και με απόλυτη ακρίβεια. Αν αλλάξω τη σειρά των βημάτων ή παραλείψω κάποιο βήμα, δε θα έχω το επιθυμητό αποτέλεσμα.

Η δομή ακολουθίας χρησιμοποιείται για την αντιμετώπιση απλών προβλημάτων όπου η εκτέλεση των εντολών είναι αποκλειστικά σειριακή. Πρακτικά, η δομή ακολουθίας περιέχεται σε κάθε αλγόριθμο.

Η δομή ακολουθίας προκύπτει από συνδυασμό τριών βασικών εντολών: της εντολής εισόδου με την οποία εισάγονται δεδομένα στον αλγόριθμο, της εντολής εξόδου με την οποία εξάγονται τα αποτελέσματά του και της εντολής εκχώρησης τιμής που εκτελεί πράξεις ανάμεσα στα δεδομένα.

Τρεις είναι οι βασικές εντολές της Ψευδογλώσσας και αντιστοιχούν στις τρεις βασικές λειτουργίες που επιτελούνται σε οποιοδήποτε κωδικοποιημένο αλγόριθμο: η εντολή εισόδου με την οποία εισάγονται δεδομένα στον αλγόριθμο, η εντολή εξόδου με την οποία εξάγονται τα αποτελέσματά του και η εντολή επεξεργασίας που εκτελεί πράξεις ανάμεσα στα δεδομένα.

α. Εντολή Εισόδου

Σύνταξη	**Διάβασε** `λίστα_μεταβλητών`
Λειτουργία	Με την εντολή αυτή ο εισάγονται τιμές από το χρήστη. Κατά την εκτέλεση της, σταματά προσωρινά η ροή εκτέλεσης του αλγόριθμου. Κάθε τιμή που εισάγεται εκχωρείται στην αντίστοιχη μεταβλητή. Όταν δοθούν τόσες τιμές, όσες και οι μεταβλητές στη λίστα, ολοκληρώνεται η είσοδος και συνεχίζει η εκτέλεση του αλγόριθμου.
Π.χ.	Διάβασε `ΦΠΑ,Τ`

Διάβασε ΦΠΑ, Τ

β. Εντολή εκχώρησης τιμής

Σύνταξη	`μεταβλητή ← έκφραση`
Λειτουργία	Με την εντολή αυτή εκτελούνται οι πράξεις που υπάρχουν στην έκφραση, υπολογίζεται η τιμή της και αποδίδεται στη μεταβλητή.
Πχ.	`Φ ← Τ*ΦΠΑ`

ΦΠΑ ← Τ*ΦΠΑ

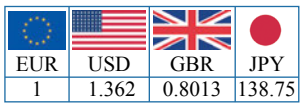
γ. Εντολή Εξόδου

Σύνταξη	**Γράψε** λίστα_αποτελεσμάτων
Λειτουργία	Με την εντολή αυτή παρουσιάζονται τα αποτελέσματα του αλγόριθμου. Η μορφή της εξόδου εξαρτάται από τον εκτελεστή και δεν είναι σημαντική. Θα μπορούσε να είναι η αναφώνηση από τον εκτελεστή, η εμφάνιση σε μια οθόνη, η εκτύπωση σε έναν εκτυπωτή. Κάθε αποτέλεσμα μπορεί ναι είναι είτε ένα μήνυμα μέσα σε εισαγωγικά οπότε παρουσιάζεται αυτούσιο, ή μια μεταβλητή οπότε παρουσιάζεται η τιμή της ή μια παράσταση οπότε υπολογίζεται και παρουσιάζεται η τιμή της.
Π.χ.	**Γράψε** Φ

Παράδειγμα 4-1. Θέλουμε να κατεβάσουμε στο κινητό μας τηλέφωνο μια εφαρμογή. Η ταχύτητα σύνδεσής μας στο διαδίκτυο είναι 1,6 MB/δευτερόλεπτο. Γ ράψτε αλγόριθμο που δέχεται ως είσοδο το μέγεθος της εφαρμογής σε MB και υπολογίζει σε πόσο χρόνο θα κατέβει η εφαρμογή.

```
Αλγόριθμος Download(μέγεθος↓, χρόνος↑)
     Γράψε 'Δώστε το μέγεθος της εφαρμογής'
     Διάβασε μέγεθος
     χρονος ← μέγεθος/1,6
     Γράψε χρόνος
Τέλος Download
```

Αλγόριθμος 4-1. Μεταφόρτωση εφαρμογής



EUR	USD	GBR	JPY
1	1.362	0.8013	138.75

Εικόνα 4-2. Ισοτιμίες συναλλάγματος

Παράδειγμα 4-2. Στο ανταλλακτήριο συναλλάγματος, ισχύουν σήμερα οι ισοτιμίες της Εικόνας 4-2. Να μετατρέψετε ένα χρηματικό ποσό δοσμένο σε ευρώ σε αντίστοιχα ποσά σε δολάρια, λίρες και γεν.

Στο ανταλλακτήριο συναλλάγματος, ισχύουν σήμερα οι ισοτιμίες της Εικόνας 4-2. Να μετατρέψετε ένα χρηματικό ποσό δοσμένο σε ευρώ σε αντίστοιχα ποσά σε δολάρια, λίρες και γεν.

```
Αλγόριθμος Συνάλλαγμα(EUR↓,USD↑,GBP↑,JPY↑)
USD ← EUR * 1.362
GBP ← EUR * 0.8013
JPY ← EUR * 138.75
Τέλος Συνάλλαγμα
```

Αλγόριθμος 4-2. Ανταλλακτήριο συναλλάγματος

Ερωτήσεις - θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Κατά την εκτέλεση του αλγορίθμου η εντολή ΔΙΑΒΑΣΕ διακόπτει προσωρινά την ροή εκτέλεσής του και περιμένει την εισαγωγή τιμών από το πληκτρολόγιο.
2. Η γενική μορφή της εντολής εκχώρησης τιμής είναι Έκφραση ←Μεταβλητή
3. Η εντολή Γράψε είναι η εντολή εισόδου της Ψευδογλώσσας.
4. Οι βασικές εντολές της Ψευδογλώσσας είναι οι εντολές εισόδου και εξόδου.
5. Η δομή ακολουθίας περιέχεται σε κάθε αλγόριθμο.
6. Η δομή ακολουθίας χρησιμοποιείται για την αντιμετώπιση προβλημάτων όπου απαιτείται αποκλειστικά σειριακή εκτέλεση εντολών.

Β. Αντιστοιχίστε κάθε εντολή της στήλης Α με τον κατάλληλο τύπο μεταβλητής της στήλης Β.

ΣτήληΑ **Εκχώρηση Τιμής**	**Στήλη Β** **Τύπος μεταβλητής**
1. Α ← 14 div 5	Α. ακέραιος Β. πραγματικός Γ. χαρακτήρας Δ. λογικός
2. Β ← “6”	
3. Γ ← 8.6	
4. Δ ← (Α + 3) / 3	
5. Ε ← Β	
6. Ζ ← Αληθής	
7. Η ← 'Αληθής'	
8. Θ ← Α > 4	
9. όνομα ← 'Γιώργος'	
10 Ηλικία ←18	

Γ. Να μετατρέψετε τις παρακάτω προτάσεις σε εντολές εκχώρησης τιμής,

1. Αύξησε το πλήθος (Π) κατά μία μονάδα
2. Αύξησε το άθροισμα (Αθρ) κατά χ
3. Υπολόγισε το μέσο όρο (ΜΟ) διαιρώντας το άθροισμα (Αθρ) δια το πλήθος (Π)
4. Διπλασίασε το Π

Δ. Δίνεται ο παρακάτω αλγόριθμος σε μορφή Ψευδογλώσσας. Να μετατραπεί σε διάγραμμα ροής.

```
Αλγόριθμος Ασκ_1
  Διάβασε Τ,Φ
  Φορ ← Τ*Φ
  ΤΤ ← Τ+Φορ
    Γράψε ΤΤ
  Τέλος Ασκ_1
```

Ε. Να συμπληρώσετε τον πίνακα παρακολούθησης τιμών του παρακάτω αλγορίθμου

```
Αλγόριθμος Ασκ
1    α ← 5
2    β ← α + 5
3    Γράψε α
4    Γράψε 2*β
Τέλος Ασκ
```

Αριθμός Γραμμής	Μεταβλητές		Έξοδος
	Α	Β	
1	5		

Ασκήσεις - Προβλήματα

1. Γράψτε αλγόριθμο ο οποίος δέχεται τα ύψη τριών μαθητών σε εκατοστά και τυπώνει το μέσο ύψος τους σε μέτρα.
2. Γράψτε αλγόριθμο ο οποίος δέχεται βαθμούς Φαρενάϊτ και υπολογίζει βαθμούς Κελσίου με βάση τον εξής τύπο: Κελσίου = 5/9 (Φαρενάϊτ -32)
3. Γράψτε αλγόριθμο ο οποίος δέχεται την τιμή ενός προϊόντος και υπολογίζει την τιμή του μαζί με τον φόρο, δεδομένου ότι ΦΠΑ = 23%.
4. Φτιάξτε έναν αλγόριθμο που δέχεται τιμή ενός προϊόντος και το ποσοστό % της έκπτωσης που έχει και υπολογίζει την μειωμένη νέα τιμή του προϊόντος.
5. Θεωρείστε ότι ένα ορθογώνιο σπίτι βρίσκεται κτισμένο μέσα σε ορθογώνιο οικόπεδο. Το υπόλοιπο του οικοπέδου είναι καλυμμένο με γκαζόν. Να γραφτεί αλγόριθμος που διαβάζει τις διαστάσεις του οικοπέδου και του σπιτιού και υπολογίζει και τυπώνει τον χρόνο σε λεπτά που χρειάζεται κάποιος να κουρέψει το γκαζόν αν σε κάθε δευτερόλεπτο κουρεύει 0,2 τετραγωνικά μέτρα γκαζόν.

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση

- να διακρίνετε και να περιγράφετε τις βασικές εντολές της δομής ακολουθίας που χρησιμοποιούνται σε έναν αλγόριθμο
- να αναγνωρίζετε από την εκφώνηση του προβλήματος, ποιες αλγοριθμικές εντολές και με ποιον συνδυασμό, απαιτούνται για την επίλυση ενός δοσμένου προβλήματος.

4.3 Δομή Επιλογής

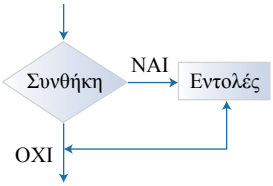
Η δομή επιλογής στην καθημερινή ζωή της Μαρίας

Ποιο δρόμο να ακολουθήσω για να γυρίσω από το σχολείο στο σπίτι; Φυσικά τον πιο σύντομο. Μήπως, όμως, ο πιο σύντομος δρόμος εξαρτάται από κάποιες συνθήκες; Πρώτα από όλα, με ποιο μεταφορικό μέσο μετακινούμαι, με τα πόδια, το ποδήλατο ή το αυτοκίνητο της μαμάς; Άλλος είναι ο συντομότερος δρόμος αν είμαι πεζή και άλλος με το αυτοκίνητο. Και αν είμαι με το αυτοκίνητο τι ώρα είναι και τι κίνηση έχει; Μήπως κάποιος δρόμος είναι κλειστός λόγω έργων; Αν δεν είναι, έχει καλώς. Αλλά αν είναι; Καλύτερα να πάω με τα πόδια και ας βρέχει !!!!

Στην επίλυση προβλημάτων, συχνά συναντάμε την ανάγκη να πάρουμε αποφάσεις και να επιλέξουμε διαφορετικούς τρόπους αντιμετώπισης, με βάση κάποια κριτήρια. Εξετάζουμε εναλλακτικές περιπτώσεις και σε κάθε μία εκτελούμε διαφορετικές εντολές. Οι περιπτώσεις διαμορφώνονται από τον έλεγχο των συνθηκών που επικρατούν. Στην καθημερινή ζωή, η απόφαση που παίρνουμε έχει τη λέξη Αν. «Αν ισχύει αυτό, θα κάνω εκείνο...». Παρόμοια, στην Ψευδογλώσσα, η δομή επιλογής εκφράζεται με την εντολή Αν σε τρεις μορφές: την απλή, τη σύνθετη και την πολλαπλή επιλογή.

4.3.1. Απλή επιλογή

Σύνταξη	**Αν** συνθήκη **τότε** **Αν** συνθήκη **τότε** εντολή Εντολές **Τέλος_αν**
Λειτουργία	Αν ισχύει η συνθήκη (δηλαδή αν είναι αληθής), τότε μόνο εκτελούνται οι εντολές. Σε κάθε περίπτωση, στη συνέχεια εκτελείται η εντολή που ακολουθεί τη δεσμευμένη λέξη τέλος_αν. Όταν θέλουμε να εκτελέσουμε μόνο μια βασική εντολή, η επιλογή μπορεί να γραφεί σε μια σειρά χωρίς τη λέξη τέλος_αν.



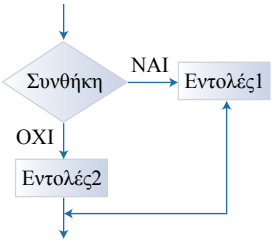
Εικόνα 4-3. Διάγραμμα ροής της απλής επιλογής

Η δομή απλής επιλογής χρησιμοποιείται όταν θέλουμε να εξετάσουμε μια συνθήκη και να εκτελέσουμε ή να παρακάμψουμε κάποιες εντολές.

4.3.2. Σύνθετη επιλογή

Η σύνθετη επιλογή είναι η πιο συχνά εμφανιζόμενη εντολή επιλογής. Με τον έλεγχο μιας συνθήκης, μπορούμε να διακρίνουμε δύο περιπτώσεις και σε κάθε μία να εκτελέσουμε διαφορετικές εντολές.

Σύνταξη	**Αν** συνθήκη **τότε** Εντολές1 **αλλιώς** Εντολές2 **Τέλος_αν**
Λειτουργία	Αν ισχύει η συνθήκη (δηλαδή αν είναι αληθής), τότε εκτελούνται μόνο οι Εντολές1. Αν η συνθήκη δεν ισχύει (αν είναι ψευδής) τότε εκτελούνται μόνο οι Εντολές2,. Σε κάθε περίπτωση, η εκτελεση του αλγόριθμου συνεχίζεται με την εντολή που ακολουθεί τη λέξη τέλος_αν.



Εικόνα 4-4. Διάγραμμα ροής της σύνθετης επιλογής

Ας δούμε τη χρήση των βασικών αυτών δομών σε δύο απλά υπολογιστικά προβλήματα.

Παράδειγμα 4-3. Να υπολογιστεί η απόλυτη τιμή ενός αριθμού.
Από τον μαθηματικό ορισμό της απόλυτης τιμής γνωρίζουμε ότι η απόλυτη τιμή ισούται με τον ίδιο τον αριθμό αν είναι θετικός ή με τον αντίθετό του αν είναι αρνητικός. Άρα, υπάρχουν δύο περιπτώσεις, που μπορούν να εξετασθούν με χρήση σύνθετης ή απλής επιλογής.

Παράδειγμα 4-4. Να υπολογιστεί ο μεγαλύτερος από δύο αριθμούς.

```
Αλγόριθμος ΑπΤιμή1(x↓, ατ↑)
Αν x < 0 τότε
    ατ ← x * -1
Αλλιώς
    ατ ← x
Τέλος_αν
Τέλος ΑπΤιμή1
```

Αλγόριθμος 4-3 (α). Απόλυτη τιμή αριθμού με σύνθετη επιλογή

```
Αλγόριθμος ΑπΤιμή2(x↓, ατ↑)
ατ ← x
Αν x < 0 τότε
    ατ ← x * -1
Τέλος_αν
Τέλος ΑπΤιμή2
```

Αλγόριθμος 4-3 (β). Απόλυτη τιμή αριθμού με απλή επιλογή

Η λογική της λύσης (β) με την απλή επιλογή έχει μια σημαντική ιδιότητα. Μπορεί να επεκταθεί για 3, ή περισσότερους αριθμούς και να εφαρμοσθεί επαναληπτικά.

```
Αλγόριθμος MAX1(x↑,y↓,μεγ↑)
Αν x < y τότε
    μεγ ← y
Αλλιώς
    μεγ ← x
Τέλος_αν
Τέλος MAX1
```

```
Αλγόριθμος MAX2(x↑,y↓,μεγ↑)
μεγ ← x
Αν μεγ < y τότε
    μεγ ← y
Τέλος_αν
Τέλος MAX2
```

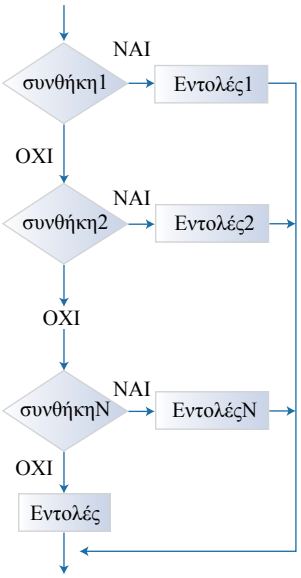
(α) *(β)*

Αλγόριθμος 4-4. Μέγιστος μεταξύ δύο αριθμών (α) με σύνθετη επιλογή, (β) με απλή επιλογή

4.3.3. Πολλαπλή επιλογή

Όταν σε κάποιο σημείο του αλγορίθμου, θέλουμε να εξετάσουμε παραπάνω από δύο περιπτώσεις και να εκτελέσουμε διαφορετική ενέργεια σε κάθε μία, τότε χρησιμοποιούμε τη δομή της πολλαπλής επιλογής. Η εξέταση των περιπτώσεων γίνεται με τον έλεγχο διαφορετικών συνθηκών και υλοποιείται στην Ψευδογλώσσα με την εντολή πολλαπλής επιλογής.

