

Φίλη μαθήτριά, Φίλε μαθητή

Το βιβλίο αυτό αποτελεί ένα χρήσιμο βοήθημα για το μάθημα της Πληροφορικής της Γ' Γυμνασίου. Στόχος του είναι να σε βοηθήσει να κατανοήσεις τις βασικές έννοιες της Πληροφορικής και να εμπλουτίσεις τις γνώσεις σου σε σύγχρονα θέματα που την αφορούν.

Συγκεκριμένα, το βιβλίο αυτό περιέχει:

- ✓ Βασικές ερωτήσεις θεωρίας.
- ✓ Απαντήσεις σε όλες τις ερωτήσεις, τις δραστηριότητες και τα θέματα συζήτησης του σχολικού βιβλίου.
- ✓ Απαντήσεις στις ασκήσεις αυτοαξιολόγησης του σχολικού βιβλίου.
- ✓ Ασκήσεις εμπέδωσης στο τέλος κάθε κεφαλαίου, για να ελέγξεις αν κατανόησες τα βασικά σημεία του μαθήματος.
- ✓ Πρόσθετες δραστηριότητες και επιπλέον θέματα για συζήτηση που θα σε βοηθήσουν να κατανοήσεις καλύτερα την ύλη.
- ✓ Πρόσθετες πληροφορίες.

Χρήστος Νικολαΐδης

Περιεχόμενα

ΕΝΟΤΗΤΑ 1: ΓΝΩΡΙΖΩ ΤΟΝ ΥΠΟΛΟΓΙΣΤΗ

ΩΣ ΕΝΙΑΙΟ ΣΥΣΤΗΜΑ – ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Κεφάλαιο 1: Εισαγωγή στην Έννοια του Αλγορίθμου και στον Προγραμματισμό	σελ. 11
Κεφάλαιο 2: Ο Προγραμματισμός στην Πράξη	σελ. 48

ΕΝΟΤΗΤΑ 2: ΧΡΗΣΗ ΕΡΓΑΛΕΙΩΝ ΕΚΦΡΑΣΗΣ, ΕΠΙΚΟΙΝΩΝΙΑΣ,

ΑΝΑΚΑΛΥΨΗΣ ΚΑΙ ΔΗΜΙΟΥΡΓΙΑΣ:

ΜΕΓΑΛΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ σελ. 153

ΕΝΟΤΗΤΑ 3: Ο ΥΠΟΛΟΓΙΣΤΗΣ ΣΤΗΝ ΚΟΙΝΩΝΙΑ

ΚΑΙ ΣΤΟΝ ΠΟΛΙΤΙΣΜΟ σελ. 197

Κατάλογος εντολών της Logo στα αγγλικά και στα ελληνικά	σελ. 220
--	-----------------

Απαντήσεις	σελ. 227
-----------------------------	-----------------



Εισαγωγή στην Έννοια του Αλγορίθμου και στον Προγραμματισμό

ΤΙ ΠΡΕΠΕΙ ΝΑ ΓΝΩΡΙΖΕΤΕ

Υπάρχουν πολλές κατηγορίες προβλημάτων. Τρεις, όμως, είναι οι βασικές: α. τα προβλήματα που λύνονται, β. τα προβλήματα που δεν μπορούμε να τα λύσουμε με τις μέχρι τώρα γνώσεις μας και γ. τα προβλήματα που έχει «αποδειχθεί» ότι δεν μπορούμε να τα λύσουμε.

Ως πρόβλημα θεωρούμε κάθε ζήτημα που τίθεται προς επίλυση. Η πρώτη μας ενέργεια για να λύσουμε πιο εύκολα ένα πρόβλημα, είναι η **καταγραφή των δεδομένων**. **Δεδομένα προβλήματος** είναι τα στοιχεία που μας είναι γνωστά και μπορούν να μας βοηθήσουν στη λύση του. Σε κάθε πρόβλημα ψάχνουμε να βρούμε την απάντηση σε μια ερώτηση. Αυτό που ψάχνουμε είναι το **ζητούμενο** του προβλήματος.

Η διαδικασία μέσω της οποίας βρίσκουμε το ζητούμενο και επιτυγχάνουμε τον επιθυμητό στόχο ονομάζεται **επίλυση προβλήματος**. Υπάρ-

χουν προβλήματα, των οποίων τη λύση μπορούμε να περιγράψουμε με ακρίβεια, και προβλήματα που δεν έχουν ακριβή λύση. Είναι απαραίτητο, επίσης, να ελέγχουμε αν τα δεδομένα του προβλήματος που έχουμε είναι επαρκή, ώστε να μπορούμε να σχεδιάσουμε την επίλυση του.

Για να επιλύσουμε ένα πρόβλημα πρέπει α. να το **κατανοήσουμε** (διακρίνοντας τα δεδομένα που έχουμε στη διάθεση μας και τα ζητούμενά του) και β. να προσδιορίσουμε το **περιβάλλον** μέσα στο οποίο εντάσσεται το πρόβλημα (χώρος του προβλήματος). Συνήθως, για να μπορέσουμε να επιλύσουμε ένα σύνθετο πρόβλημα, απαιτείται η ανάλυσή του σε απλούστερα προβλήματα. Η **διατύπωση σωστών οδηγιών** για την επίλυση ενός προβλήματος είναι εξίσου σημαντική. Οι οδηγίες που δίνουμε με λογική σειρά, ώστε να εκτελέσουμε μια εργασία ή να επιλύσουμε ένα πρόβλημα, συνθέτουν έναν **Αλγόριθμο**. **Αλγόριθμο** ονομάζουμε τη σαφή και ακριβή περιγραφή μιας σειράς ξεχωριστών οδηγιών-βημάτων, με σκοπό την επίλυση ενός προβλήματος.

Τα βήματα που αποτελούν έναν αλγόριθμο ονομάζονται **οδηγίες ή εντολές**. Αν ακολουθήσουμε τις οδηγίες ενός αλγορίθμου, στο τέλος πρέπει να προκύπτει ένα αποτέλεσμα, ένα έργο. Οι εντολές ενός αλγορίθμου πρέπει α. να εξασφαλίζουν ότι ο αλγόριθμος κάποια στιγμή θα τελειώσει ολοκληρώνοντας το σκοπό του και β. να έχουν **ακρίβεια** και **σαφήνεια**.

Πρόγραμμα ονομάζουμε την αναπαράσταση ενός αλγορίθμου γραμμένη σε μια γλώσσα που «καταλαβαίνει» ο υπολογιστής. Ένα πρόγραμμα αποτελείται από μια σειρά εντολών προς τον υπολογιστή με σκοπό την εκτέλεση κάποιας λειτουργίας ή υπολογισμού.

Η εργασία σύνταξης των προγραμμάτων ονομάζεται **προγραμματισμός**, ενώ τα άτομα που γράφουν και συντάσσουν ένα πρόγραμμα ονομάζονται **προγραμματιστές**.

Για την επικοινωνία μεταξύ προγραμματιστή και υπολογιστή χρησιμοποιείται μια κοινή τεχνητή γλώσσα που να την «καταλαβαίνουν» και οι δύο. Υπάρχουν πολλών ειδών «τεχνητές γλώσσες» αυτού του είδους που ονομάζονται **γλώσσες προγραμματισμού**. Οι γλώσσες προγραμματισμού έχουν το δικό τους αλφάβητο, λεξιλόγιο και συντακτικό. Η γλώσσα προγραμματισμού μέσω της οποίας οι εντολές δίνονται (αναπαρίστανται) στο δυαδικό σύστημα (0 και 1) ονομάζεται **γλώσσα μηχανής**.

Μερικές γνωστές γλώσσες προγραμματισμού είναι η Visual Basic (έχει αντικατασταθεί από τη .NetBasic εδώ και πολλά χρόνια), η Logo, η Pascal (έχει αντικατασταθεί από τη Delphi πριν από μερικές δεκαετίες), η C++, η Java, η PHP κ.ά.

Κάθε γλώσσα προγραμματισμού έχει ως βασικά χαρακτηριστικά το **αλφάβητο**, το **λεξιλόγιο** και το **συντακτικό**.

- Το **αλφάβητο** είναι το σύνολο των χαρακτήρων που χρησιμοποιούνται από τη γλώσσα προγραμματισμού.
- Το **λεξιλόγιο** είναι το σύνολο των εντολών-λέξεων που χρησιμοποιεί η γλώσσα.
- Το **συντακτικό** είναι το σύνολο των κανόνων που πρέπει να ακολουθούμε (κάθε εντολή συντάσσεται με συγκεκριμένο τρόπο) στη διάρκεια του προγραμματισμού.

Το περιβάλλον δημιουργίας προγραμμάτων σε μία γλώσσα προγραμματισμού ονομάζεται **περιβάλλον προγραμματισμού**. Τα βασικά εργαλεία ενός περιβάλλοντος προγραμματισμού είναι ο **κειμενογράφος** και ο **μεταφραστής**. Αν δεν έχουμε συντάξει/γράψει σωστά κάποια οδηγία (εντολή), τότε λέμε ότι έχουμε κάνει **συντακτικό λάθος**.

Τα προγράμματα που μετατρέπουν τις οδηγίες μας σε γλώσσα που «καταλαβαίνει» ο επεξεργαστής χωρίζονται σε δύο κατηγορίες, στους μεταγλωττιστές και στους διερμηνείς.

- Οι **μεταγλωττιστές (compilers)** δημιουργούν εκτελέσιμα αρχεία, δηλαδή φτιάχνουν όλη τη σειρά εντολών σε ένα αρχείο προγράμματος.
- Οι **διερμηνείς (interpreters)** ελέγχουν μία οδηγία κάθε φορά, την εκτελούν και μετά ελέγχουν την επόμενη οδηγία (π.χ. η γλώσσα προγραμματισμού Logo).

Αν δεν έχουμε κάνει συντακτικά λάθη, αλλά λάθη που προκύπτουν από τη σειρά ή λάθη από τα αποτελέσματα που δίνουν οι σωστά συντεταγμένες εντολές μας, τότε λέμε ότι έχουμε κάνει ένα **λογικό λάθος**. Σε αυτή την περίπτωση χρειάζεται να ελέγξουμε μία προς μία τις εντολές του αλγορίθμου μας, για να εντοπίσουμε ποιες εντολές δεν έχουν δοθεί με τη σωστή σειρά.

1.1 Τι ονομάζουμε πρόβλημα;

ΑΠΑΝΤΗΣΗ

Κάθε ζήτημα που τίθεται προς επίλυση ονομάζεται πρόβλημα. Σε απλή γλώσσα πρόβλημα ονομάζεται οτιδήποτε παρουσιάζεται μπροστά μας και χρειάζεται να βρούμε τη λύση του [πρόβλημα (ετυμολογικά) = αυτό που προβάλλεται, αυτό που εμφανίζεται, προκύπτει].



1.2 Πόσα είδη προβλημάτων υπάρχουν;

ΑΠΑΝΤΗΣΗ

Υπάρχουν τριών ειδών προβλήματα:

- α. Τα προβλήματα που, είτε εύκολα είτε δύσκολα, έχουν λύση.
- β. Τα προβλήματα που, μάλλον, έχουν λύση αλλά με τις παρούσες γνώσεις μας και δυνατότητες δεν μπορούν να λυθούν. Τα προβλήματα αυτά ενδέχεται στο μέλλον να καταφέρουμε να τα λύσουμε.
- γ. Τα προβλήματα που όλες οι ενδείξεις φανερώνουν ότι δεν μπορούν να λυθούν.

ΕΠΙΠΛΕΟΝ ΔΡΑΣΤΗΡΙΟΤΗΤΑ

Ψάξτε στο Διαδίκτυο να βρείτε στοιχεία για την «εικασία του Γκόλντμπαχ».



ΓΝΩΡΙΖΕΙΣ ΟΤΙ...

Ο Κομφούκιος είπε: Για δύο πράγματα δεν πρέπει να στεναχωριόμαστε: α. για τα προβλήματα που λύνονται και β. για τα προβλήματα που δεν λύνονται. Αν τα προβλήματα λύνονται, τότε δεν υπάρχει λόγος να στεναχωριόμαστε. Από την άλλη, αν ούτως ή άλλως δεν λύνονται, τότε γιατί να στεναχωριόμαστε και να «σπάμε» το κεφάλι μας;



1.3 Από τι αποτελείται ένα πρόβλημα;

ΑΠΑΝΤΗΣΗ

Ένα πρόβλημα αποτελείται από τα **δεδομένα**, δηλαδή από τα γνωστά στοιχεία που συνθέτουν το πρόβλημα και θα μας βοηθήσουν να φτάσουμε στην επίλυσή του και το **ζητούμενο**, δηλαδή το αποτέλεσμα της επίλυσης.

1.4 Τι ονομάζουμε επίλυση προβλήματος;

ΑΠΑΝΤΗΣΗ

Η διαδικασία μέσω της οποίας βρίσκουμε το ζητούμενο και επιτυγχάνουμε τον επιθυμητό στόχο ονομάζεται **επίλυση προβλήματος**. Το πρόβλημα, ως έννοια, παρομοιάζεται με έναν κόμπο, κόμπο που καλούμαστε να τον λύσουμε. Έτσι έχει επικρατήσει να λέμε «λύση» ή «επίλυση» του προβλήματος.