Σύνταξη	**Αν** συνθήκη 1 **τότε** εντολές1 **αλλιώς_αν** συνθήκη2 **τότε** εντολές2 ... **αλλιώς_αν** συνθήκηN **τότε** εντολέςN **αλλιώς** εντολές **τέλος_αν**
Λειτουργία	Πρώτα ελέγχεται η σύνθηκη1. Αν ισχύει, τότε εκτελούνται οι εντολές1. Αν δεν ισχύει ελέγχεται η συνθήκη2. Η εκτέλεση συνεχίζεται ομοίως. Όταν μια συνθήκη είναι ψευδής, προχωράμε στον έλεγχο της επόμενης. Όταν μια συνθήκη είναι αληθής, εκτελούμε τις αντίστοιχες εντολές και μεταβαίνουμε στο τέλος της δομής. Η εκτέλεση του αλγορίθμου συνεχίζεται με την εντολή που βρίσκεται μετά το τελος_αν..



Εικόνα 4-5. Διάγραμμα ροής της πολλαπλής επιλογής

Παράδειγμα 4-5. Ο δείκτης μάζας σώματος (ΔΜΣ) είναι μία ένδειξη για το βαθμό παχυσαρκίας ενός ατόμου και υπολογίζεται από τον τύπο: βάρος/ύψοςΛ2. Δεδομένων του βάρους και του ύψους ενός ατόμου, να εμφανίσετε χαρακτηρισμό που αφορά στο βαθμό παχυσαρκίας του, με βάση το ΔΜΣ.

Υπάρχουν τέσσερεις περιπτώσεις για τον χαρακτηρισμό του ατόμου, που μπορούν να εξετασθούν με μια δομή πολλαπλής επιλογής.

```
Αλγόριθμος ΧαρακτηρισμόςΑτόμου( βάρος↓, ύψος↓ )
ΔΜΣ ← βάρος / ύψος ^ 2
Αν ΔΜΣ < 18.5 τότε
     Γράψε 'ελλιποβαρές άτομο'
Αλλιώς αν ΔΜΣ < 25 τότε
     Γράψε 'άτομο με φυσιολογικό βάρος'
Αλλιώς αν ΔΜΣ < 30 τότε
     Γράψε 'υπέρβαρο άτομο'
αλλιώς
     Γράψε 'άτομο που πάσχει από παχυσαρκία'
Τέλος_αν
Τέλος ΧαρακτηρισμόςΑτόμου
```

Αλγόριθμος 4-5. Χαρακτηρισμός ατόμου με βάση το ΔΜΣ με πολλαπλή επιλογή

ΔΜΣ	Χαρακτηρισμός ατόμου
< 18,5	ελλιποβαρές
18,5 -24,9	φυσιολογικό βάρος
25 - 29,9	υπέρβαρο
> 30	πάσχει από παχυσαρκία

Η θεώρηση των περιπτώσεων μπορεί να γίνει εναλλακτικά εξετάζοντας πρώτα το βάρος και μετά τον προορισμό εμφωλευμένα.

Οι πολλαπλές επιλογές μπορούν να γίνουν και με εμφωλευμένες δομές επιλογής

4.3.4. Εμφωλευμένες Δομές Επιλογής

Σε οποιαδήποτε από τις τρεις μορφές της δομής επιλογής, απλή, σύνθετη ή πολλαπλή, μπορούμε να βάλουμε στη θέση των εντολών μια άλλη δομή επιλογής. Έτσι δημιουργείται μια εμφωλευμένη δομή, δηλαδή μια δομή επιλογής μέσα σε μια άλλη. Τοποθετώντας μια εντολή Αν μέσα σε μια άλλη, παίρνουμε υποπεριπτώσεις. Η λογική αυτή μπορεί να επεκταθεί χωρίς περιορισμό, δηλαδή να έχουμε νέα εμφωλευμένη δομή μέσα σε μία ήδη εμφωλευμένη κ.ο.κ.

Παράδειγμα 4-6. Στο ταχυδρομείο, το κόστος αποστολής υπολογίζεται συναρτήσει του προορισμού και του βάρους της επιστολής, με βάση τον πίνακα. Δεδομένων του προορισμού και του βάρους μιας επιστολής, να εμφανίσετε το ποσό που στοιχίζει η αποστολή της.

		Βάρος επιστολής	
		Μέχρι και 100 γραμμάρια	Πάνω από 100 γραμμάρια
Προορισμός επιστολής	Εσωτερικό	1 €	2 €
	Εξωτερικό	2,50 €	4 €

Δεν υπάρχει ένας μόνο συνδυασμός δομών επιλογής για την επίλυση του συγκεκριμένου προβλήματος. Αν εξετάσουμε πρώτα τις δύο περιπτώσεις για τον προορισμό με μια σύνθετη επιλογή και εμφωλευμένα πάρουμε υποπεριπτώσεις με σύνθετες επιλογές που εξετάζουν το βάρος, τότε προκύπτει εμφωλευμένη επιλογή που εξετάζει συνολικά 2 * 2 = 4 περιπτώσεις.

```
Αλγόριθμος Επιστολή(προορισμός↓,βάρος↓,κόστος↑)
Αν προορισμός = 'Εσωτερικό' τότε
     Αν βάρος <= 100  τότε
          κόστος ← 1
     αλλιώς
          κόστος ← 2
     τέλος_αν
αλλιώς
     Αν βάρος <= 100  τότε
          κόστος ← 2.5
     αλλιώς
          κόστος ← 4
     τέλος_αν
τέλος_αν
Τέλος Επιστολή
```

Αλγόριθμος 4-6. Υπολογισμός κόστους αποστολής με εμφωλευμένη επιλογή

Μια άλλη προσέγγιση είναι να εξετάσουμε τον προορισμό και το βάρος επιστολής μαζί, με μια σύνθετη λογική συνθήκη. Πρέπει προηγούμενα να έχουμε σκεφθεί όλες τις δυνατές περιπτώσεις. Ο αλγόριθμος που προκύπτει περιέχει πολλαπλή επιλογή με τέσσερις διακριτές περιπτώσεις.

```
Αλγόριθμος Επιστολή2( προορισμός↓, βάρος↓, κόστος↑ )
Αν προορισμός = 'Εσωτερικό' και βάρος <= 100 τότε
     κόστος ← 1
αλλιώς_αν προορισμός = 'Εσωτερικό' και βάρος <= 100 τότε
     κόστος ← 2
αλλιώς_αν προορισμός = 'Εξωτερικό' και βάρος <= 100 τότε
     κόστος ← 2.5
αλλιώς
     κόστος ← 4
τέλος_αν
Τέλος Επιστολή2
```

Αλγόριθμος 4-7. Υπολογισμός κόστους αποστολής με πολλαπλή επιλογή

Γενικά, στη δομή επιλογής, δεν υπάρχει ένας τρόπος λύσης. Μπορούμε να εξετάσουμε τις πιθανές περιπτώσεις με διαφορετικούς συνδυασμούς των τριών μορφών της εντολής Αν. Αυτό που είναι σημαντικό, είναι να επαληθεύσουμε ότι έχουμε λάβει υπόψη όλες τις πιθανές περιπτώσεις.

Ερωτήσεις - θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Όταν αριθμητικοί και συγκριτικοί τελεστές συνδυάζονται σε μια λογική έκφραση, οι αριθμητικές πράξεις εκτελούνται πρώτες.
2. Οι βασικοί λογικοί τελεστές είναι οι ΚΑΙ, ‘Η.
3. Η λογική έκφραση Χ Ή (ΟΧΙ Χ) είναι πάντα αληθής για κάθε τιμή της λογικής μεταβλητής Χ.
4. Η λογική πράξη της διάζευξης μεταξύ δύο προτάσεων είναι ψευδής όταν έστω μία από τις δύο προτάσεις είναι ψευδής.
5. Η λογική πράξη της διάζευξης μεταξύ δύο προτάσεων είναι αληθής όταν τουλάχιστον μία από τις δύο προτάσεις είναι αληθής.
6. Όταν δύο λογικές συνθήκες είναι αληθείς τότε η σύζευξή τους είναι αληθής.
7. Η σύγκριση ατομικών χαρακτήρων στηρίζεται στην αλφαβητική σειρά.
8. Η σύγκριση λογικών έχει έννοια μόνο στην περίπτωση του ίσου (=) και του διάφορου (<>)
9. Η λέξη ‘κακός’ θεωρείται μικρότερη από τη λέξη ‘καλός’
10. Η λογική έκφραση ‘ΜΕΓΑΛΟΣ’ > ‘ΜΙΚΡΟΣ’ είναι αληθής.
11. Στη δομή επιλογής μπορεί μία ή περισσότερες εντολές να μην εκτελεστούν.
12. Πολλαπλές επιλογές μπορούν να γίνουν και με εμφωλευμένες δομές επιλογής.
13. Μια δομή πολλαπλής επιλογής δε μπορεί να αναπαρασταθεί και με πολλές ανεξάρτητες δομές απλής επιλογής.

Β. Για κάθε μία από τις παρακάτω λογικές συνθήκες της Ψευδογλώσσας, γράψτε τον αριθμό της συνθήκης και δίπλα το γράμμα Σ ή Λ εάν θεωρείτε ότι είναι συντακτικά σωστή ή λανθασμένη, αντίστοιχα.

1. A+B = 10
2. 'ακτίνα' < 'διάμετρος'
3. A <= B*3^2
4. x > 'y'
5. x , y >= 0
6. A < 2*B ΚΑΙ B < 22
7. A < 2*B < 22
8. ΑΛΗΘΗΣ < Y
9. X <> ΑΛΗΘΗΣ
10. X>0 Η X-Y<0

Γ. Αντιστοιχίστε κάθε εντολή της στήλης Α με την κατάλληλη δομή της στήλης Β.

Στήλη Α	**Στήλη Β**
1. Εντολή Αν... τότε... αλλιώς_αν	Α. Δομή ακολουθίας
2. Εντολή Όσο.. .επανάλαβε	Β. Δομή επιλογής
3. Διάβασε χ	Γ. Δομή επανάληψης

Δ. Να συμπληρώσετε τους παρακάτω πίνακες αληθείας.

πρόταση Α	**πρόταση Β**	**Α και Β**	**Α ή Β**	**όχι Α ή Β**	**όχι (Α ή Β)**	**όχι Α και Β**
Αληθής	Αληθής					
Αληθής	Ψευδής					
Ψευδής	Αληθής					
Ψευδής	Ψευδής					

πρόταση Α	**πρόταση Β**	**πρόταση Γ**	**όχι Α ή όχι Β**	**όχι Α και (Β ή Γ)**
Αληθής	Αληθής	Αληθής		
Αληθής	Ψευδής	Ψευδής		
Ψευδής	Ψευδής	Αληθής		

Ε. Βρείτε τις τιμές των παρακάτω λογικών εκφράσεων όταν x= 9 και y= 3;

1. x-y/2 >=8
2. όχι x-3<=y+3
3. y<x και y>x/2
4. x<> y και όχι x-y/2 >=8
5. x mod 2=1
6. x<y ή x/2>=y+1

Ζ. Έστω σε αλγόριθμο ότι θέλετε να γράψετε συνθήκες που είναι αληθείς για τα παρακάτω:

1. μαθητές που είναι βαρύτεροι από 80 κιλά
2. μαθητές που είναι βαρύτεροι από 90 κιλά και ελαφρύτεροι από 50 κιλά
3. και οι δύο αριθμοί θετικοί
4. ο αριθμός1 διαιρείται από το 7
5. ο αριθμός1 διαιρείται από το 7 αλλά δε διαιρείται από το 3

Γράψτε τις συνθήκες χρησιμοποιώντας τις μεταβλητές: βάρος_μαθητή, αριθμός1, αριθμός 2

Η. Δίνεται το παρακάτω τμήμα αλγορίθμου:

```
Διάβασε βαθμός
Αν βαθμός > = 18,5 τότε
      Εμφάνισε «ΑΡΙΣΤΟΣ»
Αλλιώς
      Αν βαθμός > = 14 τότε
            Εμφάνισε «ΚΑΛΟΣ»
      Αλλιώς
            Αν βαθμός > = 10 τότε
            Εμφάνισε «ΜΕΤΡΙΟΣ»
      Αλλιώς
            Εμφάνισε «ΑΠΟΡΡΙΠΤΕΤΑΙ»
            Τέλος_αν
      Τέλος_αν
Τέλος_αν
```

1. Να το ξαναγράψετε χρησιμοποιώντας αναπαράσταση διαγράμματος ροής.
2. Να το ξαναγράψετε χρησιμοποιώντας αποκλειστικά μια δομή πολλαπλής επιλογής Αν ... Τότε ... Αλλιώς_αν....Τέλος_αν.
3. Να το ξαναγράψετε χρησιμοποιώντας αποκλειστικά μη εμφωλευμένες απλές δομές επιλογής Αν ... Τότε ... Τέλος_αν.

Θ. Τους παρακάτω τρεις αλγόριθμους έφτιαξαν 3 μαθητές ως απάντηση σε πρόβλημα που ζητά την εύρεση του μεγαλύτερου από τρεις αριθμούς.

1. Συμπληρώσετε τα κενά και στους τρεις, ώστε να δίνουν σωστό αποτέλεσμα σε ΟΛΕΣ τις περιπτώσεις.
2. Κατόπιν, δημιουργήσετε το αντίστοιχο διάγραμμα ροής του κάθε αλγορίθμου.

Α	Β	Γ
Αλγόριθμος μέγ Διάβασε x,y,z Αν x>=y τότε Max ← ____ Αλλιώς Max ← ____ τέλος αν Αν z> _____ τότε Max←z τέλος_αν Γράψε Max τέλος μέγ	Αλγόριθμος μέγ Διάβασε x, y, z Αν x>=y και x>=z τότε Max ← ____ Αλλιώς αν y>=x και _____τότε Max ← y Αλλιώς Max ← ____ τέλος αν Γράψε Max τέλος μέγ	Αλγόριθμος μέγ Διάβασε x,y,z Αν x>=y τότε Αν z >= ____ τότε Max ← z Αλλιώς Max ← ___ τέλος αν Αλλιώς Αν ____ τότε Max ← ___ Αλλιώς Max ^ τέλος αν τέλος αν Γράψε Max τέλος μέγ

Ι. Δίνεται το παρακάτω τμήμα αλγορίθμου

```
Αν X > 0 τότε
      Αν X > 5 τότε
            Y ← 1
      Αλλιώς
            Y ← 18
      Τέλος_αν
Αλλιώς
      Αν X >= -5 τότε
            Y ← 3
      Τέλος_αν
Τέλος_αν
Γράψε Y
```

1. Εκτελέσετε και βρείτε τι θα εμφανίσει για α) X = 1 β) X = 0 γ) X = 10
2. Σχεδιάσετε το αντίστοιχο διάγραμμα ροής
3. Ξαναγράψτε το, χρησιμοποιώντας αποκλειστικά μια δομή πολλαπλής επιλογής αν ... τότε ... αλλιώς_αν.

Ασκήσεις - Προβλήματα

1. Δημιουργήστε αλγόριθμους για την επίλυση της πρωτοβάθμιας και δευτεροβάθμιας εξίσωσης. Εκφράστε τους με διάγραμμα ροής και Ψευδογλώσσα.
2. Δεδομένων των καρτεσιανών συντεταγμένων χ, ψ ενός σημείου που δεν βρίσκεται πάνω στους άξονες, να εμφανίσετε το τεταρτημόριο στο οποίο ανήκει. Γράψτε τον αλγόριθμο τόσο με εμφωλευμένες σύνθετες επιλογές όσο και με πολλαπλή επιλογή.
3. Ηλεκτρική εταιρία χρεώνει την κατανάλωση ρεύματος σύμφωνα με την παρακάτω κλίμακα:
   - Τις πρώτες 200 μονάδες (0-200) προς 0,25 €./ μονάδα
   - Τις επόμενες 1000 μονάδες (201-1200) προς 0,40 €./ μονάδα
   - Τις πέρα των 1200 μονάδων προς 0,50 €./ μονάδα

Να γίνει αλγόριθμος που δέχεται τον αριθμό των μονάδων που καταναλώθηκαν από έναν πελάτη και εμφανίζει το ποσό των χρημάτων που χρωστάει ο πελάτης στην ηλεκτρική εταιρία.

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση

- να διακρίνετε και να περιγράφετε τα βασικά χαρακτηριστικά της δομής επιλογής που χρησιμοποιείται σε έναν αλγόριθμο
- να αναγνωρίζετε και να χρησιμοποιείτε μορφές της δομής επιλογής για την επίλυση ενός δοσμένου προβλήματος

4.4 Δομή Επανάληψης

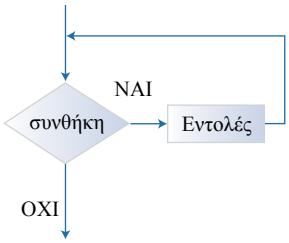
Η δομή επανάληψης στην καθημερινή ζωή της Κυριακής

Σήμερα πήρα τον έλεγχο του Α΄ τετραμήνου. Γράφει τους βαθμούς μου σε κάθε ένα από τα 10 μαθήματα, αλλά δε γράφει το μέσο όρο μου. Πώς θα τον υπολογίσω; Παίρνω το κομπιουτεράκι και αρχίζω. Πληκτρολογώ 15 και μετά +. Πληκτρολογώ 19 και μετά +. Πληκτρολογώ τον επόμενο βαθμό και μετά +. Στο τέλος πληκτρολογώ / και μετά 10. Ο μέσος όρος μου εμφανίζεται στην οθόνη. Επανέλαβα την ίδια δουλειά 10 φορές!