1.5 Ποια πρέπει να είναι η πρώτη μας ενέργεια για να προχωρήσουμε στην επίλυση ενός προβλήματος;

ΑΠΑΝΤΗΣΗ

Το πρώτο πράγμα που πρέπει να κάνουμε είναι να κατανοήσουμε το πρόβλημα, δηλαδή να κατανοήσουμε ποια είναι τα δεδομένα και το ζητούμενο του προβλήματος.

1.6 Ποιοι παράγοντες πρέπει να προϋπάρχουν για την επίλυση ενός προβλήματος;

ΑΠΑΝΤΗΣΗ

Αρχικά πρέπει να υπάρχει **σωστή και σαφής διατύπωση** του προβλήματος, έτσι ώστε να κατανοήσουμε τα δεδομένα του και ποιο είναι το ζητούμενο. Έπειτα, πρέπει να προσδιορίσουμε ποιο είναι το **περιβάλλον** του προβλήματος, δηλαδή σε ποιο πλαίσιο εντάσσεται (είναι μαθηματικό πρόβλημα; είναι αίνιγμα; είναι θεωρητικό;). Τέλος πρέπει να διαπιστώσουμε, αν τα δεδομένα που έχουμε στη διάθεσή μας επαρκούν για την επίλυση του προβλήματος.

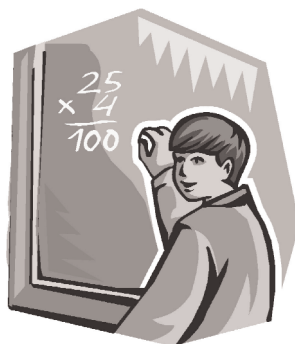
1.7 Έχουν όλα τα προβλήματα σαφή επίλυση;

ΑΠΑΝΤΗΣΗ

Είναι γεγονός ότι δεν έχουν όλα τα προβλήματα σαφή επίλυση. Το πρόβλημα « $1+2=;$ » έχει σαφή επίλυση, το 3. Το πρόβλημα «ποιο είναι το γευστικότερο φαγητό για ένα παιδικό πάρτι» δεν έχει σαφή επίλυση.

Μπορεί, μάλιστα, να έχει πολλά και διαφορετικά αποτελέσματα, γιατί πρόκειται για ένα απόλυτα υποκειμενικό ζήτημα (ερώτημα).

Κάποιοι επιστήμονες αναφέρουν, μεταξύ σοβαρού και αστείου, ότι «συμπέρασμα ονομάζουμε το σημείο στο οποίο οι επιστήμονες βαρέθηκαν να ψάχνουν άλλο».



1.8 Ποιος είναι ο καλύτερος τρόπος για να επιλύσουμε ένα πολύπλοκο πρόβλημα;

ΑΠΑΝΤΗΣΗ

Ο καλύτερος τρόπος για να επιλύσουμε ένα πολύπλοκο πρόβλημα είναι **να το αναλύσουμε σε μικρότερα απλούστερα προβλήματα**. Η τα-

κτική αυτή ονομάζεται και κλίμακα (σκάλα). Αν έχουμε ένα πολύ ψηλό τοίχο και δεν μπορούμε να πηδήσουμε τόσο ψηλά, ώστε να τον περάσουμε, τότε δημιουργούμε (π.χ. με πέτρες ή σκάβοντας στην επιφάνειά του) ένα είδος σκάλας και ανεβαίνουμε λίγο-λίγο, σκαλοπάτι-σκαλοπάτι. Έτσι, αυτό που στην αρχή φάνταζε ακατόρθωτο, έγινε κατορθωτό κάνοντας πολλά μικρά αλλά εύκολα βήματα.

Και το μεγαλύτερο ταξίδι
ξεκινάει με ένα απλό βήμα.
Κινέζικη παροιμία

1.9 Ποιος είναι ο καλύτερος τρόπος για να καθοδηγήσουμε κάποιον/κάποια στην επίλυση ενός προβλήματος;

ΑΠΑΝΤΗΣΗ

Ο καλύτερος τρόπος για να καθοδηγήσουμε κάποιον/κάποια στην επίλυση ενός προβλήματος είναι να του/της δώσουμε **ξεκάθαρες και κατανητές οδηγίες**. Φυσικά, πρέπει ταυτόχρονα να γνωρίζουμε πώς να επιλύσουμε το πρόβλημα.

BIBΛΙΟ ΜΑΘΗΤΗ

Εισαγωγική Δραστηριότητα (σελ. 178)

Προσπαθήστε να δώσετε σε κάποιο συμμαθητή σας σαφείς και ακριβείς οδηγίες, για να παρασκευάσει ένα ποτήρι φρέσκο χυμό πορτοκαλιού.

ΑΠΑΝΤΗΣΗ

Ο μαθητής πρέπει να αναφέρει στην τάξη ότι ο τρόπος σύνταξης των εντολών εξαρτάται και από το άτομο στο οποίο απευθύνονται οι οδηγίες. Όταν απευθυνόμαστε σε έναν άγγλο πρέπει να του μιλήσουμε στα αγγλικά (ή σε οποιαδήποτε άλλη γλώσσα μπορεί να καταλαβαίνει). Το ίδιο ισχύει και στο θέμα της ευφυΐας. Άλλες οδηγίες θα δώσουμε σε ένα παιδί 5 ετών και άλλες οδηγίες θα δώσουμε σε έναν ενήλικα 22 ετών. Με αυτό θέλουμε να τονίσουμε ότι δεν είναι απαραίτητο να δίνουμε τις εντολές πάντα σαν να είμαστε ρομπότ σε ταινία επιστημονικής φαντασίας. Ας δούμε μία προσέγγιση στο ζητούμενο της εισαγωγικής δραστηριότητας.



ΑΠΛΗ ΠΡΟΣΕΓΓΙΣΗ¹

ΟΔΗΓΙΕΣ ΠΑΡΑΣΚΕΥΗΣ ΦΡΕΣΚΟΥ ΧΥΜΟΥ ΠΟΡΤΟΚΑΛΙΟΥ

- ✓ Χρειάζεσαι 3 πορτοκάλια.
- ✓ Χρειάζεσαι 1 στύφτη (εργαλείο κουζίνας που στύβει εσπεριδοειδή).
- ✓ Θα κόψεις τα πορτοκάλια στη μέση.
- ✓ Θα πάρεις το μισό από κάθε πορτοκάλι και θα το πιέσεις, ενώ ταυτόχρονα περιστρέφεις την κομμένη πλευρά του, επάνω στη μυτερή πλευρά του στύφτη.
- ✓ Μόλις το μισό πορτοκάλι σταματήσει να βγάζει χυμό, το αφαιρείς από το στύφτη και το πετάς (εκτός αν θέλει η μητέρα σου να κάνει τις φλούδες γλυκό πορτοκάλι).
- ✓ Επαναλαμβάνεις την ίδια διαδικασία και για τα υπόλοιπα πέντε κομμένα τεμάχια των πορτοκαλιών.
- ✓ Χύνεις το περιεχόμενο του στύφτη μέσα σε ένα ποτήρι.