Πολλά προβλήματα απαιτούν την επανάληψη της ίδιας διαδικασίας. Η λογική των επαναληπτικών διαδικασιών εφαρμόζεται όταν μια σειρά από εντολές πρέπει να εφαρμοσθεί σε ένα σύνολο περιπτώσεων οι οποίες έχουν κάτι κοινό. Οι εντολές επανάληψης που υποστηρίζονται από την Ψευδογλώσσα είναι τρεις, η εντολή Όσο, η Μέχρις_ότου και η Για.

4.4.1. Εντολή Όσο

Η εντολή Όσο χρησιμοποιείται για την επανάληψη της ίδιας διαδικασίας όσο ικανοποιείται μια συνθήκη.



Εικόνα 4-6. Διάγραμμα ροής της επανάληψης Όσο

Σύνταξη	**Όσο** συνθήκη **επανάλαβε** εντολές **τέλος_επανάληψης**
Λειτουργία	Αρχικά ελέγχεται αν ισχύει η συνθήκη. Αν ισχύει τότε εκτελούνται οι εντολές και ελέγχεται ξανά η συνθήκη. Η διαδικασία επαναλαμβάνεται Όσο η συνθήκη είναι αληθής. Όταν κάποτε ελεγχθεί η συνθήκη και είναι ψευδής, η επανάληψη τελειώνει και εκτελείται η εντολή που ακολουθεί τη λέξη τέλος_επανάληψης.

Παράδειγμα 4-7. Ο καθηγητής έβαλε τους βαθμούς του Α΄ τετραμήνου στην Πληροφορική και θέλει να βρει το μέσο όρο του τμήματος και πόσοι μαθητές είναι άριστοι (βαθμός πάνω από 18). Οι βαθμοί κυμαίνονται από 0 έως 20. Δεδομένου ότι το πλήθος των μαθητών είναι άγνωστο, ο καθηγητής θέλει να τελειώσει την εισαγωγή των βαθμών όταν δώσει την ειδική τιμή -1. Η επανάληψη γίνεται για κάθε μαθητή. Το πλήθος των μαθητών δεν είναι γνωστό εκ των προτέρων. Μπορεί να είναι και μηδέν (0). Ενδείκνυται η χρήση της εντολής Όσο. Χρησιμοποιούνται τρεις μεταβλητές που ενημερώνονται μέσα στην επανάληψη:

- πλήθος_άριστων: μετράει πόσοι είναι άριστοι (μετρητής),
- άθροισμα: αθροίζει τους βαθμούς (αθροιστής),
- πλήθος_μαθητών: μετράει πόσοι είναι όλοι οι μαθητές (μετρητής)

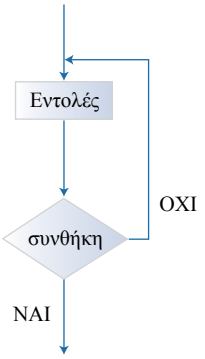
```
Αλγόριθμος Πληροφορική
πλήθος_άριστων ← 0
πλήθος_μαθητών ←0
άθροισμα ← 0
Διάβασε βαθμός
Όσο βαθμός <> -1 επανάλαβε
     Αν βαθμός > 18 τότε
          πλήθος_άριστων ←πλήθος_άριστων + 1
     τέλος_αν
     άθροισμα ← άθροισμα + βαθμός
     πλήθος_μαθητών ← πλήθος_μαθητών + 1
     Διάβασε βαθμός
τέλος_επανάληψης
Αν πλήθος_μαθητών <> 0 τότε
     μέσος_όρος ← άθροισμα / πλήθος_μαθητών
     Γράψε μέσος_όρος, πλήθος_άριστων
αλλιώς
     Γράψε 'Δε δόθηκε ούτε ένας βαθμός'
τέλος_αν
Τέλος Πληροφορική
```

Αλγόριθμος 4-8. Μέσος όρος βαθμολογίας

4.4.2. Εντολή Μέχρις_ότου

Η εντολή Μέχρις_ότου χρησιμοποιείται για την επανάληψη της ίδιας διαδικασίας μέχρι να ικανοποιηθεί μια συνθήκη.

Σύνταξη	**Αρχή_επανάληψης** εντολές **μέχρις_ότου** συνθήκη
Λειτουργία	Αρχικά, εκτελούνται οι εντολές. Μετά, ελέγχεται αν ισχύει η συνθήκη. Αν δεν ισχύει εκτελούνται ξανά οι εντολές. Η διαδικασία επαναλαμβάνεται μέχρι η συνθήκη να γίνει αληθής. Όταν ολοκληρωθεί η επανάληψη, εκτελείται η εντολή που ακολουθεί τη μέχρις_ότου.



Εικόνα 4-7. Διάγραμμα ροής της επανάληψης μέχρις_ότου

Παράδειγμα 4-8. Με το χαρτζιλίκι που μαζεύει στον κουμπαρά της, η Μάρθα θέλει να αγοράσει υπολογιστή. Η μητέρα της υποσχέθηκε την πρώτη εβδομάδα να της δώσει 25€ και κάθε εβδομάδα να της δίνει 5€ παραπάνω. Δεδομένης της τιμής του υπολογιστή, βρείτε σε πόσες εβδομάδες θα μαζέψει τα χρήματα από το χαρτζιλίκι της. Επίσης, εμφανίστε το ποσό που πιθανώς θα περισσέψει μετά την αγορά.

Στο παράδειγμα αυτό, η επανάληψη γίνεται για κάθε εβδομάδα. Το πλήθος των εβδομάδων δεν είναι γνωστό εκ των προτέρων αλλά είναι τουλάχιστον ένα (1). Ενδείκνυται η χρήση της εντολής Μέχρις_ότου

Χρησιμοποιούνται τρεις μεταβλητές που ενημερώνονται μέσα στην επανάληψη:

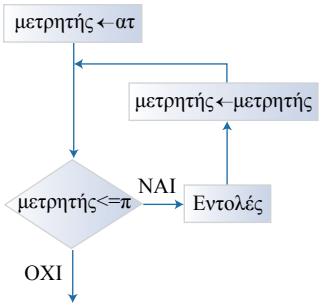
- εβδομάδα: μετράει τις εβδομάδες (μετρητής)
- κουμπαράς: αθροίζει το χαρτζιλίκι κάθε βδομάδας (αθροιστής)
- χαρτζιλίκι: αυξάνεται κάθε βδομάδα κατά 5 €

```
Αλγόριθμος Χαρτζιλίκι ( τιμήΗΥ↓ )
κουμπαράς ← 0
εβδομάδα ← 0
χαρτ ζιλίκι ← 25
Αρχή_επανάληψης
     κουμπαράς ← κουμπαράς + χαρτζιλίκι
     εβδομάδα ← εβδομάδα + 1
     χαρτζιλίκι ← χαρτζιλίκι + 5
μέχρις_ότου κουμπαράς >= τιμήΗΥ
Αν κουμπαράς > τιμήΗΥ τότε
     Γράψε 'Περίσσεψαν ', κουμπαράς - τιμήΗΥ
αλλιώς
     Γράψε 'Πάλι άδειος ο κουμπαράς!'
τέλος_αν
Τέλος Χαρτζιλίκι
```

Αλγόριθμος 4-9. Συγκέντρωση ποσού για αγορά υπολογιστή

4.4.3. Εντολή Για

Χρησιμοποιείται όταν θέλουμε να εκτελεσθεί μια διαδικασία για προκαθορισμένο πλήθος επαναλήψεων.



Εικόνα 4-8. Διάγραμμα ροής της επανάληψης Για

Σύνταξη	**Για** μετρητής **από** ατ **μέχρι** ττ **[με_βήμα** βμ**]** εντολές **Τέλος επανάληψης**
Λειτουργία	Με την εντολή Για επιτελείται η επαναληπτική εκτέλεση της ομάδας των εντολών για κάθε τιμή του μετρητή ξεκινώντας από την αρχική τιμή ατ μέχρι και την τελική τιμή ττ, μεταβαλλόμενος κάθε φορά κατά το βήμα βμ. Το βήμα μπορεί να παραληφθεί όταν ισούται με ένα. Ο αριθμός των επαναλήψεων της εντολής Για είναι προκαθορισμένος και γνωστός κατά την εκτέλεση του αλγόριθμου.

Παράδειγμα 4-9. Το τμήμα Β1 έχει 25 μαθητές. Θέλουμε να καταχωρήσουμε τα ονόματα και τους βαθμούς των μαθητών στο μάθημα «Άλγεβρα», να υπολογίσουμε το μέσο όρο τους και να εμφανίσουμε το όνομα του μαθητή που πήρε

```
Αλγόριθμος Άλγεβρα (μο↑, καλύτερος↑)
Σ ← 0
Για μαθητής από 1 μέχρι 25
     Διάβασε βαθμός, όνομα
     Αν μαθητής = 1 τότε μέγιστος — βαθμός
     Σ ← Σ + βαθμός
     Αν βαθμός > μέγιστος τότε
          μέγιστος ← βαθμός
          καλύτερος ← όνομα
τέλος_αν
τέλος_επανάληψης
μο ← Σ / 25
Γράψε μο, καλύτερος
Τέλος Άλγεβρα
```

Αλγόριθμος 4-10. Μέσος όρος και μέγιστος βαθμός

τον καλύτερο βαθμό. Μόνο ένας από τους μαθητές έχει πάρει τον καλύτερο βαθμό του τμήματος και δεν είναι σίγουρο ότι ο βαθμός είναι το 20.

Η επανάληψη γίνεται για κάθε μαθητή. Οι μαθητές είναι 25, άρα το πλήθος των επαναλήψεων είναι γνωστό και ενδείκνυται η χρήση της εντολής Για.

Η εύρεση του μέγιστου γίνεται με σύγκριση του βαθμού κάθε επόμενου μαθητή με τον μεγαλύτερο βαθμό από όλους τους προηγούμενους (μέγιστος). Η μεταβλητή καλύτερος ενημερώνεται με το αντίστοιχο όνομα.

Στην Εικόνα 4-9 συνοψίζονται οι τρεις εντολές της δομής επανάληψης.

	Όσο	**Μέχρις_ότου**	**Για**
Εκτέλεση εντολών	όσο συνθήκη αληθής (μέχρι να γίνει ψευδής)	όσο συνθήκη ψευδής (μέχρι να γίνει αληθής)	όσο μετρητής <= τελική_τιμή, με βήμα > 0 όσο μετρητής >= τελική_τιμή, με βήμα < 0
Αριθμός επαναλήψεων	άγνωστος εκ των προτέρων	άγνωστος εκ των προτέρων	προκαθορισμένος
Ελάχιστος αριθμός επαναλήψεων	0	1	0

Εικόνα 4-9. Σύγκριση των δομών επανάληψης

4.4.4 Εμφωλευμένες δομές επανάληψης

Οποιαδήποτε δομή επανάληψης μπορεί να εμφωλευτεί σε μια άλλη. Η ιδέα είναι απλή. Θέλουμε να επαναλάβουμε μια διαδικασία. Αν για παράδειγμα στον αλγόριθμο "Άλγεβρα, θέλουμε να κάνουμε την ίδια εργασία όχι μόνο για το τμήμα Β1 αλλά και για τα πέντε τμήματα της τάξης Β τότε η επανάληψη Για μαθητής από 1 μέχρι 25 θα εμφωλευτεί σε μια νέα επανάληψη Για τμήμα από 1 μέχρι 5.

Παράδειγμα 4-10. Να εκτυπωθεί η προπαίδεια του 1 και να επεκταθεί για όλους τους αριθμούς από 1 μέχρι 10.

```
Αλγόριθμος ΠροπαίδειαΤου1
Για α από 1 μέχρι 10
        Γράψε α,'*',1,'=',α*1
Τέλος_επανάληψης
Τέλος ΠροπαίδειαΤου1
```

Αλγόριθμος 4-11. Εκτύπωση της προπαίδειας του αριθμού 1

```
Αλγόριθμος Προπαίδεια
Για β από 1 μέχρι 10
      Για α από 1 μέχρι 10
          Γράψε α,'*',β,'=',α*β
      Τέλος_επανάληψης
Τέλος επανάληψης
Τέλος Προπαίδεια
```

Αλγόριθμος 4-12. Εκτύπωση της προπαίδειας των αριθμών 1 ως 10

Ας κλείσουμε την δομή επανάληψης, με τον αλγόριθμο που υλοποιεί την προσεγγιστική μέθοδο του Newton για τον υπολογισμό της τετραγωνικής ρίζας που αναφέρθηκε στην αρχή του κεφαλαίου.

Παράδειγμα 4-11. Να υπολογιστεί η τετραγωνική ρίζα πραγματικού αριθμού με δεδομένη ακρίβεια.

```
Αλγόριθμος ΤετραγωνικήΡίζα   (x↓,  ακρίβεια↓, τρ↑)
νέα_προσέγγιση ← x /10
Αρχή_επανάληψης
     τρ ← νέα προσέγγιση
     νέα_προσέγγιση ← (τρ + x /   τρ) / 2
     ΑπΤιμή1(νέα προσέγγιση  - τρ , διαφορά)
μέχρις ότου διαφορά < ακρίβεια
τρ ← νέα προσέγγιση
Τέλος ΤετραγωνικήΡίζα
```

Αλγόριθμος 4-13. Προσεγγιστικός υπολογισμός της τετραγωνικής ρίζας αριθμού

Όταν οι τιμές της ρίζας του x που υπολογίζονται σε δύο διαδοχικές επαναλήψεις διαφέρουν λιγότερο από την επιθυμητή ακρίβεια τερματίζει ο βρόχος. ΑπΤιμή1 είναι ο αλγόριθμος του Παραδείγματος 2.4

Αναδρομή

Η αναδρομή δίνει τη δυνατότητα να διατυπώσουμε αλγόριθμους που περιέχουν βρόχους, εναλλακτικά χωρίς βρόχους. Αποδεικνύεται ότι κάθε μη αναδρομικός αλγόριθμος μπορεί να μετασχηματισθεί σε ισοδύναμο αναδρομικό και κάθε αναδρομικός σε ισοδύναμο μη αναδρομικό. Στο παρακάτω παράδειγμα, βλέπουμε στην πράξη, την επίλυση ενός προβλήματος με δομή επανάληψης και με αναδρομή.

Παράδειγμα 4-12. Να υπολογιστεί το παραγοντικό θετικού ακέραιου αριθμού ν, το οποίο ορίζεται ως ν! = 1 *2* .*ν.

Κατά μια έννοια, ο αναδρομικός αλγόριθμος είναι επαναληπτική διαδικασία αφού ξανακαλεί τον εαυτό του μέχρι να γίνει ν = 1. Ωστόσο, δεν περιέχει δομή επανάληψης!

```
Αλγόριθμος ΠαραγΑ(ν↓, Π↑)
Αν ν=1 τότε
     Π ← 1
Αλλιώς
     ΠαραγΑ(ν-1, Π)
     Π ← Π * ν
Τέλος_αν
Τέλος ΠαραγΑ
```

```
Αλγόριθμος ΠαραγΕ(ν↓, Π↑)
Π ← 1
Για i από 1 μέχρι ν
     Π ← Π * i
Τέλος επανάληψης
Τέλος ΠαραγΕ
```

Αλγόριθμος 4-14. Υπολογισμός του παραγοντικού, (α) με αναδρομή, (β) χωρίς αναδρομή

Ερωτήσεις - θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Η εντολή επανάληψης ΟΣΟ ... ΕΠΑΝΑΛΑΒΕ εκτελείται τουλάχιστον μία φορά.
2. Η ψευδογλώσσα υποστηρίζει τρεις εντολές επανάληψης, την εντολή ΟΣΟ, την εντολή ΜΕΧΡΙΣ_ΟΤΟΥ και την εντολή ΓΙΑ.
3. Στην εντολή ΓΙΑ ο βρόχος επαναλαμβάνεται για προκαθορισμένο αριθμό επαναλήψεων.
4. Στην επαναληπτική δομή Όσο ... επανάλαβε δεν γνωρίζουμε εκ των προτέρων το πλήθος των επαναλήψεων.
5. Η εντολή επανάληψης ΓΙΑ ... ΑΠΟ ... ΜΕΧΡΙ ... ΜΕ_ΒΗΜΑ μπορεί να χρησιμοποιηθεί, όταν έχουμε άγνωστο αριθμό επαναλήψεων.
6. Η εντολή επανάληψης ΜΕΧΡΙΣ_ΟΤΟΥ εκτελείται υποχρεωτικά τουλάχιστον μία φορά.
7. Ο βρόχος Για Κ από 5 μέχρι 1 εκτελείται 5 φορές.
8. Κάθε βρόχος που υλοποιείται με την εντολή ΟΣΟ ... ΕΠΑΝΑΛΑΒΕ μπορεί να γραφεί και με χρήση της εντολής ΓΙΑ ... ΑΠΟ ... ΜΕΧΡΙ.
9. Ένα τμήμα αλγορίθμου που εκτελείται επαναληπτικά αποκαλείται βρόχος.
10. Ο βρόχος Για Κ από 5 μέχρι 5 δεν εκτελείται καμία φορά.

Β. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με το κατάλληλο παράδειγμα εντολών της στήλης Β

Στήλη Α **Στοιχεία ψευδογλώσσας**	**Στήλη Β** **Παραδείγματα εντολών**
1. εντολή εκχώρησης 2. δομή επιλογής 3. δομή επανάληψης	α. Αν α = 1 τότε Εμφάνισε ‘‘Α’’ αλλιώς Εμφάνισε ‘‘πρώτος” Τέλος_αν
	β. Οσο Χ < 0 επανάλαβε Χ <-- Χ - 1 Τέλος_επανάληψης
	γ. α <-- β + 1
	δ. Αρχή_επανάληψης Ι <-- Ι - 1 Μέχρις_ότου Ι < 0
	ε. Αν Χ = 2 τότε Χ <-- Χ/2 Τέλος_αν
	στ. Για k από 9 μέχρι 5 με_βήμα -1 Εμφάνισε 3*k Τέλος_επανάληψης

Γ. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με το κατάλληλο της στήλης Β

Στήλη Α **Εντολές**	**Στήλη Β** **Προτάσεις**
1. Οσο συνθήκη επανάλαβε εντολές Τέλος_επανάληψης 2. Για μετρητής από ατ μεχρι ττ με_βήμα βμ εντολές Τέλος_Επανάληψης 3. Αρχή_επανάληψης εντολές Μέχρι_ότου συνθήκη	α. Ο βρόχος επανάληψης τερματίζεται όταν η συνθήκη είναι αληθής β. Ο βρόχος επανάληψης τερματίζεται όταν η συνθήκη είναι ψευδής γ. Ο βρόχος επανάληψης εκτελείται οπωσδήποτε μία φορά δ. Ο βρόχος επανάληψης είναι δυνατό να μην εκτελεστεί ε. Ο βρόχος επανάληψης εκτελείται για συγκεκριμένο πλήθος επαναλήψεων

Δ. Έστω τμήμα αλγορίθμου με μεταβλητές A, B, C, D, X και Y.

D <-- 2
Για X από 2 μέχρι 5 με_βήμα 2
A <-- 10*X
B <-- 5*X + 10
C <-- A + B - (5*X)
D <-- 3*D - 5
Y <-- A + B - C + D
Τέλος_επανάληψης

Να βρείτε τις τιμές των μεταβλητών A, B, C, D, X και Y σε όλες τις επαναλήψεις. Να γίνει και το Διαγραμμα ροής

Ε. Πόσες φορές θα εκτελεστούν και τι θα εμφανίσουν οι παρακάτω επαναλήψεις:

Για i από 1 μέχρι 3
    Εμφάνισε i+3
Τέλος_επανάληψηςy <-- 8
Αρχή_επανάληψης
    y <-- y + 1
    Εμφάνισε y
Μέχρις_ότου y >= 7
y <-- 3
Όσο y Φ 8 επανάλαβε
    Εμφάνισε y
    y <-- y + 2
Τέλος_επανάληψης

ΣΤ. Δίνονται οι παρακάτω αλγόριθμοι να μετατραπούν σε διαγράμματα ροής και στη συνέχεια να παρουσιαστούν οι πίνακες τιμών

1.

```
Αλγόριθμος Μετατροπή
    Χ <—— 0
    Για Κ από 1 μέχρι 10
        Διάβασε Λ
        Αν Λ > 0 τότε
            Χ <— Χ + Λ
        αλλιώς
            Χ <— Χ - Λ
        Τέλος_αν
    Τέλος_επανάληψης
    Εμφάνισε Χ
Τέλος Μετατροπή
```

```
2.
Αλγόριθμος Αριθμοί
     A <-- 1
     B <-- 1
     N <-- 0
     M <-- 2
     Όσο B < 6 επανάλαβε
     X <-- A + B
           Αν X MOD 2=0 τότε
                 N <-- N + 1
           αλλιώς
                 M <-- M + 1
           Τέλος_αν
           A <— B
           B <— X
           Εμφάνισε N, M, B
     Τέλος_επανάληψης
     Εμφάνισε X
Τέλος Αριθμοί

3.
X <-- 2
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
     Y <-- X DIV 2
     Z <-- A_M(X/3)
     ΑΝ Z > 0 ΤΟΤΕ
           A<-- Z
     ΑΛΛΙΩΣ
           A<-- Y
     ΤΕΛΟΣ_ΑΝ
     ΓΡΑΨΕ X, Y, Z, A
     X <-- X + 3
ΜΕΧΡΙΣ_ΟΤΟΥ X > 10
```

Ασκήσεις - Προβλήματα

1. Επεκτείνετε τον αλγόριθμο Μέγιστος_β (Παράδειγμα 4-4) ώστε να βρίσκει το μεγαλύτερο από τρεις αριθμούς. Γενικεύστε για οποιοδήποτε πλήθος αριθμών.
2. Μπορεί μια δομή πολλαπλής επιλογής να μετατραπεί σε ισοδύναμη με τη χρήση ανεξάρτητων δομών απλής επιλογής;
3. Ποια από τις τρεις δομές αλγορίθμου ή ποιο συνδυασμό αυτών θα χρησιμοποιούσατε για τα παρακάτω προβλήματα: α) υπολογισμός γενικού μέσου όρου μαθητή σε δέκα μαθήματα β) προσαρμογή προφορικού βαθμού ώστε να μη διαφέρει από τον γραπτό πάνω από τρεις μονάδες γ) υπολογισμός ποσοστού άριστων μαθητών για 10 τμήματα με 20 μαθητές το καθένα. δ) υπολογισμός πηλίκου δύο αριθμών ε) εύρεση του ψηλότερου από 10 παίκτες μιας ομάδας μπάσκετ.
4. Κατασκευάστε το διάγραμμα ροής του αλγορίθμου ΧαρακτηρισμόςΑτόμου (Παράδειγμα 4-5). Ξαναγράψτε τον αλγόριθμο σε Ψευδογλώσσα χρησιμοποιώντας εμφωλευμένες δομές σύνθετης επιλογής.

Κατασκευάστε το διάγραμμα ροής του νέου αλγορίθμου.

5. Γράψτε αλγόριθμο σε Ψευδογλώσσα που δέχεται 100 αριθμούς από τον χρήστη και τους εμφανίζει α) με τη σειρά που δόθηκαν (επαναληπτικά) β) με σειρά αντίθετη από αυτή που δόθηκαν (αναδρομικά).
6. Γράψτε αλγόριθμο ο οποίος διαβάζει τον βαθμό ενός μαθητή (από 0 έως 20). Αν ο βαθμός είναι εκτός των επιτρεπτών ορίων τότε ζητείται νέος αριθμός. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να δοθεί σωστός αριθμός. Υλοποιήστε τον αλγόριθμο και με εντολή Όσο και με Μέχρις_ότου.
7. Γράψτε αλγόριθμο ο οποίος δέχεται αριθμούς και τυπώνει το μέσο όρο των αριθμών που δόθηκαν. Η εισαγωγή αριθμών σταματά όταν δοθεί το 0 ή αρνητικός αριθμός.
8. Γράψτε αλγόριθμο ο οποίος δέχεται αριθμούς και τυπώνει το μέσο όρο των αριθμών που δόθηκαν. Η εισαγωγή αριθμών σταματά όταν το άθροισμά τους ξεπεράσει το 1000.
9. Γράψτε αλγόριθμο ο οποίος δέχεται 100 αριθμούς και τυπώνει το μέσο όρο των αριθμών που δόθηκαν.
10. Να αναπτύξετε αλγόριθμο που δέχεται το πολύ 100 αριθμούς από τους οποίους το πολύ 20 θα είναι αρνητικοί, και τυπώνει το ποσοστό των θετικών και των αρνητικών αριθμών που εισήχθησαν. (το μηδέν δεν είναι ούτε θετικός ούτε αρνητικός)
11. Ασανσέρ έχει όριο βάρους 350 κιλά. Γράψτε αλγόριθμο ο οποίος δέχεται επαναληπτικά την εισαγωγή του βάρους κάθε ατόμου που θέλει να μπει στο ασανσέρ και σταματά όταν παραβιάζεται το όριο βάρους. Στο τέλος, τυπώνει το πλήθος και το συνολικό βάρος των ατόμων που μπήκαν στο ασανσέρ.
12. Υλοποιήσετε τον αλγόριθμο του Ευκλείδη για τον υπολογισμό του μέγιστου κοινού διαιρέτη α) με δομή επανάληψης β) με αναδρομή.

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να διακρίνετε και να περιγράφετε τα βασικά χαρακτηριστικά των εντολών επανάληψης που χρησιμοποιούνται σε έναν αλγόριθμο
- να αναγνωρίζετε από την εκφώνηση του προβλήματος, ποιες αλγοριθμικές δομές και με ποιον συνδυασμό, απαιτούνται για την επίλυση ενός δοσμένου προβλήματος
- να συνδυάζετε, σε σειρά ή εμφωλευμένα, διάφορες αλγοριθμικές δομές για την επίλυση υπολογιστικών προβλημάτων

4.5 Συντακτικά και λογικά λάθη

Στη διάρκεια συγγραφής αλγορίθμων διαπιστώνονται λάθη τα οποία οφείλονταν στη λανθασμένη σύνταξη εντολών, στη λάθος αποτύπωση εκφράσεων λόγω λανθασμένης χρήσης τελεστών, στη χρήση δεσμευμένων λέξεων ως ονόματα μεταβλητών, ασυμβατότητες στους τύπους δεδομένων κ.ά. Όλα αυτά τα λάθη προκαλούσαν αδυναμία μετάφρασης του προγράμματος με αποτέλεσμα να εντοπίζονται με σχετική ευκολία. Τέτοιου είδους λάθη ονομάζονται **συντακτικά λάθη**.

Εκτός όμως, από τα συντακτικά υπάρχουν και τα **λογικά λάθη** τα οποία εμφανίζονται στη διάρκεια εκτέλεσης του αλγορίθμου και οφείλονται σε σφάλματα κατά την υλοποίηση. Αυτού του είδους τα λάθη παράγουν ανεπιθύμητη έξοδο (λανθασμένη λύση) ή ανεπιθύμητη συμπεριφορά και συχνά είναι δύσκολο να εντοπιστούν. Η εργασία ελέγχου αφορά στον έλεγχο του αλγορίθμου με δοκιμαστικά δεδομένα, με τη μορφή πίνακα τιμών.

Εκτέλεση αλγορίθμου με πίνακα τιμών

Ο προγραμματιστής εκτελεί τον αλγόριθμο στο χαρτί με εικονικά δεδομένα, εξάγει τα αποτελέσματα και συγκρίνει τα δικά του αποτελέσματα με τα αναμενόμενα.

Παράδειγμα 4-13. Αλγόριθμος αντιμετάθεσης μεταβλητών. Μια διαδικασία πολύ χρήσιμη στον προγραμματισμό είναι η ανταλλαγή των τιμών δύο μεταβλητών. Σκοπός της αντιμετάθεσης των μεταβλητών x και y, είναι να βρεθεί η αρχική τιμή του x στη μεταβλητή y και το αντίθετο. Η λύση θα έλεγε κανείς ότι είναι απλή. Βάλε την τιμή του x στο y και μετά βάλε την τιμή του y στο x.

Ας δούμε τον πίνακα τιμών του αλγόριθμου αυτού. Έστω ότι αρχικά x=5 και y=6. Δίνοντας στο x την τιμή του y, η αρχική τιμή του x χάνεται. Και οι δυο μεταβλητές έχουν την ίδια τιμή, την τιμή του y!

Η λύση βρίσκεται στη χρήση τρίτης βοηθητικής μεταβλητής, όπου αποθηκεύεται προσωρινά η τιμή του x πριν αλλάξει. Στο σχήμα που ακολουθεί, φαίνεται η χρονική σειρά των εντολών που οδηγούν στην αντιμετάθεση των x, y.

Ερωτήσεις - θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Τα λογικά λάθη είναι συνήθως λάθη σχεδιασμού και δεν προκαλούν τη διακοπή της εκτέλεσης του αλγορίθμου
2. Η παράλειψη μιας εντολής Τέλος_αν είναι λογικό λάθος
3. Η χρήση της εντολής ΜΟ ← α + β + γ / 3 αντί της ΜΟ ← (α + β + γ) / 3 είναι λογικό λάθος
4. Λογικά λάθη είναι τα λάθη που οφείλονται στη λανθασμένη σύνταξη αλγορίθμου
5. Συντακτικά λάθη εμφανίζονται στη διάρκεια εκτέλεσης του αλγορίθμου

Β. Δίνονται οι παρακάτω λανθασμένες εντολές για τον υπολογισμό του μέσου όρου δύο αριθμών:

1. Γ ← Α + Β / 2 2. Γ ← (Α + Β / 2
3. Γ ← (Α + Β / 2) 4. Γ ← (Α + Β) : 2

Να εντοπίσετε το συντακτικό ή το λογικό λάθος της κάθε εντολής

Ασκήσεις - Προβλήματα

1. Ελέγξτε αν ο παραπάκω αλγόριθμος κάνει αντιμετάθεση των μεταβλητών x, y.

```
Αλγόριθμος ΤσωςΑντιμετάθεση(x↑↓, y↑↓)
        x ← x + y
        y ← x - y
        x ← x - y
Τέλος ΤσωςΑντιμετάθεση
```

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να περιγράφετε τα είδη λαθών ενός αλγορίθμου
- να εκτελείτε έναν αλγόριθμο κάνοντας πίνακα τιμών
- να εντοπίζετε και να διορθώνετε λογικά λάθη σε απλά παραδείγματα αλγορίθμων

4.6 Τεκμηρίωση

Στους αλγόριθμους που είδαμε διαπιστώσαμε ότι θα ήταν πιο κατανοητοί αν υπήρχαν κάποια σχόλια τα οποία να διευκρινίζουν τις λειτουργίες τους. Τα σχόλια σε έναν αλγόριθμο εισάγονται με το θαυμαστικό (!) και συνήθως τοποθετούνται στις επικεφαλίδες, σε ομάδες εντολών, σε δομές ελέγχου. Τοποθετούνται επίσης για διευκρινήσεις, αλλά και στα ονόματα των αναγνωριστικών. Η λογική της χρήσης των σχολίων είναι ο αλγόριθμος να μπορεί να γίνει κατανοητός από οποιονδήποτε προγραμματιστή.

```
Αλγόριθμος μέγιστος
!Υπολογισμός μεγίστου τριών αριθμών
Διάβασε x,y,z        ! Εισαγωγή τριών αριθμών
Αν x>=y τότε !   Σύγκριση των δύο αριθμών
      Max← x !   μέγιστος είναι ο πρώτος
Αλλιώς
      Max← y !    μέγιστος είναι ο δεύτερος
τέλος_αν ! βρήκα το μέγιστο των δύο πρώτων
Αν z> Max τότε       ! Σύγκριση τρίτου με το μέγιστο των δύο

       Max←z !    Μέγιστος όλων είναι ο τρίτος
τέλος_αν     !    Αλλιώς κράτα τον προηγούμενο μέγιστο
Γράψε Max
Τέλος μέγιστος
```

Αλγόριθμος 4-15. Τεκμηρίωση με σχόλια στον αλγόριθμο υπολογισμού του μέγιστου

Ερωτήσεις - θέματα για συζήτηση

Α. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με m κατάλληλο στοιχείο της στήλης Β

Στήλη Α	Στήλη Β
α. εγχειρίδιο χρήσης	1. Ηλεκτρονικό εγχειρίδιο
β. εγχειρίδιο αναφορών	2. Πληροφορίες για την αποτελεσματική χρήση του συστήματος
γ. σύστημα άμεσης βοήθειας	3. Περιγραφή των λειτουργιών του συστήματος, ταξινομημένες αλφαβητικά

Β. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με m κατάλληλο στοιχείο της στήλης Β

Στήλη Α	Στήλη Β
Α. Τεχνική τεκμηρίωση	1. Λεπτομερή περιγραφή του προβλήματος
	2. Πλαίσιο σχολίων
	3. Αλγόριθμοι
	4. Δομές Δεδομένων
Β. Εσωτερική τεκμηρίωση	5. Διευκρινήσεις,
	6. Πίνακες Δοκιμών
	7. Λειτουργικά σχόλια
	8. Ονόματα Αναγνωριστικών

Γ Τοποθετήστε σχόλια στον παρακάτω αλγόριθμο ώστε να γίνεται κατανοητός:

```
ΑΛΓΟΡΙΘΜΟΣ ΑΣΚΗΣΗ
     ΓΡΑΨΕ "Δώσε ακτίνα"
     ΔΙΑΒΑΣΕ R
     L ← 2*3.14←R
     A ← 3.14*R^2
     ΓΡΑΨΕ " περίμετρος:", L
     ΓΡΑΨΕ " εμβαδό:", A
ΤΕΛΟΣ ΑΣΚΗΣΗ
```

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να αναγνωρίζετε τη σημασία της τεκμηρίωσης σε έναν αλγόριθμο
- να εντοπίζετε και να περιγράφετε τη λειτουργία σχολίων σε απλά παραδείγματα αλγορίθμων

κεφάλαιο 5

Θέματα Εφαρμοσμένης Επιστήμης των Υπολογιστών

Δεδομένα & Δομές Δεδομένων

Στόχοι

Στόχοι του κεφαλαίου είναι οι μαθητές:

- Να συνδέσουν την έννοια των υπολογιστικών δεδομένων με τα αντίστοιχα πραγματικά δεδομένα ενός προβλήματος
- Να αναφέρουν τους βασικούς τύπους δεδομένων
- Να αντιληφθούν τη χρησιμότητα των δομών δεδομένων
- Να μπορούν να διακρίνουν τις στατικές από τις δυναμικές δομές δεδομένων
- Να υλοποιούν απλούς αλγορίθμους με τη βοήθεια πινάκων

Λέξεις κλειδιά

Δεδομένα και Δομές Δεδομένων	Δεδομένα, τύποι δεδομένων, δομές δεδομένων, στατικές δομές, δυναμικές δομές, πίνακας, διάσταση πίνακα, δείκτης πίνακα, αναζήτηση στοιχείου, ταξινόμηση

Εισαγωγικές ερωτήσεις

1. Γνωρίζεις με ποιον τρόπο αποθηκεύονται τα δεδομένα στη μνήμη ενός υπολογιστή;
2. Πώς πιστεύεις ότι διαχειρίζεται ο υπολογιστής το μεγάλο πλήθος δεδομένων του προγράμματος μαθητολογίου του σχολείου σου;
3. Γιατί ο τηλεφωνικός κατάλογος έχει τα ονόματα των συνδρομητών με αλφαβητική σειρά;
4. Περίγραψε τον τρόπο με τον οποίο αναζητάς γρήγορα το τηλέφωνο ενός φίλου σου σε έναν έντυπο τηλεφωνικό κατάλογο.