1. Απευθύνεται σε νοήμονες ενήλικες και ανήλικους συμμαθητές μας.



ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ

1. Πάρε ένα στύφτη.
2. Πάρε 3 πορτοκάλια μεγέθους από 50 έως 60 γραμμάρια.
3. Ακούμπησέ τα στον πάγκο της κουζίνας.
4. Πάρε ένα μαχαίρι.
5. Πάρε το πορτοκάλι που είναι πιο κοντά σου.
6. Τοποθέτησε το μαχαίρι στη μέση του πορτοκαλιού. Μέση είναι το σημείο που ισαπέχει από το κοτσάνι και το κάτω σημάδι που υπάρχει σε κάθε πορτοκάλι.
7. Κόψε το πορτοκάλι στη μέση.
8. Τοποθέτησε τα δύο κομμάτια κοντά στο στύφτη.
9. Επανάλαβε άλλες 2 φορές τις διαδικασίες από το 5 έως το 8 και μετά πήγαινε στο 10.
10. Πάρε το κομμάτι του πορτοκαλιού που είναι πιο κοντά σου.
11. Τοποθέτησε το κέντρο της κυκλικής επιφάνειας του τεμαχίου επάνω στη μυτερή ακμή του στύφτη.
12. Πίεσε με δύναμη 700 γραμμαρίων το τεμάχιο προς τα κάτω.
13. Περίστρεψε το τεμάχιο δεξιόστροφα με ταχύτητα 40 στροφές το λεπτό.

14. Επανάλαβε το 13 μέχρι να πάψει η ροή υγρών. Μετά σταμάτα την περιστροφή και πήγαινε στο 15.
15. Σήκωσε το τεμάχιο 20 εκατοστά πάνω από το σύφτη.
16. Πέτα το τεμάχιο πορτοκαλιού στα σκουπίδια.
17. Επανάλαβε 5 φορές τις διαδικασίες από το 10 έως το 16 και μετά πήγαινε στο 18.
18. Πιάσε το σύφτη
19. Πάρε ένα ποτήρι.
20. Χύσε το περιεχόμενο του σύφτη μέσα στο ποτήρι.
21. Αν το ποτήρι γεμίσει, σταμάτα τη διαδικασία.
21. Αν το περιεχόμενο του σύφτη αδειάσει, σταμάτα τη διαδικασία.

1.10 Τι ονομάζουμε αλγόριθμο;

ΑΠΑΝΤΗΣΗ

Αλγόριθμο ονομάζουμε τη σαφή και ακριβή περιγραφή μιας σειράς ξεχωριστών οδηγιών-βημάτων, με σκοπό την επίλυση ενός προβλήματος ή την εκτέλεση μιας λειτουργίας. Δεν είναι απαραίτητο ο αλγόριθμος να οδηγεί σε λύση ενός προβλήματος. Μπορεί να δημιουργεί μια κατάσταση, να καταγράφει ή να υπολογίζει δεδομένα κ.λπ.

1.11 Από πού πήρε το όνομά του ο «αλγόριθμος»;

ΑΠΑΝΤΗΣΗ

Από το μαθηματικό Μοχάμετ Ιμπν Μούζα Αλ Χουβαρίζμι (Μωάμεθ, γιος του Μούζα από το Χουβαρίζμι) που έζησε στην Περσία από το 780 έως το 850 μ.Χ. Θεωρείται ο πρώτος που εισήγαγε την έννοια του αλγορίθμου αναφερόμενος σε μια **μαθηματική επεξεργασία αριθμών**. Η λέξη αλγόριθμος προήρθε από την παράφραση μέρους του ονόματός του στα λατινικά: Αλ Χουβαρίζμι => algorismus. Στα τέλη του 17ου αιώνα μ.Χ. η ονομασία συνδυάστηκε, εξαιτίας ακουστικής παραδρομής, με την ελληνική λέξη «αριθμός» και μετατράπηκε στη λέξη αλγόριθμος².

2. Στο σημείο αυτό προκύπτει το εξής **ερώτημα**: Αν η έννοια του αλγορίθμου πρωτοεμφανίστηκε το 800 μ.Χ., τότε πώς μπορούσαν οι μεγάλοι μαθηματικοί της αρχαιότητας να λύσουν δυσεπίλυτα προβλήματα που απαιτούν την έννοια του αλγορίθμου και στη σύγχρονη εποχή καταφέραμε να τα λύσουμε μόνο με τη βοήθεια ηλεκτρονικού υπολογιστή;

ΕΠΙΠΛΕΟΝ ΔΡΑΣΤΗΡΙΟΤΗΤΑ

Ο μαθητής μπορεί να ψάξει στο Διαδίκτυο για το Βοϊκό πρόβλημα που έθεσε ο Αρχιμήδης, καθώς και για την αλληλογραφία του με άλλους μαθηματικούς της εποχής του και τις οδηγίες που έδινε (αλγορίθμους) για την επίλυση προβλημάτων.



ΒΙΒΛΙΟ ΜΑΘΗΤΗ

Ερωτήσεις (σελ. 179)

? Πριν προχωρήσουμε παρακάτω προσπάθησε να απαντήσεις στις ακόλουθες ερωτήσεις:

1. Τι θα συμβεί αν ξεχάσουμε την οδηγία 9 στον παρακάτω αλγόριθμο;

1. Άνοιξε το μάτι της κουζίνας στο 2.
2. Βάλε 3 λίτρα νερό σε μία κατσαρόλα χωρητικότητας 4 λίτρων.
3. Τοποθέτησε την κατσαρόλα στο μάτι της κουζίνας, που έχεις ήδη ανάψει.
4. Πρόσθεσε στην κατσαρόλα μία κουταλιά του καφέ αλάτι.
5. Περίμενε μέχρι να βράσει το νερό.
6. Βγάλε τα μακαρόνια από το πακέτο.
7. Βάλε τα μακαρόνια στην κατσαρόλα.
8. Ανακάτεψε τα μακαρόνια για 10 λεπτά.
9. Κλείσε το μάτι της κουζίνας που άνοιξες.
10. Βγάλε την κατσαρόλα από το μάτι της κουζίνας.
11. Άδειασε τα μακαρόνια από την κατσαρόλα σε ένα σουρωτήρι.
12. Ρίξε κρύο νερό από τη βρύση στα μακαρόνια για 20 δευτερόλεπτα.
13. Άφησε για 2 λεπτά τα μακαρόνια να στραγγίξουν.
14. Σερβίρισε τα μακαρόνια στο πιάτο.
15. Πρόσθεσε σε κάθε πιάτο 3 κουταλιές της σούπας τριμμένο τυρί.

ΑΠΑΝΤΗΣΗ

Σε παλαιού τύπου «μάτι» κουζίνας το μάτι θα πυρώσει και θα καταναλώσουμε άσκοπα ρεύμα μέχρι να ανακαλύψουμε ότι ξεχάσαμε να το κλείσουμε. Σε σύγχρονο «μάτι» κουζίνας το μάτι θα σβήσει από μόνο του λίγα δευτερόλεπτα μετά την απομάκρυνση της κατσαρόλας από πάνω του.

2. Μπορούμε να αντιμεταθέσουμε τις οδηγίες 7 και 8;

ΑΠΑΝΤΗΣΗ

Βεβαίως και μπορούμε, αλλά αυτό θα οδηγήσει σε **λογικό λάθος** το οποίο δεν θα οδηγήσει στο επιθυμητό αποτέλεσμα (δηλαδή να φτιάξουμε μακαρόνια που να τρώγονται). Το σημαντικό που πρέπει να γνωρίζουμε είναι ότι στον προγραμματισμό τίποτα, εκτός από τη λογική μας, δεν πρόκειται να μας εμποδίσει να κάνουμε λογικά σφάλματα, και γι' αυτό πρέπει να είμαστε διαρκώς συγκεντρωμένοι και να έχουμε το νου μας όταν προγραμματίζουμε μία γλώσσα.

3. Τι θα συμβεί αν αντικαταστήσουμε την οδηγία στο βήμα 4 με την οδηγία «πρόσθεσε αλάτι»;

ΑΠΑΝΤΗΣΗ

Όπως αναφέραμε και προηγουμένως το τι θα συμβεί εξαρτάται από την ευφυΐα του συστήματος (ή του ανθρώπου) στο οποίο απευθυνόμαστε. Αν πρόκειται για σύστημα μηχανής (λογισμικού) πρέπει προηγουμένως να έχουμε προσδιορίσει μια διαδικασία που να ορίζει τι σημαίνει «πρόσθεσε αλάτι». Μια τέτοια διαδικασία θα μπορούσε να ήταν μία μέτρηση του όγκου του νερού και ανάλογα με τις τιμές που θα είχαμε δώσει θα προσέθετε και τα αντίστοιχα γραμμάρια αλατιού. Αν δεν έχουμε ορίσει μια τέτοια διαδικασία, τότε θα πάρουμε ένα μήνυμα λάθους του τύπου «δεν ξέρω τι σημαίνει ΒΑΛΕ ΑΛΑΤΙ». Αν απευθυνόμαστε σε άνθρωπο στην ουσία του λέμε: «βάλε όσο αλάτι σου αρέσει». Κάθε άνθρωπος έχει τα δικά του γούστα, οπότε βάζει όσο αλάτι θεωρεί αυτός απαραίτητο, λιγότερο ή περισσότερο από μία κουταλιά του γλυκού.

4. Αν αντιμεταθέσουμε τις οδηγίες 1 και 2, θα υπάρξει κάποιο πρόβλημα στον αλγόριθμο;

ΑΠΑΝΤΗΣΗ

Όχι, κανένα απολύτως. Μπορούμε να πούμε, μάλιστα, ότι οικολογικά είναι σωστότερος ο αλγόριθμος, καθώς έτσι σπαταλάμε λιγότερο ρεύμα και συμβάλλουμε με αυτόν τον τρόπο στην προστασία του περιβάλλοντος.

1.12 Ποιο πρέπει να είναι το βασικό χαρακτηριστικό ενός αλγορίθμου;

ΑΠΑΝΤΗΣΗ

Το βασικό χαρακτηριστικό ενός αλγορίθμου πρέπει να είναι η έκφραση των εντολών του **με απλά λόγια**, ώστε να είναι απόλυτα κατανοητές από το δέκτη των εντολών.

1.13 Γιατί λέμε ότι τα πολλά διαφορετικά προγράμματα που μπορεί να εκτελεστούν σε έναν υπολογιστή αποτελούν, σήμερα, τον κύριο λόγο χρήσης τους από τόσο μεγάλο μέρος του πληθυσμού;

ΑΠΑΝΤΗΣΗ

Με τη δυνατότητα χρήσης πολλών διαφορετικών προγραμμάτων ο υπολογιστής, ως εργαλείο, γίνεται **πολλαπλά χρήσιμος**, καθώς μπορεί να εκτελέσει πολλές εργασίες σε αντίθεση με τις περισσότερες από τις υπόλοιπες συσκευές που μας περιβάλλουν. Για παράδειγμα, το ψυγείο μόνο ψύχει, το ρολόι δείχνει μόνο την ώρα και μας ειδοποιεί (ξυπνητήρι), η παιχνιδομηχανή παίζει παιχνίδια. Ο υπολογιστής, όμως, μπορεί να παίζει παιχνίδια, να εκτελεί μαθηματικές πράξεις, να παίζει μουσική, να αναπαράγει ταινίες, να ελέγχει το σύστημα ασφαλείας του κτιρίου, να στέλνει μηνύματα στο Διαδίκτυο, να στέλνει και να δέχεται φαξ (τηλεομοιοτύπα) κ.ά.

1.14 Τι είναι ένα πρόγραμμα;

ΑΠΑΝΤΗΣΗ

Ένα πρόγραμμα είναι η αναπαράσταση ενός **αλγορίθμου** γραμμένη σε γλώσσα κατανοητή από τον υπολογιστή (δηλαδή από το λειτουργικό του σύστημα) για τον οποίο προορίζεται το πρόγραμμα. Όταν εκτελείται ένα πρόγραμμα, τότε λέμε ότι το πρόγραμμα, αυτό **«τρέχει»**. Για παράδειγμα, όταν λέμε ότι «το τάδε πρόγραμμα τρέχει σε Linux» εννοούμε ότι οι εντολές του μπορούν να εκτελεστούν μόνο σε έναν υπολογιστή που έχει το λειτουργικό σύστημα Linux.



ΓΝΩΡΙΖΕΙΣ ΟΤΙ...

Ο υπολογιστής δεν αποτελείται μόνο από το μηχάνημα. Είναι συνδυασμός μηχανής και λειτουργικού συστήματος. Το ίδιο λειτουργικό σύστημα μπορεί να υπάρχει σε υπολογιστές από διαφορετικό υλικό. Για παράδειγμα το Linux μπορεί να χρησιμοποιηθεί σε κάθε, σχεδόν, υπολογιστή που κυκλοφορεί σήμερα. Έτσι, όταν αγοράζουμε ένα πρόγραμμα το πρώτο που πρέπει να ελέγχουμε είναι αν είναι συμβατό (αν τρέχει) στο λειτουργικό σύστημα του υπολογιστή μας.