5. Δεδομένα & Δομές Δεδομένων

5.1 Τα δεδομένα και η αναπαράστασή τους

Η έννοια των δεδομένων είναι πολύ βασική για την Πληροφορική. Τα δεδομένα προέρχονται από τον χώρο των προβλημάτων. Μπορούμε να πούμε ότι τα δεδομένα αποτελούν μια «αφαίρεση» του πραγματικού κόσμου. Για παράδειγμα, ένας μαθητής είναι μια οντότητα με μεγάλο πλήθος χαρακτηριστικών και ιδιοτήτων, όπως: ονοματεπώνυμο, όνομα πατρός, ημερομηνία γέννησης, αγαπημένη ομάδα, αγαπημένο φαγητό, ενδιαφέροντα, φίλοι κ.ά. Για το μαθητολόγιο του σχολείου είναι απαραίτητη η αναπαράσταση μόνο ενός μικρού μέρους από τα παραπάνω χαρακτηριστικά, όπως το ονοματεπώνυμο, το όνομα πατρός και η ημερομηνία γέννησης του μαθητή. Η Εικόνα 5-1 δείχνει την αντιστοίχηση ανάμεσα σε οντότητητες και έννοιες του πραγματικού κόσμου με έννοιες της Πληροφορικής, με ένα παράδειγμα.

Πραγματικός κόσμος	**Πληροφορική**	**Παράδειγμα**
Οντότητες (αντικείμενα)	Δεδομένα	Μαθητής, Βαθμός
Σχέσεις οντοτήτων	Συσχετίσεις δεδομένων	Ο μαθητής έχει βαθμούς
Ενέργειες	Αλγόριθμοι	Ο μέσος όρος υπολογίζεται αθροίζοντας τους βαθμούς και διαιρώντας με το πλήθος τους

Εικόνα 5-1. Αντιστοίχηση του πραγματικού κόσμου με έννοιες της Πληροφορικής

Ο μαθητής και ο βαθμός του αποτελούν οντότητες του πραγματικού κόσμου. Η Πληροφορική αναπαριστά αυτές τις οντότητες ως δεδομένα. Στον πραγματικό κόσμο ο μαθητής έχει βαθμούς. Η Πληροφορική συσχετίζει τα δεδομένα του μαθητή με τα δεδομένα "βαθμοί". Ο μέσος όρος βαθμολογίας του μαθητή υπολογίζεται από τα δεδομένα των βαθμών του με συγκεκριμένο τρόπο. Ο υπολογισμός αυτός είναι μια ενέργεια που θα υλοποιηθεί από έναν αλγόριθμο.
Στόχος της Πληροφορικής είναι να μελετήσει, να αναπαραστήσει και να επεξεργαστεί τα δεδομένα για να δημιουργήσει πληροφορία και γνώση που θα αξιοποιηθούν στην επίλυση προβλημάτων.

Δεδομένα (data) είναι τα στοιχεία που αναπαρίστανται σε αλγορίθμους και προγράμματα και υφίστανται επεξεργασία.

Όλα τα στοιχεία που θα αναπαρασταθούν σε έναν αλγόριθμο ή ένα πρόγραμμα ονομάζονται δεδομένα, είτε είναι είσοδοι είτε έξοδοι (αποτελέσματα).

Θα πρέπει εδώ να σημειωθεί ότι η έννοια των δεδομένων στην Πληροφορική δεν είναι ταυτόσημη με την έννοια των δεδομένων μιας εκφώνησης σε ένα πρόβλημα π.χ. Μαθηματικών ή Φυσικής, αλλά είναι πιο γενική. Έστω για παράδειγμα το απλό πρόβλημα υπολογισμού του εμβαδού τριγώνου που έχει βάση β και ύψος υ. Τα στοιχεία που σχετίζονται με το πρόβλημα είναι η βάση, το ύψος,

Ο αριθμός 27 παριστάνεται ως 00011011, ενώ το λατινικό γράμμα Α παριστάνεται ως 01000001. Το 27 με 2 bytes θα παριστάνονταν ως 0000000000011011.

αλλά και το εμβαδόν του τριγώνου. Τα στοιχεία αυτά είναι τα δεδομένα που θα αναπαρασταθούν στον αλγόριθμο υπολογισμού του εμβαδού.

Όλα τα δεδομένα αποθηκεύονται στον υπολογιστή. Τα κυκλώματα των ψηφιακών υπολογιστών λειτουργούν σε δύο καταστάσεις, 0 και 1. Το 0 και το 1 ονομάζονται **δυαδικά ψηφία** ή **bits** (**bi**nary digi**ts**). Άρα, τα ψηφιακά δεδομένα είναι σειρές από 0 και 1. Τα δυαδικά ψηφία ομαδοποιούνται σε **οκτάδες** που ονομάζονται **bytes**. Ένα byte μπορεί να παριστάνει ένα γράμμα, έναν αριθμό, ένα σύμβολο ή ένα τμήμα αυτών. Οι ακέραιοι αριθμοί συνήθως αναπαρίσταται με 4 ή 8 bytes. Το πλήθος των bits που χρησιμοποιείται για να αναπαρασταθούν τα δεδομένα καθορίζει το εύρος τους.

Οι γλώσσες προγραμματισμού υψηλού επιπέδου δίνουν την δυνατότητα χρήσης *τύπων δεδομένων* που αποκρύπτουν από τον προγραμματιστή την αναπαράσταση σε bits και bytes. Ο **τύπος δεδομένων** (data type) σε μια γλώσσα προγραμματισμού αναφέρεται σε ένα είδος δεδομένων που μπορεί να παραστήσει η γλώσσα. Ένας τύπος δεδομένων καθορίζεται από το *όνομά* του, το *είδος* και το *εύρος* των τιμών του καθώς και από τις *πράξεις* που μπορούν να εφαρμοστούν σε αυτόν. Οι τύποι δεδομένων διακρίνονται σε **απλούς** και σε **σύνθετους**.

Στο προηγούμενο κεφάλαιο είδαμε τη χρήση των απλών τύπων δεδομένων στα πλαίσια της κωδικοποίησης ενός αλγορίθμου. Όπως ακριβώς και στη ψευδογλώσσα που χρησιμοποιήσαμε για τις ανάγκες του βιβλίου, έτσι και στις πραγματικές γλώσσες προγραμματισμού οι τύποι δεδομένων που υπάρχουν στις περισσότερες από αυτές είναι ο **ακέραιος**, ο **πραγματικός**, ο **λογικός** και ο **χαρακτήρας**. Η Εικόνα 5-2 συνοψίζει τα χαρακτηριστικά των τεσσάρων αυτών απλών τύπων δεδομένων.

Τύπος δεδομένων	Είδος τιμών	Παραδείγματα τιμών	Βασικές πράξεις
Ακέραιος	Ακέραιοι αριθμοί με πρόσημο	0, 37, -156	Σύγκριση, πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση
Πραγματικός	Πραγματικοί αριθμοί με υποδιαστολή	1.67, -0.0345, 3.14159	Σύγκριση, πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση
Λογικός	Οι δύο λογικές τιμές Ψευδής και Αληθής	Ψευδής, Αληθής	Σύγκριση
Χαρακτήρας	Σύμβολα	‘A’, ‘a’, ‘γ’, ‘@’, ‘8’, ‘€’	Σύγκριση

Εικόνα 5-2. Απλοί τύποι δεδομένων

Οι απλοί τύποι δεδομένων χρησιμοποιούνται σε απλά προβλήματα, όπως το πρόβλημα υπολογισμού του εμβαδού τριγώνου. Η βάση και το ύψος του τριγώνου θα αναπαρασταθούν με τον τύπο δεδομένων “πραγματικός”, εφόσον οι διαστάσεις είναι πραγματικοί αριθμοί. Το εμβαδόν του τριγώνου θα αναπαρασταθεί επίσης με τον τύπο δεδομένων “πραγματικός”, εφόσον προκύπτει από υπολογισμό που περιέχει τη βάση και το ύψος. Στο μαθητολόγιο παριστάνονται

δεδομένα όπως το φύλο, η τάξη και το τμήμα του μαθητή, οι προφορικοί βαθμοί και ο τελικός μέσος όρος και ο χαρακτηρισμός της φοίτησής του. Στους ακόλουθους πίνακες (Εικόνα 5-3, Εικόνα 5-4), περιγράφονται οι τύποι δεδομένων για τα στοιχεία του μαθητολογίου. Ο προφορικός βαθμός είναι ακέραιος αριθμός με τιμές από 0 ως 20. το φύλο του μαθητή έχει την τιμή Α (Άρρεν) ή Θ (Θήλυ), άρα μπορεί να παρασταθεί ως χαρακτήρας. Η τάξη Α, Β ή Γ Λυκείου επίσης μπορεί να παρασταθεί με έναν χαρακτήρα, ενώ το τμήμα είναι ακέραιος αριθμός με τιμές 1, 2, 3, κλπ. Ο μέσος όρος βαθμολογίας είναι πραγματικός αριθμός, π.χ. 18,7. Ο χαρακτηρισμός της φοίτησης ως επαρκούς είναι Αληθής αν η φοίτηση είναι επαρκής και Ψευδής στην αντίθετη περίπτωση, άρα είναι δεδομένο λογικού τύπου.

Δεδομένα προβλήματος	**Είδος και εύρος αποδεκτών τιμών**	**Τύπος δεδομένων**
Βαθμός Α' τετραμήνου	Ακέραιος από 0 ως 20	Ακέραιος
Φύλο μαθητή	'Α' ή 'Θ'	Χαρακτήρας
Τάξη μαθητή	'Α', 'Β' ή 'Γ'	Χαρακτήρας
Τμήμα μαθητή	1, 2, 3 ή 4 ...	Ακέραιος
Μέσος όρος τετραμήνων	Πραγματικός από 0 ως 20	Πραγματικός
Επαρκής φοίτηση	Αληθής ή Ψευδής	Λογικός

Εικόνα 5-3. Παραδείγματα δεδομένων του μαθητολογίου και των τύπων τους

	Α' τετράμηνο	Β' τετράμηνο	Μ.Ο. προφορικών	Γραπτός βαθμός	Μ.Ο. μαθήματος
Ιστορία	18	19	18.5	17	17.75
Μαθηματικά	17	16	16.5	18.5	17.5
Αρχαία	18	18	18	20	19

Εικόνα 5-4. Βαθμολογίες ενός μαθητή σε τρία μαθήματα

Σε πολλές γλώσσες προγραμματισμού υλοποιούνται σύνθετοι τύποι δεδομένων. Ένα τέτοιο παράδειγμα είναι οι συμβολοσειρές (strings) που αποτελούνται από μια σειρά χαρακτήρων για να αναπαραστήσουν δεδομένα κειμένου. Ένα παράδειγμα συμβολοσειράς είναι η λέξη «Πληροφορική». Στο παρόν βιβλίο δεν γίνεται διάκριση μεταξύ μεταβλητών τύπου χαρακτήρα και τύπου συμβολοσειράς. Έτσι για παράδειγμα η συμβολοσειρά «καλημέρα κόσμε!» θεωρείται ως μεταβλητή τύπου χαρακτήρα, χωρίς να μας απασχολεί πώς ο υπολογιστής αποθηκεύει και διαχειρίζεται αυτό το δεδομένο. Άλλωστε η τάση στις σύγχρονες γλώσσες προγραμματισμού είναι να καταργούνται οι αυστηροί ορισμοί των τύπων δεδομένων και να αφήνουμε την πρωτοβουλία στο ίδιο το πρόγραμμα για το πώς θα αποθηκεύσει τα δεδομένα του.

Ερωτήσεις - θέματα για συζήτηση

Α. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με την κατάλληλη τιμή της στήλης Β

Στήλη Α Τύπος δεδομένων	Στήλη Β Τιμές
i.Ακέραιος	1.0
ii.Πραγματικός	2.Ψευδής
iii.Λογικός	3.'$'
iv.Χαρακτήρας	4.'Β'
	5.-156
	6.1.14
	7.49
	8.'Υ'
	9.-0.0827
	10.'#'

Β. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με τον κατάλληλο τύπο δεδομένων της στήλης Β

Στήλη Α Δεδομένα Προβλήματος	Στήλη Β Τύπος Δεδομένων
a. Ύψος τριγώνου	1. Ακέραιος
b. Βάση τριγώνου	2. Πραγματικός
c. Φύλλο Επιβάτη	3. Χαρακτήρας
d. Σειρά Καθίσματος Επιβάτη	4. Λογικός
e. Αριθμός θέσης	
f. Ηλικία Επιβάτη	
g. Επιβιβάστηκε ο επιβάτης;	

Γ. Ερωτήσεις

1. Περιγράψτε ένα πρόβλημα από: α) τη Φυσική, β) την καθημερινότητά σας, γ) το σχολικό περιβάλλον, και εντοπίστε τα δεδομένα που πρέπει να αναπαρασταθούν από τον αλγόριθμο που θα λύσει το πρόβλημα.

Ασκήσεις - Προβλήματα

1. Ένας μετεωρολογικός σταθμός θέλει να εξάγει στατιστικά για τη θερμοκρασία μίας πόλης. Ποια είναι τα δεδομένα που πρέπει να συγκεντρωθούν για: α) να υπολογιστεί η ώρα κατά την οποία καταγράφεται η μέγιστη θερμοκρασία

της πόλης μέσα σε μία μέρα; β) να υπολογιστεί η μέση θερμοκρασία της πόλης στις 12:00 το μεσημέρι κατά τη διάρκεια των καλοκαιρινών μηνών;

2. Ποιον από τους βασικούς τύπους δεδομένων θα χρησιμοποιούσατε για τα παρακάτω στοιχεία: α) ηλικία, β) τιμή προϊόντος, γ) θερμοκρασία ατμόσφαιρας, δ) αν βρέχει ή όχι;

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να περιγράφετε τι είναι τα δεδομένα και ποια είναι η σχέση τους με τα προγράμματα
- να διακρίνετε πώς αναπαριστά τα δεδομένα ο υπολογιστής και πώς οι γλώσσες προγραμματισμού υψηλού επιπέδου
- να αναγνωρίζετε και να ονομάζετε τα δεδομένα συγκεκριμένων προβλημάτων
- να περιγράφετε τι είναι οι τύποι δεδομένων και να επιλέγετε κατάλληλους τύπους δεδομένων για συγκεκριμένα προβλήματα

5.2 Δομές δεδομένων

Τα περισσότερα προβλήματα που καλούμαστε να λύσουμε με έναν υπολογιστή έχουν μεγάλο πλήθος δεδομένων καθώς έτσι εκμεταλλευόμαστε και τα μεγάλα πλεονεκτήματα των υπολογιστών που είναι η ταχύτητα εκτέλεσης των πράξεων και η διαχείριση πολλών δεδομένων. Τα δεδομένα αυτά δεν είναι ασύνδετα μεταξύ τους καθώς συνήθως έχουν κοινά χαρακτηριστικά ή αναφέρονται σε μια κοινή οντότητα. Για παράδειγμα οι βαθμοί ενός μαθητή έχουν ως κοινό σημείο αναφοράς το όνομα του μαθητή στον οποίο ανήκουν. Αντίστοιχα ο κατάλογος επιβατών μιας αεροπορικής πτήσης έχουν ως κοινό στοιχείο τον αριθμό του δρομολογίου της πτήσης αυτής.

Όταν τα δεδομένα είναι σύνθετα και πρέπει να συσχετιστούν μεταξύ τους, είναι σημαντικό να οργανωθούν ώστε, αφενός μεν να λύνεται σωστά το πρόβλημα, αφετέρου δε να γίνεται αποδοτική διαχείριση της μνήμης και των υπολογιστικών πόρων. Για το λόγο αυτό δημιουργούνται οι δομές δεδομένων.

Δομή δεδομένων είναι ένας τρόπος οργάνωσης, συσχέτισης και αποθήκευσης των δεδομένων.

Στις δομές δεδομένων εφαρμόζονται λειτουργίες όπως:

- **Εισαγωγή**: ένα νέο στοιχείο δεδομένων προστίθεται στη δομή και συσχετίζεται με τα στοιχεία που ήδη υπάρχουν σε αυτή.
- **Διαγραφή**: ένα στοιχείο αφαιρείται από τη δομή.
- **Προσπέλαση της δομής**: γίνεται πρόσβαση στα στοιχεία της με συγκεκριμένη μέθοδο.
- **Αναζήτηση στοιχείου** μέσα στη δομή.
- **Ταξινόμηση**: τα στοιχεία της δομής αναδιοργανώνονται έτσι ώστε να ακολουθούν συγκεκριμένη διάταξη.

Η επιλογή των κατάλληλων δομών δεδομένων είναι μέρος της λύσης του προ-

βλήματος. Υπάρχουν δομές που είναι κατάλληλες για συγκεκριμένους τύπους προβλημάτων. Ένα από τα κριτήρια επιλογής δομών δεδομένων είναι ο τρόπος διαχείρισης της μνήμης του υπολογιστή. Με βάση αυτό το κριτήριο, οι δομές δεδομένων διακρίνονται σε δύο (2) κατηγορίες.

Οι **στατικές δομές δεδομένων** έχουν σταθερό μέγεθος το οποίο καθορίζεται κατά την δημιουργία (συγγραφή) του προγράμματος. Το μέγεθος αυτό δεν αλλάζει, επομένως οι δομές αυτές αποθηκεύουν συγκεκριμένο πλήθος δεδομένων.