1.15 Από τι αποτελείται ένα πρόγραμμα;

ΑΠΑΝΤΗΣΗ

Ένα πρόγραμμα αποτελείται από μία σειρά **εντολών** που δίνονται στο λειτουργικό σύστημα του υπολογιστή με σκοπό να εκτελέσει κάποια συγκεκριμένη εργασία ή να κάνει κάποιους υπολογισμούς.

1.16 Τι είναι ο προγραμματισμός;

ΑΠΑΝΤΗΣΗ

Η εργασία **σύνταξης** των εντολών που απαρτίζουν ένα πρόγραμμα υπολογιστή ονομάζεται **προγραμματισμός**. Τα άτομα που έχουν τις γνώσεις για να κάνουν αυτή την εργασία ονομάζονται **προγραμματιστές** ή **προγραμματιστές λογισμικού**.

1.17 Τι συμβαίνει κάθε φορά που εκτελείται ένα πρόγραμμα;

ΑΠΑΝΤΗΣΗ

Κάθε φορά που εκτελείται ένα πρόγραμμα δίνεται (από το πρόγραμμα) ένα σύνολο εντολών στην Κεντρική Μονάδα Επεξεργασίας (αφού περάσουν πρώτα από τη μνήμη RAM). Η Κεντρική Μονάδα Επεξεργασίας εκτελεί πιστά τις εντολές αυτές. Αν το αποτέλεσμα του προγράμματος θα είναι το επιθυμητό ή όχι εξαρτάται από το πρόγραμμα και από τα δεδομένα που χειρίζεται και όχι από την Κεντρική Μονάδα Επεξεργασίας.

1.18 Τι είναι η γλώσσα μηχανής;

ΑΠΑΝΤΗΣΗ

Ο υπολογιστής και τα περιφερειακά του συστήματα είναι ψηφιακές συσκευές που λειτουργούν με το δυαδικό σύστημα. Αυτό σημαίνει ότι η γλώσσα που χρησιμοποιούν για να «συνεννοούνται» μεταξύ τους και με το περιβάλλον αποτελείται από δύο και μόνο ψηφία: το μηδέν (0) και το ένα (1). Αυτή η γλώσσα που έχει σαν αλφάβητο το 0 και το 1 και που χρησιμοποιείται για την επικοινωνία των ψηφιακών συσκευών του υπολογιστή ονομάζεται **γλώσσα μηχανής**.

1.19 Τι είναι οι γλώσσες προγραμματισμού;

ΑΠΑΝΤΗΣΗ

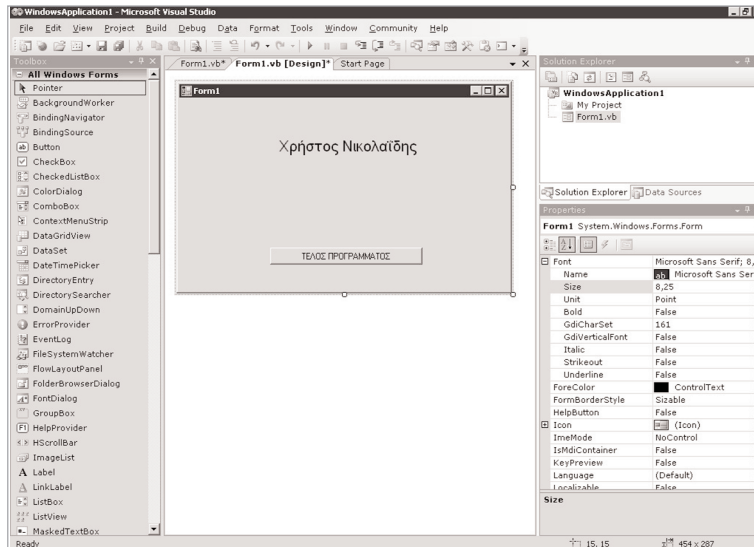
Ο υπολογιστής και τα περιφερειακά του, ως ψηφιακές συσκευές, χρησιμοποιούν τη γλώσσα μηχανής (0 και 1). Ο άνθρωπος είναι πολύ δύσκολο, αν όχι ακατόρθωτο, να μεταφράσει τις σκέψεις και τα θέλω του στο δυαδικό σύστημα. Γι' αυτόν το λόγο έχουν επινοηθεί κάποιες **τεχνητές γλώσσες** που λειτουργούν ως **ενδιάμεσοι** μεταξύ της φυσικής γλώσσας (αυτή που μιλάμε και καταλαβαίνουμε) και της γλώσσας μηχανής (αυτή που «καταλαβαίνει» ο υπολογιστής), έτσι **ώστε να μπορούμε να δίνουμε εντολές** στον υπολογιστή. Αυτές οι γλώσσες ονομάζονται **γλώσσες προγραμματισμού**. Αν θέλουμε να φτιάξουμε ένα πρόγραμμα για τον υπολογιστή μας, πρέπει να το γράψουμε σε κάποια γλώσσα προγραμματισμού.

1.20 Ποια είναι τα βασικά χαρακτηριστικά μιας γλώσσας προγραμματισμού;

ΑΠΑΝΤΗΣΗ

Τα βασικά χαρακτηριστικά μιας γλώσσας προγραμματισμού είναι τρία:

- Αλφάβητο
- Λεξιλόγιο
- Συντακτικό



1.21 Τι είναι το αλφάβητο μιας γλώσσας προγραμματισμού;

ΑΠΑΝΤΗΣΗ

Το αλφάβητο μιας γλώσσας προγραμματισμού είναι το σύνολο των χαρακτήρων που χρησιμοποιούνται από τη γλώσσα.



ΓΝΩΡΙΖΕΙΣ ΟΤΙ...

Στα βιβλία προγραμματισμού δεν αναφέρεται πουθενά ο όρος «αλφάβητο» της γλώσσας προγραμματισμού. Όλες οι γλώσσες προγραμματισμού που κυκλοφορούν στην αγορά, εκτός από κάποιες εξαιρέσεις, χρησιμοποιούν το λατινικό αλφάβητο για τις εντολές και όλους τους χαρακτήρες (λατινικούς, ελληνικούς, αραβικούς κ.ά.) για την ονομασία μεταβλητών. Οι εξαιρέσεις είναι γλώσσες που μεταφράστηκαν για εκπαιδευτικούς λόγους (π.χ. η Logo) ή για πρακτικούς λόγους όπως στην Κίνα που το αλφάβητο είναι εντελώς διαφορετικό από το λατινικό, αν και εκεί ακόμη προτιμούν την «κλασική» έκδοση με τους λατινικούς χαρακτήρες.

1.22 Τι είναι το λεξιλόγιο μιας γλώσσας προγραμματισμού;

ΑΠΑΝΤΗΣΗ

Το λεξιλόγιο μιας γλώσσας προγραμματισμού είναι το σύνολο των λέξεων (ή καλύτερα των **εντολών**) που χρησιμοποιεί η γλώσσα αυτή³.

3. Ο όρος *λέξη* δεν χρησιμοποιείται από κανένα προγραμματιστή σήμερα. Οι γλώσσες προγραμματισμού δεν έχουν «*λέξεις*», έχουν *εντολές* (*commands*), *λειτουργίες* (*functions*), *σταθερές* (*constants*), *μεταβλητές* (*variables*) κ.λπ. Μην ξεχνάτε ότι το πρόγραμμα είναι μια σειρά εντολών και όχι μια ποσολογία.

1.23 Τι είναι το συντακτικό μιας γλώσσας προγραμματισμού;

ΑΠΑΝΤΗΣΗ

Το συντακτικό μιας γλώσσας προγραμματισμού είναι το σύνολο των **κανόνων** με τις οποίες συντάσσονται οι εντολές της. Για παράδειγμα, η σύνταξη της υποθετικής εντολής **ΤΥΠΩΣΕ**, η οποία εμφανίζει μια λέξη στην οθόνη, θα μπορούσε να είναι η εξής:

```
ΤΥΠΩΣΕ "ΚΕΙΜΕΝΟ"
```

Αν τη συντάξουμε με την παρακάτω μορφή

```
"ΚΕΙΜΕΝΟ" ΤΥΠΩΣΕ
```

τότε θα έχουμε ένα μήνυμα λάθους και η εντολή δεν θα εκτελεστεί.

Ας δούμε ένα πραγματικό παράδειγμα. Η εντολή `echo`⁴ στη γλώσσα προγραμματισμού PHP –η οποία χρησιμοποιείται για τη δημιουργία ιστότοπων στο Διαδίκτυο– συντάσσεται ως εξής:

```
<? echo "κείμενο"; ?>
```

Έτσι, αν θέλουμε στην ιστοσελίδα μας να εμφανιστεί η λέξη ΚΑΛΗΜΕΡΑ θα δώσουμε (με τη γλώσσα προγραμματισμού PHP) την εντολή:

```
<? echo "ΚΑΛΗΜΕΡΑ"; ?>
```

4. *Echo*: εντολή που χρησιμοποιείται για την προβολή χαρακτήρων στην οθόνη.

1.26 Τι είναι ο κειμενογράφος ενός περιβάλλοντος προγραμματισμού;

ΑΠΑΝΤΗΣΗ

Πρόκειται για ένα εξειδικευμένο κειμενογράφο που χρησιμεύει στη **σύνταξη** των **εντολών** που απαρτίζουν το πρόγραμμα. Με τον όρο σύνταξη εννοούμε τη δημιουργία αλλά και την επεξεργασία-διόρθωση των ήδη γραμμένων εντολών. Όταν λέμε εξειδικευμένος κειμενογράφος, εννοούμε ότι προσφέρει στον προγραμματιστή κάποιες ευκολίες που έχουν να κάνουν με τη σύνταξη των εντολών. Σε γενικές γραμμές πρόκειται για έναν κλασικό κειμενογράφο, όπως το Σημειωματάριο των Windows.

1.27 Τι είναι ο μεταφραστής ενός περιβάλλοντος προγραμματισμού;

ΑΠΑΝΤΗΣΗ

Πρόκειται για τη διαδικασία που α. **ελέγχει τον κώδικα για τυχόν συντακτικά λάθη** και β. μεταφράζει τη γλώσσα προγραμματισμού σε γλώσσα μηχανής. Ο **μεταφραστής**, όπως άλλωστε και ο κειμενογράφος, στο 99% των περιπτώσεων είναι ενσωματωμένος στο περιβάλλον προγραμματισμού⁵. Οι μεταφραστές χωρίζονται σε δύο βασικές κατηγορίες: στους **μεταγλωττιστές (compilers)** και στους **διερμηνείς (interpreters)**.

5. Παρ' ότι η γλώσσα προγραμματισμού είναι κάτι διαφορετικό, συνηθίζουμε να ονομάζουμε «γλώσσα» το πρόγραμμα που μας επιτρέπει να προγραμματίζουμε σε αυτήν τη γλώσσα. Για παράδειγμα, ζητάμε να αγοράσουμε τη γλώσσα «Delphi», ενώ στην πραγματικότητα ζητάμε να αγοράσουμε το πρόγραμμα που μέσα από το δικό του περιβάλλον προγραμματισμού μας επιτρέπει να δημιουργήσουμε προγράμματα χρησιμοποιώντας τη γλώσσα Delphi.

1.28 Ποιες είναι οι διαφορές μεταξύ μεταγλωττιστών και διερμηνέων;

ΑΠΑΝΤΗΣΗ

Ένας μεταγλωττιστής (compiler) θα ελέγξει όλη τη σειρά εντολών για συντακτικά λάθη και μετά θα τη μετατρέψει σε **εκτελέσιμο πρόγραμμα**, δηλαδή σε μια σειρά από 0 και 1 που θα τροφοδοτηθούν στην Κε-

ντρική Μονάδα Επεξεργασίας, όταν εκτελέσουμε το πρόγραμμα. Ο διερμηνέας (interpreter) ελέγχει μία εντολή κάθε φορά για συντακτικά λάθη, στη συνέχεια την εκτελεί και μετά ελέγχει την επόμενη οδηγία.

1.29 Τι ονομάζουμε συντακτικό λάθος κατά τον προγραμματισμό;

ΑΠΑΝΤΗΣΗ

Όπως προαναφέρθηκε, ένα από τα βασικά χαρακτηριστικά μιας γλώσσας προγραμματισμού είναι το συντακτικό, δηλαδή ο τρόπος σύνταξης της κάθε εντολής. Αν δώσουμε προς εκτέλεση μία εντολή με λάθος τρόπο, δηλαδή όχι με τη σωστή της σύνταξη, τότε το περιβάλλον προγραμματισμού θα μας εμφανίσει ένα μήνυμα λάθους. Το λάθος αυτό ονομάζεται **συντακτικό λάθος**. Τα περισσότερα περιβάλλοντα προγραμματισμού είναι ιδιαίτερα αυστηρά ως προς την τήρηση του συντακτικού και δεν επιτρέπουν το παραμικρό λάθος⁶.

6. Τα τελευταία χρόνια τα περιβάλλοντα προγραμματισμού τείνουν να είναι πολύ αυστηρά κάνοντας έτσι τη χρήση μιας γλώσσας προγραμματισμού πολύ δύσκολη και εκνευριστική υπόθεση. Το ανάλογο παράδειγμα στην καθημερινή μας γλώσσα είναι να έρθει ένας ξένος και να σας ρωτήσει: «θέλω να πάω στο Αθήνα» και εσείς να του πείτε «δεν κατάλαβα τίποτα από ό,τι μου είπες», μόνο και μόνο επειδή έκανε το λάθος να πει «στο Αθήνα» αντί «στην Αθήνα». Η γλώσσα προγραμματισμού HTML (για δημιουργία ιστοσελίδων) έγινε μέσα σε λίγους μήνες γνωστή σε όλο τον κόσμο ακριβώς επειδή είχε «καθαρό συντακτικό». Τα τελευταία χρόνια, κάποιοι πανεπιστημιακοί παράγοντες προσπαθούν να τη μετατρέψουν σε «γλώσσα με αυστηρό συντακτικό» αλλιώς, όπως είναι φυσικό, δεν βρίσκει αποδοχή από το ευρύ κοινό.