Οι **δυναμικές δομές δεδομένων** δεν έχουν σταθερό μέγεθος και αυ- ξομειώνονται με τη μέθοδο της δυναμικής παραχώρησης μνήμης. Εφαρ-μόζονται σε προβλήματα όπου το μέγεθος των δεδομένων δεν είναι γνωστό από την αρχή ή δε μπορεί να εκτιμηθεί, καθώς και σε περιπτώσεις όπου είναι κρίσιμο να μην δεσμεύεται μνήμη που δε θα χρησιμοποιηθεί.

Ερωτήσεις - θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Ο πίνακας είναι μία δυναμική δομή δεδομένων.
2. Σε μία δυναμική δομή δεδομένων τα δεδομένα αποθηκεύονται υποχρεωτικά σε συνεχόμενες θέσεις μνήμης.
3. Η ταξινόμηση είναι μια από τις βασικές λειτουργίες επί των δομών δεδομένων.
4. Οι δυναμικές δομές έχουν σταθερό μέγεθος.
5. Η προσπέλαση είναι μια από τις βασικές πράξεις επί των δομών δεδομένων.

Β. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Οι τύποι δεδομένων διακρίνονται σε _____ και _____
2. Δομή δεδομένων είναι ένας τρόπος _____ , συσχέτισης και _____ των δεδομένων
3. Με βάση τον τρόπο διαχείρισης της μνήμης του υπολογιστή, οι δομές δεδομένων διακρίνονται σε δύο (2) κατηγορίες:
   α) _____ β) _____

Γ. Ερωτήσεις

1. Μπορείτε να περιγράψετε περιπτώσεις από τον πραγματικό κόσμο όπου δημιουργούνται: α) ουρές, β) στοίβες, γ) ιεραρχίες (δέντρα) και δ) γράφοι;

Ασκήσεις - Προβλήματα

1. Περιγράψτε τη δομή δεδομένων: α) για τα μηνιαία έξοδα της οικογένειάς σας, β) για το εβδομαδιαίο πρόγραμμα των μαθημάτων σας στο σχολείο, γ) για τις κινήσεις των παικτών σε ένα παιχνίδι σκακιού, δ) για την ταξινόμηση των θηλαστικών (αναζητήστε πληροφορίες σε κάποιο βιβλίο βιολογίας).

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να περιγράφετε τι είναι οι δομές δεδομένων
- να επιλέγετε στατικές ή δυναμικές δομές δεδομένων για συγκεκριμένα προβλήματα

5.3 Πίνακες

Η πιο διαδεδομένη δομή δεδομένων είναι η στατική δομή του πίνακα δεδομένων, δηλαδή έναν τρόπο οργάνωσης των δεδομένων.

Ο **πίνακας** είναι μία διάταξη στοιχείων του ίδιου τύπου με συγκεκριμένο μέγεθος, στην οποία τα στοιχεία αναφέρονται με το ίδιο όνομα και με έναν ή περισσότερους δείκτες.

Ο πίνακας είναι μία δομή δεδομένων. Άλλες δομές δεδομένων είναι οι λίστες, οι ουρές, οι στοίβες, τα δένδρα και οι γράφοι.

Οι πίνακες διατάσσουν τα στοιχεία τους σε γραμμές και στήλες. Ένας πίνακας N στοιχείων **μίας διάστασης** ή **μονοδιάστατος** αποτελείται από μία γραμμή (ή ισοδύναμα από μία στήλη) N στοιχείων. Τα στοιχεία αυτά βρίσκονται στις θέσεις 1 ως N, όπου N είναι το πλήθος των στοιχείων του πίνακα. Η θέση κάθε στοιχείου ονομάζεται **δείκτης** και είναι ένας ακέραιος με τιμές από 1 ως N. Όλα τα στοιχεία του πίνακα έχουν το ίδιο όνομα, αλλά καθένα έχει διαφορετικό δείκτη.

Η δομή του πίνακα καθορίζει ότι τα δεδομένα οργανώνονται σε συνεχόμενες θέσεις.

Έστω ότι θέλουμε να αποθηκεύσουμε τη θερμοκρασία μιας πόλης που καταγράφεται κάθε μία ώρα μέσα σε ένα 24ωρο, δηλαδή θέλουμε να αποθηκεύσουμε συνολικά 24 θερμοκρασίες. Μετά την καταγραφή και αποθήκευση των δεδομένων αυτών θέλουμε να τα επεξεργαστούμε ως εξής: να υπολογίσουμε τη μέση θερμοκρασία του 24ωρου, τη μέγιστη και την ελάχιστη. Εφόσον τα στοιχεία είναι πολλά, είναι ομοειδή και θέλουμε να τα αποθηκεύσουμε και να τα επεξεργαστούμε μαζί, είναι βολικό να χρησιμοποιήσουμε έναν πίνακα εικοσιτεσσάρων (24) πραγματικών αριθμών στον οποίον τα στοιχεία θα είναι αποθηκευμένα σε συνεχόμενες θέσεις. Όπως φαίνεται στην Εικόνα 2-28, ο πίνακας ονομάζεται Θ. Το πρώτο στοιχείο του πίνακα ονομάζεται Θ[1] ("Θ του 1", ή "Θ με δείκτη 1") και σε αυτό αποθηκεύουμε την τιμή για την πρώτη θερμοκρασία που καταγράφουμε, δηλαδή την τιμή 29. Οι δείκτες των στοιχείων του πίνακα είναι ακέραιοι αριθμοί με τιμές από 1 ως 24.Το τελευταίο στοιχείο του πίνακα ονομάζεται Θ[24] και συμφωνούμε ότι αντιστοιχεί στην τελευταία θερμοκρασία που θα καταγραφεί. Τα στοιχεία στις θέσεις 2 ως 23 συμφωνούμε ότι αντιστοιχούν στις ενδιάμεσες διαδοχικές καταγραφές θερμοκρασίας. Ο υπολογισμός της μέσης θερμοκρασία του 24ωρου γίνεται αθροίζοντας τα 24 στοιχεία του πίνακα και διαιρώντας το άθροισμα με το πλήθος τους. Η αποθήκευσή τους σε διαδοχικές θέσεις πίνακα, θα δούμε παρακάτω ότι διευκολύνει σημαντικά την υλοποίηση του αλγορίθμου.

δείκτης

	Θ
Θ[1]	29
Θ[2]	28,2
Θ[3]	27,9
Θ[4]	28,2
...	...
Θ[24]	27,9

Εικόνα 5-5. Μονοδιάστατος πίνακας Θ, 24 θερμοκρασιών

Η **διάσταση** του πίνακα καθορίζει το πλήθος των δεικτών που χρησιμοποιούνται για κάθε στοιχείο του. Ένας πίνακας **δύο διαστάσεων** ή **δισδιάστατος**, αποτελείται γενικά από M γραμμές και N στήλες, άρα από M * N στοιχεία. Κάθε στοιχείο αναφέρεται με το όνομα του πίνακα και δύο δείκτες: τον αριθμό της γραμμής και τον αριθμό της στήλης. Σε έναν πίνακα δύο διαστάσεων με όνομα Π, τα στοιχεία του αναφέρονται ως Π[i, j] όπου i ο δείκτης της γραμμής και j ο δείκτης της στήλης.

Ας επεκτείνουμε το παράδειγμα του πίνακα θερμοκρασιών καταγράφοντας τις θερμοκρασίες της πόλης ανά ώρα του 24ωρου, για μία ολόκληρη εβδομάδα.

Χρειαζόμαστε 7 * 24 = 168 στοιχεία για τα επτά 24ωρα. Μπορούμε επομένως να δημιουργήσουμε έναν πίνακα 168 στοιχείων. Μας βολεύει να διατηρήσουμε τη διάταξή τους ανά ημέρες και ώρες και γι' αυτό θα δημιουργήσουμε ένα πίνακα 168 στοιχείων δύο διαστάσεων, με όνομα Θ7, ο οποίος θα αποτελείται από 7 γραμμές και 24 στήλες. Επομένως για τον πίνακα αυτόν ισχύουν τα εξής:

- Κάθε μία από τις 7 γραμμές του πίνακα αποτελείται από εικοσιτέσσερα (24) στοιχεία.
- Κάθε μία από τις 24 στήλες του πίνακα αποτελείται από 7 στοιχεία.
- Τα στοιχεία του πίνακα ονομάζονται Θ7[κ, λ] όπου κ είναι ο δείκτης της γραμμής και λ ο δείκτης της στήλης στην οποία ανήκει το στοιχείο.Το κ μπορεί να πάρει τις ακέραιες τιμές 1 ως 7, ενώ το λ μπορεί να πάρει τις ακέραιες τιμές από 1 ως 24.

Επιπλέον, για να αποθηκεύσουμε τα δεδομένα του συγκεκριμένου προβλήματος θερμοκρασιών στον πίνακα Θ7, κάνουμε τις εξής συμβάσεις:

- Οι επτά (7) γραμμές αντιστοιχούν στα επτά (7) εικοσιτετράωρα, δηλαδή η 1η γραμμή αντιστοιχεί στο πρώτο 24ωρο της εβδομάδας, κ.ο.κ.
- Οι στήλες αντιστοιχούν στις ώρες των 24ωρων. Η 1η στήλη αντιστοιχεί στις θερμοκρασίες της 1ης ώρας όλων των 24ωρων.
- Το στοιχείο Θ7[1,1] που βρίσκεται στην 1η γραμμή και 1η στήλη του πίνακα αντιστοιχεί στην πρώτη θερμοκρασία που καταγράφουμε για το πρώτο 24ωρο.
- Το στοιχείο Θ7[κ, λ] για οποιοδήποτε κ με τιμές από 1 ως 7 και οποιοδήποτε λ με τιμές από 1 ως 24, αντιστοιχεί στη θερμοκρασία που καταγράφηκε για το κ 24ωρο την λ ώρα.

Ο πίνακας Θ7 μαζί με τα δεδομένα που περιέχει φαίνεται στην Εικόνα 5-6:

Θ7 Θ7[1,1] Θ7[1, 24]

1η ημέρα	29	28,2	27,9	28,2	...	27,9
	28	27,8	28,4	28,6	...	26,1
	...	...	...	...	...	...
	30	30,3	31	28,6	...	25
7η ημέρα	30	30,3	31	28,6	...	25

Εικόνα 5-6. Πίνακας θερμοκρασιών 24τετραώρου για 7 ημέρες

5.3.1 Βασικές λειτουργίες σε μονοδιάστατους πίνακες

Οι συνηθέστερες λειτουργίες σε πίνακες μίας διάστασης είναι το διάβασμα και η εμφάνιση των στοιχείων τους, ο υπολογισμός του αθροίσματος και του μέσου όρου, η εύρεση του μέγιστου και του ελάχιστου, η αναζήτηση και η ταξινόμηση.

Διάβασμα και εμφάνιση

Γενικά, για να διαβάσουμε όλα τα στοιχεία ενός πίνακα μίας διάστασης με σειρά από το πρώτο μέχρι το τελευταίο, χρησιμοποιούμε επανάληψη η οποία μετράει από 1 μέχρι το πλήθος των στοιχείων του πίνακα. Ο μετρητής χρησιμοποιείται

- Π[ί, j]: το στοιχείο της γραμμής i και στήλης j, 1 ≤ i ≤M, 1 ≤ j ≤N.
- Π[1, j]: το στοιχείο της γραμμής 1 και στήλης j, 1 < j <N.
- Π[ί, 1]: το στοιχείο της γραμμής i και στήλης 1, 1 < j <N.

ως δείκτης των στοιχείων του πίνακα και μεταβάλλεται ώστε να λάβει την τιμή όλων των δεικτών.

```
Αλγόριθμος ΔιάβασμαΠίνακα(A↑)
Για i από 1 μέχρι N
      Διάβασε A[i]
Τέλος_επανάληψης
Τέλος ΔιάβασμαΠίνακα
```

Αλγόριθμος 5-1. Διάβασμα στοιχείων πίνακα μίας διάστασης

Άθροισμα και μέσος όρος

Για τον υπολογισμό του αθροίσματος των στοιχείων ενός πίνακα χρησιμοποιούμε αθροιστή (π.χ. Σ), τον οποίο αρχικοποιούμε με την τιμή 0. Κατόπιν γράφουμε επανάληψη η οποία προσθέτει ένα ένα τα στοιχεία στον αθροιστή. Ο υπολογισμός του μέσου όρου (μέσης τιμής), μπορεί να γίνει μετά την άθροιση διαιρώντας το άθροισμα με το πλήθος των στοιχείων.

```
Αλγόριθμος ΆθροισμαΜΟΠίνακα( A↓ , Σ↑ , ΜΟ↑ )
Σ ← 0
Για i από 1 μέχρι N
      Σ ← Σ + A[i]
Τέλος_επανάληψης
ΜΟ ← Σ / N
Τέλος ΆθροισμαΜΟΠίνακα
```

Αλγόριθμος 5-2. Άθροισμα και μέσος όρος στοιχείων πίνακα μίας διάστασης

Ελάχιστο και μέγιστο

Για τον υπολογισμό του ελάχιστου στοιχείου χρησιμοποιούμε μια μεταβλητή στην οποία θα αποθηκεύσουμε το αποτέλεσμα, π.χ. με όνομα min (minimum).

```
Αλγόριθμος ΕλάχιστοΠίνακα( A↓ , min↑ )
min ← A[1]
Για i από 2 μέχρι N
      Αν min < A[i] τότε min ← A[i]
Τέλος_επανάληψης
Τέλος ΕλάχιστοΠίνακα
```

Αλγόριθμος 5-3. Υπολογισμός ελάχιστου

Θεωρούμε ότι αρχικά το ελάχιστο στοιχείο του πίνακα είναι το πρώτο του. Κατόπιν γράφουμε μια δομή επανάληψης. Οποτεδήποτε σε κάποια θέση του πίνακα βρεθεί ένα στοιχείο που είναι μικρότερο από το μέχρι τώρα ελάχιστο, το ελάχιστο αντικαθίσταται με αυτό το μικρότερο στοιχείο. Η διαδικασία τερματίζει, όταν ελεγχθούν όλα τα στοιχεία του πίνακα. Αντίστοιχα γίνεται ο υπολογισμός για το μέγιστο.

Αναζήτηση

Η αναζήτηση είναι μια λειτουργία που εκτελείται συχνά σε πίνακες. Το πρόβλημα της αναζήτησης είναι η εύρεση μιας δεδομένης τιμής ή ενός εύρους τιμών που ικανοποιούν κάποια συνθήκη σε έναν πίνακα. Το πρόβλημα αυτό μπορεί να λυθεί με πολλούς τρόπους. Στην πιο απλή λύση του, διατρέχουμε όλο τον πίνακα και συγκρίνουμε όλα τα στοιχεία του με την τιμή που αναζητάμε.

```
Αλγόριθμος ΑναζήτησηΒαθμών(Β↓, πλήθος↑)
πλήθος← 0
Για ι από 1 μέχρι N
      Αν A[i] >=18 τότε πλήθος ← πλήθος + 1
Τέλος_επανάληψης
Αν πλήθος<> 0 τότε
      Γράψε 'Οι μαθητές με βαθμό από 18 και πάνω είναι: ',
πλήθος
αλλιώς
      Γράψε 'Δεν υπάρχουν μαθητές'
Τέλος_αν
Τέλος ΑναζήτησηΒαθμών
```

Παράδειγμα 5-1. Εύρεση του αριθμού των μαθητών που έχουν βαθμό στην πληροφορική από 18 και πάνω

Σειριακή αναζήτηση

Ο αλγόριθμος της σειριακής αναζήτησης σε μονοδιάστατο πίνακα, ελέγχει με τη σειρά ένα προς ένα τα στοιχεία του πίνακα μέχρι να βρεθεί το στοιχείο που αναζητούμε ή να εξαντληθεί ο πίνακας.

```
Αλγόριθμος ΣειριακήΑναζήτηση(A↓, key↓, θέση↑)
θέση ← -1
i ← 1
Όσο i <= N και θέση = -1 επανάλαβε
      Αν A[i] = key τότε θέση ← i
      i ← i +1
Τέλος_επανάληψης
Αν θέση <> -1 τότε
      Γράψε 'Βρέθηκε στη θέση', θέση
αλλιώς
      Γράψε 'Δεν υπάρχει στον πίνακα'
Τέλος_αν
Τέλος ΣειριακήΑναζήτηση
```

Η μεταβλητή «θέση» χρησιμοποιείται για να αποθηκευτεί η θέση στην οποία βρέθηκε το στοιχείο. Αν όμως η τιμή της παραμείνει -1 ως το τέλος, τότε το στοιχείο δεν υπάρχει στον πίνακα.

Αλγόριθμος 5-4. Σειριακή αναζήτηση

Δυαδική αναζήτηση

Η δυαδική αναζήτηση είναι ένας ταχύτερος αλγόριθμος αναζήτησης σε ταξινομημένο πίνακα. Διαιρεί τον πίνακα σε δύο μέρη, επιλέγοντας το μεσαίο στοιχείο κάθε φορά και συγκρίνοντάς το με το στοιχείο που ψάχνουμε.

Η λογική της δυαδικής αναζήτησης μας είναι γνωστή από την αναζήτηση στον έντυπο τηλεφωνικό κατάλογο. Ο τηλεφωνικός κατάλογος περιέχει τα τηλέφωνα ταξινομημένα αλφαβητικά ως προς το επώνυμο.

```
Αλγόριθμος ΔυαδικήΑναζήτηση (A↓, key↓, θέση↑)
θέση ← 0
start ← 1
end ← N
όσο end >= start και θέση =0 επανάλαβε
     μέση ← start + ((end - start) div 2)
      Αν A[μέση] = key τότε
            θέση ← μέση
      αλλιώς_αν A[μέση] < key τότε
        start ← μέση + 1
      αλλιώς
        end ← μέση - 1
      τέλος_αν
 τέλος_επανάληψης
Τέλος ΔυαδικήΑναζήτηση
```

Αλγόριθμος 5-5. Δυαδική αναζήτηση

Έστω ότι ψάχνουμε να βρούμε το τηλέφωνο της κυρίας Τακτικάκη. Αρχικά ανοίγουμε τον κατάλογο σε κάποια τυχαία σελίδα, περίπου στη μέση του. Βλέπουμε ποιο γράμμα υπάρχει σε κείνη τη σελίδα. Αν ανοίξαμε μια σελίδα με επίθετα από 'Κ' και εμείς ψάχνουμε επίθετο από 'Τ' , πρέπει να κοιτάξουμε παρακάτω. Ανοίγουμε ξανά περίπου στη μέση, από τη σελίδα του 'Κ' που βρισκόμαστε μέχρι το τέλος του καταλόγου. Ας υποθέσουμε ότι ανοίγουμε σε μια σελίδα που περιέχει επώνυμα που αρχίζουν με το γράμμα 'Ρ'. Πρέπει να ξανανοίξουμε λίγο παρακάτω. Ξανανοίγουμε στο μισό του υπολοίπου και βλέπουμε το γράμμα 'Φ'. Άρα πρέπει ν' ανοίξουμε προς τα πίσω αυτή τη φορά. Τελικά ανοίγουμε στο 'Τ', κάνουμε ένα μικρό ξεφύλλισμα προς το 'Τα' και βρίσκουμε το τηλέφωνο που αναζητούμε.

Ταξινόμηση

Η ταξινόμηση συνίσταται στην αλλαγή της θέσης των στοιχείων του πίνακα ώστε να ακολουθούν μια συγκεκριμένη διάταξη, π.χ. αύξουσα ή φθίνουσα σειρά. Είδαμε στην προηγούμενη ενότητα ότι η ταξινόμηση παίζει πολύ σημαντικό ρόλο στην αύξηση της ταχύτητας με την οποία γίνονται οι αναζητήσεις σε έναν πίνακα καθώς σε έναν ταξινομημένο πίνακα μπορούμε να χρησιμοποιήσουμε την πολύ γρηγορότερη δυαδική αναζήτηση.

Υπάρχουν πολλοί αλγόριθμοι ταξινόμησης, όπως η **ταξινόμηση με εισαγωγή** (insertion sort), η **ταξινόμηση με επιλογή** (selection sort), η **ταξινόμηση ευθείας ανταλλαγής** ή **φυσαλίδα** (bubble sort), η **γρήγορη ταξινόμηση** (quicksort), κ.ά. Οι αλγόριθμοι αυτοί διαφέρουν στον τρόπο εξέτασης των στοιχείων και επομένως στην απόδοσή τους. Άλλοι είναι πιο γρήγοροι κι άλλοι πιο αργοί.

Ο αλγόριθμος της ταξινόμησης με εισαγωγή κατά αύξουσα σειρά λειτουργεί ως εξής: βρίσκει τη θέση που πρέπει να τοποθετηθεί (εισαχθεί) κάθε στοιχείο του πίνακα σε σχέση με τα προηγούμενά του και το εισάγει σ' αυτή τη θέση.

Η ταξινόμηση ξεκινάει από το δεύτερο στοιχείο του πίνακα (τρέχον στοιχείο) το οποίο πρέπει να τοποθετηθεί στη σωστή θέση ως προς το προηγούμενό του. Γίνεται η σύγκριση των δύο στοιχείων (2ο με 1ο) και αν το τρέχον (2ο) είναι μικρότερο από το πρώτο, ανταλλάσσουν θέσεις μεταξύ τους. Έτσι τα δύο πρώτα στοιχεία ταξινομούνται μεταξύ τους. Ο αλγόριθμος κατόπιν συνεχίζει επαναληπτικά με τα επόμενα στοιχεία του πίνακα. Κάθε στοιχείο που εξετάζεται (τρέχον στοιχείο), συγκρίνεται διαδοχικά με τα προηγούμενά του και αν το τρέχον στοιχείο είναι μικρότερο από κάποιο προηγούμενό του, ανταλλάσσεται με αυτό που συγκρίθηκε. Η ταξινόμηση του πίνακα με τον αλγόριθμο αυτόν επιτυγχάνεται σταδιακά.Ο αλγόριθμος της ταξινόμησης με εισαγωγή μπορεί να γραφτεί σε ψευδογλώσσα ως εξής:

```
Αλγόριθμος ΤαξινόμησηΜεΕισαγωγή (A↓↑)
Για i από 2 μέχρι N
      j ← i
      όσο A[j]< A[j-1] και j > 1 επανάλαβε
            t ← A[j]
            A[j] ← A[j-1]
            A[j-1] ← t
            j← j-1
      τέλος_επανάληψης
τέλος_επανάληψης
Τέλος ΤαξινόμησηΜεΕισαγωγή
```

Αλγόριθμος 5-6. Ταξινόμηση με εισαγωγή

5.3.2 Βασικές λειτουργίες σε πίνακες δύο ή περισσότερων διαστάσεων

Η επεξεργασία των στοιχείων πινάκων δύο διαστάσεων γίνεται συνήθως με εμφωλευμένες επαναλήψεις οι οποίες διασχίζουν τα στοιχεία του πίνακα γραμμή προς γραμμή ή στήλη προς στήλη. Η λειτουργία αναζήτησης στοιχείων σε πίνακα δύο διαστάσεων μπορεί επίσης να εφαρμοστεί σε μία γραμμή, σε μία στήλη ή σε ολόκληρο τον πίνακα. Εκτός από μονοδιάστατους και δισδιάστατους μπορούμε να δημιουργήσουμε και πολυδιάστατους πίνακες, στους οποίους κάθε στοιχείο διακρίνεται από περισσότερους από δύο δείκτες, εφόσον αυτό είναι αναγκαίο για το πρόβλημα.

Στο Παράδειγμα 5-2 γίνεται εφαρμογή των λειτουργιών πινάκων μίας και δύο διαστάσεων σε ένα πραγματικό πρόβλημα με θερμοκρασίες.

```
Αλγόριθμος Θερμοκρασίες
Για i από 1 μέχρι 7
     Για j από 1 μέχρι 24
          Διάβασε Θ7[ί, j]
     τέλος_επανάληψης
τέλος_επανάληψης
Για i από 1 μέχρι 7
     H[i]- 0
τέλος_επανάληψης
Για j από 1 μέχρι 24
     Ω[j]<- 0
τέλος_επανάληψης
Για i από 1 μέχρι 7
     Για j από 1 μέχρι 24
          H[i]- H[i] + Θ7[i, j ]
          Ω^]- Ω^] + Θ7 [i, j]
     τέλος_επανάληψης
τέλος_επανάληψης
Για i από 1 μέχρι 7
     H[i]- H [i] / 24
     Γράψε 'Ημέρα ', i, ', μέση θερμοκρασία: ', H[i]
τέλος_επανάληψης
Για j από 1 μέχρι 24
     Ω[j]- Ω[j] /7
     Γράψε 'Ώρα ', j-1, ':00, μέση θερμοκρασία: ', Ω[j]
τέλος_επανάληψης
Τέλος Θερμοκρασίες
```

Παράδειγμα 5-2. Πρόβλημα «Θερμοκρασίες»

Ερωτήσεις - θέματα για συζήτηση

Α. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Ο πίνακας είναι μια δομή δεδομένων με συγκεκριμένο μέγεθος.
2. Δύο στοιχεία ενός πίνακα μπορούν να έχουν τον ίδιο δείκτη.
3. Οι δείκτες των στοιχείων ενός πίνακα μπορούν να είναι πραγματικοί αριθμοί.
4. Η διάσταση ενός πίνακα καθορίζει το πλήθος των δεικτών που χρησιμοποιούνται για κάθε στοιχείο του.
5. Όλα τα στοιχεία του πίνακα πρέπει να είναι διαφορετικά μεταξύ τους.
6. Τα στοιχεία ενός ταξινομημένου πίνακα ακολουθούν μια συγκεκριμένη διάταξη.
7. Η σειριακή αναζήτηση είναι ο ταχύτερος αλγόριθμος αναζήτησης.
8. Η δυαδική αναζήτηση μπορεί να εφαρμοστεί σε οποιοδήποτε πίνακα.
9. Για να διαβάσουμε πίνακες δύο ή περισσοτέρων διαστάσεων χρησιμοποιούμε συνήθως εμφωλευμένες επαναλήψεις.
10. Ο συμβολισμός Π[j, i] δηλώνει το στοιχείο του δισδιάστατου πίνακα Π που βρίσκεται στη γραμμή j και στη στήλη i.

Β. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με ένα στοιχείο της στήλης Β (Η στήλη Β περιέχει παραπάνω στοιχεία):

ΣΤΗΛΗ Α	**ΣΤΗΛΗ Β**
1. Διάβασε Π[5]	Α. Άθροισμα των στοιχείων ενός πίνακα.
2. Κ←0 Για i από 1 μέχρι Ν Κ←Κ+Α[i] Τέλος_επανάληψης	Β. Εύρεση του μέσου όρου ενός πίνακα
3. Διάβασε Π[3,7]	Γ. Διάβασμα ενός στοιχείου μονοδιάστατου πίνακα.
4. Αν Α[k]<Α[k-1] τότε	Δ. Διάβασμα ενός στοιχείου δισδιάστατου πίνακα.
5. Χ←0 Για i από 1 μέχρι Ν Αν Α[i]=0 τοτε Χ←Χ+1 Τέλος επανάληψης	Ε. Σύγκριση του πρώτου και του τελευταίου στοιχείου ενός πίνακα.
	ΣΤ. Εύρεση των μηδενικών στοιχείων ενός πίνακα.
	Ζ. Σύγκριση δύο γειτονικών στοιχείων ενός πίνακα.
	Η. Εύρεση του μέγιστου στοιχείου ενός πίνακα.

Γ. Εκτελέστε τις εντολές του παρακάτω αλγορίθμου και στη συνέχεια συμπληρώστε τα κελιά του πίνακα Π που ακολουθεί:

```
Π[1]←4
Π[2]←2*Π[1]
Π[3]←Π[1]+Π[2]
Π[4]←Π[1]*Π[2]
Π[5]←Π[Π[1]]+1
```

Πίνακας Π:

Θέση	1	2	3	4	5
Περιεχόμενο					

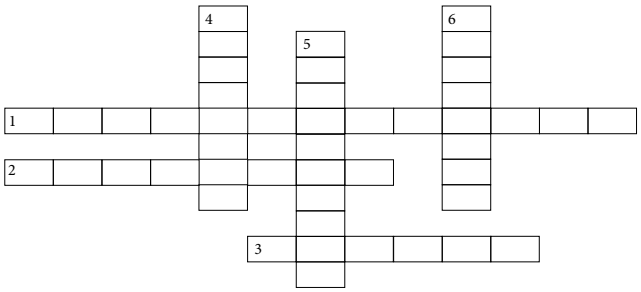
Δ. Λύστε το σταυρόλεξο που ακολουθεί:

ΟΡΙΖΟΝΤΙΑ:

1. Πίνακας με δύο διαστάσεις.
2. Δηλώνει τη θέση κάθε στοιχείου.
3. Στο στοιχείο Π[2,5] καθορίζεται από τον αριθμό 5.

ΚΑΘΕΤΑ:

4. Είδος αναζήτησης.
5. Βασική λειτουργία σε πίνακες.
6. Είδος δομής δεδομένων με συγκεκριμένο μέγεθος.



Ε. Ερωτήσεις

1. Ανατρέξτε στο πρόβλημα με τις θερμοκρασίες επτά ημερών - εικοσιτεσσάρων ωρών. Ο πίνακας Θ7 αποτελείται από 7 γραμμές και 24 στήλες. Θα μπορούσαν τα συγκεκριμένα δεδομένα να αποθηκευτούν σε πίνακα με 24 γραμμές και 7 στήλες; Περιγράψτε τις συμβάσεις που θα καθορίζατε για τον πίνακα αυτόν.
2. Στο πρόβλημα των θερμοκρασιών επτά ημερών, τι ταξινομήσεις θα είχε νόημα να κάνουμε στον πίνακα με τις θερμοκρασίες; Τι θα γινόταν αν ταξινομούσαμε τον δισδιάστατο πίνακα Θ7 γραμμή προς γραμμή;

Ασκήσεις - Προβλήματα

1. Αποθηκεύστε σε έναν πίνακα 10 θέσεων την προπαίδεια του αριθμού 5 (5, 10, 15,.)
2. Διαβάστε τα στοιχεία ενός πίνακα 100 θέσεων και στη συνέχεια εμφανίστε τα ξεκινώντας από το τελευταίο που διαβάσατε έως το πρώτο.
3. Ο παρακάτω κώδικας υπολογίζει πόσες από τις 100 φωτογραφίες που έχουν αναρτηθεί σε έναν λογαριασμό κοινωνικής δικτύωσης έχουν πάνω από 50 «Μου αρέσει». Μετατρέψτε τον παρακάτω κώδικα σε ισοδύναμο διάγραμμα ροής:

```
Αλγόριθμος Φωτογραφίες
Κ←0
Για i από 1 μέχρι 400
    Αν Β[i]>50 τότε
         Κ←Κ+1
    Τέλος_Αν
Τέλος_επανάληψης
Εμφάνισε Κ
Τέλος Φωτογραφίες
```

4. Το παρακάτω τμήμα κώδικα προσπαθεί λανθασμένα να υπολογίσει το μέγιστο στοιχείο ενός δισδιάστατου πίνακα 50Χ50:

```
1. ΜΑΧ←Π[1,1]
2. Για i από 2 μέχρι 50
3.     Για i από 1 μέχρι 50
4.        Αν Π[i,j]>100 τότε
5.              ΜΜΑΧ←Π[i,j]
```

```
6.            Τέλος_Αν
7.      Τέλος_επανάληψης
8.     Εμφάνισε MAX
```

Εντοπίστε τα λάθη στον κώδικα γράφοντας τον αριθμό της γραμμής και περιγράφοντας το λάθος που εντοπίσατε. Στη συνέχεια ξαναγράψτε τον κώδικα ώστε να εμφανίζει σωστά το μέγιστο στοιχείο.

5. Στον τελικό των 100 μέτρων στους Ολυμπιακούς αγώνες λαμβάνουν μέρος 8 αθλητές σε 8 διαδρόμους και οι χρόνοι τους καταγράφονται σε έναν πίνακα Π 8 θέσεων. Να κατασκευάσετε αλγόριθμο όπου:
   Α. Θα διαβάζετε το χρόνο που πέτυχε κάθε αθλητής και θα τον αποθηκεύετε στον πίνακα.
   Β. Θα βρίσκετε και θα εμφανίζετε το διάδρομο του αθλητή που κέρδισε τον αγώνα.
   Γ. Θα εμφανίζετε τον αριθμό των αθλητών που πέτυχαν επίδοση κάτω από 10 δευτερόλεπτα.
6. Ο Αλέξανδρος πήγε στο super market και αγόρασε 30 διαφορετικά προϊόντα. Θέλει να επαληθεύσει το σύνολο του λογαριασμού του και γι' αυτό το λόγο κατασκεύασε έναν πίνακα όπου πέρασε τις τιμές των προϊόντων που αγόρασε. Βοήθησέ τον κατασκευάζοντας έναν αλγόριθμο όπου:
   Α. Θα αθροίζει τις τιμές του πίνακα και θα εμφανίζει τον συνολικό λογαριασμό.
   Β. Θα βρίσκει εάν υπάρχει στον πίνακα προϊόν με τιμή ίση με 8,50€ και θα εμφανίζει κατάλληλο μήνυμα («Ναι υπάρχει/ Όχι δεν υπάρχει»).
   Γ. Θα ταξινομεί τις τιμές των προϊόντων από τη μικρότερη προς τη μεγαλύτερη.

7. Να διαβαστούν τα στοιχεία μονοδιάστατου πίνακα: α) από το τελευταίο προς το πρώτο, β) πρώτα στις μονές και κατόπιν στις ζυγές θέσεις, γ) από τα άκρα προς το μέσο του πίνακα (πρώτο, τελευταίο, δεύτερο, προτελευταίο, ...).
8. Πρώτος είναι ένας αριθμός που διαιρείται μόνο με τον εαυτό του και τη μονάδα. Να γραφτεί αλγόριθμος που υπολογίζει και τοποθετεί σε πίνακα τους 20 πρώτους θετικούς αριθμούς.
9. Οι δόσεις για την αγορά ενός υπολογιστή υπολογίζονται ως εξής: ο πελάτης επιλέγει το ποσό της προκαταβολής και το πλήθος των μηνιαίων δόσεων, ενώ το ποσό των δόσεων προσαυξάνεται με ένα ετήσιο επιτόκιο. Να γραφτεί αλγόριθμος που υπολογίζει και καταχωρεί σε πίνακα το ποσό κάθε δόσης για την αγορά του υπολογιστή.
10. Αλλάξτε τον αλγόριθμο της σειριακής αναζήτησης ώστε να είναι πιο αποδοτικός σε ταξινομημένους πίνακες.
11. Σε ένα κατάστημα πώλησης ηλεκτρονικών αγαθών, πωλούνται 10 διαφορετικά μοντέλα smartphones. Να γραφτεί αλγόριθμος που:
    α) να διαβάζει και να καταχωρεί σε πίνακα τα μοντέλα των τηλεφώνων και την τιμή τους,

β) να τα εμφανίζει ταξινομημένα με αύξουσα τιμή,
γ) να ζητάει μια τιμή από το χρήστη και να εμφανίζει όλα τα smartphones με τιμή μικρότερη ή ίση από αυτήν.

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να επιλέγετε τον κατάλληλο τύπο για κάθε δεδομένο που θέλετε να χρησιμοποιήσετε.
- να εντοπίζετε και να περιγράφετε δομές δεδομένων για συγκεκριμένα προβλήματα.
- να αναγνωρίζετε σε ποια προβλήματα είναι κατάλληλη και σε ποια είναι απαραίτητη η χρήση πινάκων.
- να αποφασίζετε τη διάσταση των πινάκων που θα χρησιμοποιήσετε.
- να επιλέγετε τον κατάλληλο αλγόριθμο αναζήτησης για συγκεκριμένο πρόβλημα.
- να εφαρμόζετε τις λειτουργίες πινάκων μίας ή δύο διαστάσεων σε συγκεκριμένα προβλήματα.
- να σχεδιάζετε και να υλοποιείτε λειτουργίες σε πίνακες.

Βιβλιογραφία

1. H. Abelson, G. J. Sussman, *Structure and Interpretation of Computer Programs*, 2nd Edition, 1996.
2. P. F. Drucker, *Landmarks of Tomorrow: A Report on the New "Post-Modern "* World, 1959.
3. R. Elmasri, S. B. Navathe, *Fundamentals of Database Systems,* Sixth Edition, Addison-Wesley, 2010.
4. A. Hodges, Άλαν Τιούρινγκ: *Το αίνιγμα*, εκδόσεις Τραυλός, 2004.
5. H. Lewis, X. Παπαδημητρίου. *Στοιχεία Θεωρίας Υπολογισμού*, εκδόσεις Κριτική, 2005
6. M. Sipser, *Εισαγωγή στη Θεωρία Υπολογισμού*, Πανεπιστημιακές Εκδόσεις Κρήτης, 2007
7. E. Horowitz, S. Sahni, *Fundamentals of Data Structures in Pascal,* W. H. Freeman; Fourth Edition, 1993.
8. B. Kernighan και D. Ritchie, *Η Γλώσσα Προγραμματισμού C*, Εκδόσεις Κλειδάριθμος, Αθήνα 1990.
9. D. Knuth, *The art of Computer Programming, Volume 1: Fundamental Algorithms*. Third Edition, Reading, Massachusetts: Addison-Wesley, 1997.
10. G. Polya, *Πώς να το λύσω*, Εκδόσεις Καρδαμίτσα, Αθήνα 1991.
11. P. Rechenberg, *Εισαγωγή στην Πληροφορική - Μία ολοκληρωμένη παρουσίαση*, Εκδόσεις Κλειδάριθμος, 1992.
12. A. Βακάλη, κ.ά., *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*, Ο.Ε.Δ.Β, 2008.
13. N. Αντωνάκος κ.ά.,*Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*, Ο.Ε.Δ.Β, 1999.
14. N. Ιωαννίδης, K. Μαρινάκης, Σ. Μπακογιάννης, *Δομημένη σχεδίαση Προγράμματος Αλγοριθμική*, Εκδόσεις N. Τεχνολογιών, 1991.
15. I. Μανωλόπουλος, *Δομές Δεδομένων - Μία προσέγγιση με Pascal*, Εκδόσεις Art of Text, 1997.

Πηγές on-line

1. M. Armbrust, e.a. (2009). *Above the Clouds: A Berkeley View of Cloud Computing*, TR No. UCB/EECS-2009-28, [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html.
2. D. Evans (August 19, 2011). *Introduction to Computing: Explorations in Language, Logic, and Machines,* [Online]. Available: http://www.computingbook.org/FullText.pdf.
3. R. Sedgewick, K. Wayne (2003). *An Introduction to Computer Science*, [Online]. Available: ftp://ftp.cs.princeton.edu/pub/people/ rs/book0/pdf/IntroCS.book.pdf.
4. R. Sedgewick, K. Wayne (2013, Oct. 13). *Introduction to Programming in Java*, Booksite [Online]. Available: http://introcs.cs.princeton.edu/java/home/.
5. R. Sedgewick, K. Wayne (2014, Jun 15). *Algorithms*, (4th .Edition), Booksite [Online]. Available: http://algs4.cs.princeton.edu/home/.
6. R. Sedgewick, P. Flajolet. (2013). *An Introduction to the Analysis of Algorithms*, Booksite [Online]. Available: http://aofa.cs.princeton.edu/ home/.
7. R. Toal (2004). *Programming Paradigms*, [Online]. Available: http://cs.lmu.edu/~ray/notes/paradigms.
8. P. Van Roy (2009). *Programming Paradigms for Dummies*: *What Every Programmer Should Know*, [Online]. Available: http://www.info.ucl.ac.be/~pvr/VanRoyChapter.pdf.
9. UNESCO. *UNESCO World Report: Towards Knowledge Societies, 2005*, http://unesdoc.unesco.org/images/0014/001418/141843e.pdf.
10. Wikipedia. *Programming paradigm* (2014, Jun 19), [Online]. Available: http://en.wikipedia.org/wiki/Programming_paradigm.
11. Wikipedia. *Comparison of programming languages* (2014, July 06), [Online]. Available: http:// en.wikipedia.org/wiki/ Comparison_of_programming_languages.
12. Κασσιός Γ., *Εισαγωγή στο Συναρτησιακό Προγραμματισμό*, http://cgi.di.uoa.gr/~kassios/courses/fp/notes/0.pdf.

Λεξικό Βασικών Όρων

Αλγόριθμος. Η ακριβής περιγραφή μιας σειράς βημάτων που απαιτούνται για την επίλυση ενός προβλήματος.

Αναδρομικός αλγόριθμος. Ο αλγόριθμος που καλεί άμεσα ή έμμεσα τον εαυτό του μία ή περισσότερες φορές, επιλύοντας κάθε φορά ένα πρόβλημα της ίδιας φύσης με το αρχικό, αλλά μικρότερου μεγέθους.

Αναζήτηση. Λειτουργία σε πίνακα, που συνίσταται στην εξέταση των στοιχείων του για την εύρεση ενός ζητούμενου.

Αναλυτική μεθοδολογία. Μέθοδος σχεδιασμού με διαδοχικά βήματα.

Αντικειμενοστραφής προγραμματισμός. Μεθοδολογία ανάπτυξης προγραμμάτων προστακτικού τύπου με κεντρική ιδέα την κλάση αντικειμένων.

Βρόχος. Μια επαναληπτική διαδικασία.

Γλώσσες προγραμματισμού. Αναπαράσταση αλγορίθμου με κείμενο σε γλώσσα περιορισμένου συνόλου κωδικοποιημένων εντολών, που είναι κατανοητή από τον υπολογιστή.

Δεδομένα (data). Τα στοιχεία που αναπαρίστανται σε αλγορίθμους και προγράμματα και υφίστανται επεξεργασία.

Δεδομένο προβλήματος .Ένα γνωστό ή αποδεκτό στοιχείο το οποίο χρησιμοποιείται ως βάση ή προϋπόθεση στην επίλυση προβλημάτων.

Διάγραμμα ροής. Αναπαράσταση αλγορίθμου με χρήση γεωμετρικών σχημάτων.

Διερμηνευτής. Μεταφραστικό πρόγραμμα που αναλύει, μεταφράζει και εκτελεί το πηγαίο πρόγραμμα εντολή προς εντολή.

Δομή δεδομένων. Τρόπος οργάνωσης, συσχέτισης και αποθήκευσης των δεδομένων.

Δομημένος προγραμματισμός. Στυλ προγραμματισμού το οποίο βασίζεται στην έννοια της κλήσης διαδικασιών.

Δυναμικές δομές δεδομένων. Δομές δεδομένων που δεν έχουν σταθερό μέγεθος και αυξομειώνονται με τη μέθοδο της δυναμικής παραχώρησης μνήμης.

Εκσφαλματωτής. Ειδικό πρόγραμμα που επιτρέπει την άμεση εκτέλεση και παρακολούθηση του πηγαίου προγράμματος με σκοπό τον εντοπισμό των λογικών λαθών.

Επιστήμη Υπολογιστών. Η επιστήμη που ασχολείται με τους υπολογιστές και τους υπολογισμούς.

Λογικά λάθη. Λάθη στη λογική του προγράμματος που οδηγούν σε αδυναμία εκτέλεσης ή σε λανθασμένα αποτελέσματα. Συμβαίνουν και εντοπίζονται κατά την εκτέλεση του προγράμματος.

Λογισμικό. Τα προγράμματα που περιέχουν τις εντολές προς τις συσκευές του υπολογιστή.

Μεταβλητή. Μια ποσότητα που αναπαριστά ένα στοιχείο που έχει νόημα στον πραγματικό κόσμο και η τιμή της μπορεί να αλλάζει κατά τη διάρκεια εκτέλεσης του αλγορίθμου.

Μεταγλωττιστής. Πρόγραμμα ποιυ διαβάζει και ελέγχει όλο το πηγαίο πρόγραμμα, και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής..

Μεταφραστής. Πρόγραμμα που αναλαμβάνει να μετατρέψει το πηγαίο πρόγραμμα σε γλώσσα μηχανής.

Παράμετρος. Μεταβλητή που επιτρέπει το πέρασμα τιμών από ένα τμήμα προγράμματος σε ένα άλλο.

Πίνακας. Διάταξη στοιχείων του ίδιου τύπου με συγκεκριμένο μέγεθος, στην οποία όλα τα στοιχεία αναφέρονται με το ίδιο όνομα και έναν ή περισσότερους δείκτες.

Πληροφορική. Η επιστήμη που ασχολείται με την αναπαράσταση, αποθήκευση και επεξεργασία της πληροφορίας.

Πολυπλοκότητα αλγορίθμων. Κλάδος της Θεωρίας Υπολογισμού που μελετά την απόδοση των αλγορίθμων.

Πρόβλημα. Μια κατάσταση που απαιτεί λύση αλλά η λύση της δεν είναι γνωστή ούτε προφανής

Στατικές δομές δεδομένων. Δομές δεδομένων που έχουν σταθερό μέγεθος και καθορίζεται κατά την δημιουργία (συγγραφή) του προγράμματος.

Συνθετική μεθοδολογία. Μέθοδος σχεδίασης βασισμένη στη «σταδιακή εξέλιξη». Η γενική της αρχή είναι ότι η λύση ενός προβλήματος επιτυγχάνεται από τη σύνθεση των επί μέρους στοιχείων και δεδομένων.

Συντάκτης. Ειδικό πρόγραμμα που βοηθάει στη συγγραφή ενός προγράμματος.

Συντακτικά λάθη. Λάθη που παραβιάζουν τους κανόνες ορθογραφίας ή σύνταξης της γλώσσας με την οποία εκφράζεται ένας αλγόριθμος ή ένα πρόγραμμα. Συμβαίνουν κατά τη σύνταξη του προγράμματος και εντοπίζονται κατά τη μετάφραση του.

Ταξινόμηση. Λειτουργία σε πίνακα, που συνίσταται στην αλλαγή της θέσης των στοιχείων του ώστε να ακολουθούν μια συγκεκριμένη διάταξη, π.χ. αύξουσα ή φθίνουσα.

Τελεστής. Το σύμβολο που χρησιμοποιείται για την εκτέλεση μιας πράξης.

Τύπος δεδομένων. Το είδος δεδομένων που μπορεί να παραστήσει μια γλώσσα προγραμματισμού.

Υλικό. Τα στοιχεία ενός υπολογιστή που έχουν υλική υπόσταση.

Υπολογισιμότητα. Κλάδος της επιστήμης των υπολογιστών που αναζητά τι μπορεί να υπολογισθεί και τι όχι

Υπολογιστικά προβλήματα. Προβλήματα που ζητάμε να βρούμε την απάντηση που ικανοποιεί τα δεδομένα του.

Υποπρόγραμμα. Τμήμα πηγαίου κώδικα που εκτελεί συγκεκριμένη λειτουργία, μπορεί να διαχωριστεί από το υπόλοιπο πρόγραμμα και να χρησιμοποιηθεί πολλές φορές σε διαφορετικά σημεία ενός ή περισσότερων προγραμμάτων.

Ευρετήριο εικόνων

Εικόνα 1-1. Ταξινόμηση της Επιστήμης Υπολογιστών από την ACM (Association for Computing Machinery) 12
Εικόνα 2-1. Κατηγορίες προβλημάτων με κριτήριο την επιλυσιμότητα 19
Εικόνα 2-2. Στάδια επίλυσης προβλήματος 20
Εικόνα 2-4. Παρουσίαση του προβλήματος σε φυσική και αλγεβρική γλώσσα 21
Εικόνα 2-3. Δεδομένα χωρίς πληρότητα 21
Εικόνα 2-5. Προετοιμασία ταξιδιού 22
Εικόνα 2-6. Συνθετική μεθοδολογία 23
Εικόνα 3-1. Σειριακή αναζήτηση του ονόματος «Ωραίου» 35
Εικόνα 3-2. Δυαδική Αναζήτηση του ονόματος «Παπαδάκης» 36
Εικόνα 3-3. Χρόνοι εκτέλεσης αλγορίθμων με χρόνο εκτέλεσης εντολής 1 ns 37
Εικόνα 3-5. Παράλληλη εκτέλεση αλγορίθμου 38
Εικόνα 3-4. Σειριακή και παράλληλη εκτέλεση αλγορίθμου 38
Εικόνα 3-6. Αναδρομικός αλγόριθμος 39
Εικόνα 3-7. Σύμβολα διαγραμμάτων ροής 42
Εικόνα 3-8. Τρεις αναπαραστάσεις του ίδιου αλγορίθμου 43
Εικόνα 4-7. Διάγραμμα ροής της επανάληψης μέχρις_ότου 67
Εικόνα 4-8. Διάγραμμα ροής της επανάληψης Για 68
Εικόνα 4-9. Σύγκριση των δομών επανάληψης 69
Εικόνα 5-1. Αντιστοίχηση του πραγματικού κόσμου με έννοιες της Πληροφορικής 83
Εικόνα 5-2. Απλοί τύποι δεδομένων 84
Εικόνα 5-3. Παραδείγματα δεδομένων του μαθητολογίου και των τύπων τους 85
Εικόνα 5-6. Πίνακας θερμοκρασιών 24τετραώρου για 7 ημέρες 91

Ευρετήριο αλγορίθμων

Αλγόριθμος 4-2. Ανταλλακτήριο συναλλάγματος 54
Αλγόριθμος 4-1. Μεταφόρτωση εφαρμογής 54
Αλγόριθμος 4-3 (β). Απόλυτη τιμή αριθμού με απλή επιλογή 58
Αλγόριθμος 4-3 (α). Απόλυτη τιμή αριθμού με σύνθετη επιλογή 58
Αλγόριθμος 4-4. Μέγιστος μεταξύ δύο αριθμών 58
(α) με σύνθετη επιλογή, (β) με απλή επιλογή 58
Αλγόριθμος 4-5. Χαρακτηρισμός ατόμου με βάση το ΔΜΣ με πολλαπλή επιλογή 59
Αλγόριθμος 4-6. Υπολογισμός κόστους αποστολής με εμφωλευμένη επιλογή 60
Αλγόριθμος 4-7. Υπολογισμός κόστους αποστολής με πολλαπλή επιλογή 61
Αλγόριθμος 4-8. Μέσος όρος βαθμολογίας 67
Αλγόριθμος 4-9. Συγκέντρωση ποσού για αγορά υπολογιστή 68
Αλγόριθμος 4-10. Μέσος όρος και μέγιστος βαθμός 68
Αλγόριθμος 4-11. Εκτύπωση της προπαίδειας του αριθμού 1 69
Αλγόριθμος 4-12. Εκτύπωση της προπαίδειας των αριθμών 1 ως 10 70
Αλγόριθμος 4-13. Προσεγγιστικός υπολογισμός της τετραγωνικής ρίζας αριθμού 70
Αλγόριθμος 4-14. Υπολογισμός του παραγοντικού, ... 71
(α) με αναδρομή, (β) χωρίς αναδρομή 71
Αλγόριθμος 4-15. Τεκμηρίωση με σχόλια στον αλγόριθμο υπολογισμού του μέγιστου 78
Αλγόριθμος 5-2. Άθροισμα και μέσος όρος στοιχείων πίνακα μίας διάστασης 92
Αλγόριθμος 5-1. Διάβασμα στοιχείων πίνακα μίας διάστασης 92
Αλγόριθμος 5-3. Υπολογισμός ελάχιστου 92
Αλγόριθμος 5-4. Σειριακή αναζήτηση 93
Αλγόριθμος 5-5. Δυαδική αναζήτηση 94
Αλγόριθμος 5-6. Ταξινόμηση με εισαγωγή 95

Βάσει του ν. 3966/2011 τα διδακτικά βιβλία του Δημοτικού, του Γυμνασίου, του Λυκείου, των ΕΠΑ.Λ. και των ΕΠΑ.Σ. τυπώνονται από το ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ και διανέμονται δωρεάν στα Δημόσια Σχολεία. Τα βιβλία μπορεί να διατίθενται προς πώληση, όταν φέρουν στη δεξιά κάτω γωνία του εμπροσθόφυλλου ένδειξη «ΔΙΑΤΙΘΕΤΑΙ ΜΕ ΤΙΜΗ ΠΩΛΗΣΗΣ». Κάθε αντίτυπο που διατίθεται προς πώληση και δεν φέρει την παραπάνω ένδειξη θεωρείται κλεψίτυπο και ο παραβάτης διώκεται σύμφωνα με τις διατάξεις του άρθρου 7 του νόμου 1129 της 15/21 Μαρτίου 1946 (ΦΕΚ 1946,108, Α').

Απαγορεύεται η αναπαραγωγή οποιουδήποτε τμήματος αυτού του βιβλίου, που καλύπτεται από δικαιώματα (copyright), ή η χρήση του σε οποιαδήποτε μορφή, χωρίς τη γραπτή άδεια του Υπουργείου Παιδείας, Θρησκευμάτων και Αθλητισμού / ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ.

Κωδικός Βιβλίου: 0-24-0565
ISBN 978-960-06-5205-5

ΙΤΥΕ ΔΙΟΦΑΝΤΟΣ
ινστιτούτο τεχνολογίας υπολογιστών & εκδόσεων

(01) 000000 0 24 0565 0