ΕΠΙΠΛΕΟΝ ΔΡΑΣΤΗΡΙΟΤΗΤΑ

Ψάξτε στο Διαδίκτυο ή ρωτήστε φιλικά σας πρόσωπα μεγαλύτερης ηλικίας από εσάς για την προσπάθεια επιβολής της Καθαρεύουσας κατά τη διάρκεια του 19ου και του 20ού αιώνα.

1.30 Τι ονομάζουμε λογικό λάθος κατά τον προγραμματισμό;

ΑΠΑΝΤΗΣΗ

Το λάθος που δημιουργείται από το λανθασμένο τρόπο χρήσης των σω-

στά συντεταγμένων εντολών ονομάζεται **λογικό λάθος**⁷. Στην ουσία πρόκειται για σωστές εντολές, εκτελεσμένες με λάθος σειρά ή με σειρά που δεν αποφέρει τα αναμενόμενα αποτελέσματα. Αυτό που πρέπει να κάνουμε είναι να ελέγξουμε μία προς μία τις εντολές του αλγορίθμου, ώστε να εντοπίσουμε σε ποιο σημείο των εντολών υπάρχει πρόβλημα. Στον υπολογιστή, όπως και στους ανθρώπους, αν θέλουμε να μεταδώσουμε με ακρίβεια το μήνυμά μας δεν πρέπει να θεωρούμε τίποτα αυτονόητο.



7. Στην επιστήμη της Ψυχολογίας αναφέρεται ότι: «Νεύρωση είναι η μη έξυπνη συμπεριφορά ενός έξυπνου ανθρώπου». Κατά τον ίδιο τρόπο λειτουργεί και το λογικό λάθος στον προγραμματισμό: κάνουμε σωστά πράγματα με λάθος σειρά.

ΒΙΒΛΙΟ ΜΑΘΗΤΗ

Ερωτήσεις (σελ. 185)

1. Γιατί πρέπει να κατανοούμε καλά ένα πρόβλημα, πριν να το επιλύσουμε;

ΑΠΑΝΤΗΣΗ

Είναι αυτονόητο ότι πρέπει να κατανοήσουμε ένα πρόβλημα πριν την επίλυσή του. Δεν υπάρχει «καλή κατανόηση» και «μέτρια κατανόηση». Ή έχουμε κατανοήσει το πρόβλημα ή δεν το έχουμε κατανοήσει. Αν δεν το έχουμε κατανοήσει «καλά», τότε απλά δεν το έχουμε κατανοήσει. Η κατανόηση των παραμέτρων ενός προβλήματος μας βοηθά στο να θέσουμε τις **λογικές βάσεις** για να το επιλύσουμε. **Η βάση για την επίλυση κάθε προβλήματος είναι να γνωρίζουμε το πρόβλημα.** Αν δεν έχουμε καταλάβει «καλά» ένα πρόβλημα, τότε δεν προσπαθούμε να λύσουμε αυτό καθαυτό το πρόβλημα αλλά μια παραλλαγή του: αυτήν που καταλάβαμε.

2. Ποιες διαδικασίες μας βοηθούν στην κατανόηση ενός προβλήματος;

ΑΠΑΝΤΗΣΗ

Οι διαδικασίες που μας βοηθούν στην κατανόηση ενός προβλήματος είναι:

- α. **Η σωστή διατύπωση των δεδομένων του προβλήματος.** Το πρόβλημα πρέπει να μας παρουσιάζεται με τρόπο κατανοητό και σε γλώσσα που να

την καταλαβαίνουμε. Ακόμη και το απλούστατο $1 + 1 =$; αν γίνει η εκφώνησή του στα σουαχίλι, δεν θα μπορέσουμε να το λύσουμε, αν δεν γνωρίζουμε τη γλώσσα σουαχίλι!

- β. **Η πλήρης παρουσίαση των δεδομένων του προβλήματος.** Αν δεν έχουμε στη διάθεσή μας όλα τα δεδομένα που είναι απαραίτητα για την επίλυση ενός προβλήματος, τότε δεν θα μπορέσουμε ποτέ να το κατανοήσουμε και φυσικά να το λύσουμε.
- γ. **Η σαφής διατύπωση του ζητούμενου.** Τι μας ζητάει το πρόβλημα να βρούμε; Αν δεν ξέρουμε τι ψάχνουμε να βρούμε τότε, ακόμη και αν το βρούμε κατά λάθος, δεν θα είμαστε σίγουροι ότι αυτό ήταν το ζητούμενο.

3. Τι είναι ένας αλγόριθμος;

ΑΠΑΝΤΗΣΗ

Αλγόριθμο ονομάζουμε τη σαφή και ακριβή περιγραφή μιας σειράς ξεχωριστών οδηγιών-βημάτων, με σκοπό την επίλυση ενός προβλήματος.

4. Ποιες είναι οι βασικές ιδιότητες ενός αλγορίθμου;

ΑΠΑΝΤΗΣΗ

Οι ιδιότητες που πρέπει να έχει ένας αλγόριθμος είναι:

- α. **η ρύθμιση των εντολών ώστε κάποια στιγμή να τελειώσει,**
- β. **οι εντολές του να έχουν ακρίβεια και σαφήνεια.**

Στο σημείο αυτό να προσθέσουμε ότι οι ιδιότητες αυτές δεν χαρακτηρίζουν υποχρεωτικά έναν αλγόριθμο, αλλά είναι οι ιδιότητες που πρέπει (σύμφωνα με ορισμένους) να έχει ένας αλγόριθμος. Τι σημαίνει αυτό; Ένας αλγόριθμος δεν πρέπει απαραίτητα να είναι ακριβής, μπορεί να έχει και λάθη ή και ασάφειες (ψάξτε στο Διαδίκτυο για την ασαφή λογική [= fuzzy logic]). Δεν είναι απαραίτητο να είναι σαφής και απλός, μπορεί να είναι περίπλοκος αλλά αποτελεσματικός. Τέλος, δεν είναι απαραίτητο ότι πρέπει κάποια στιγμή να τελειώνει. Για παράδειγμα, ο αλγόριθμος που ρυθμίζει ένα βηματοδότη δεν τελειώνει, επειδή, αν τελείωνε, θα «τελείωνε» μαζί του και η ζωή του ασθενούς!



5. Τι είναι πρόγραμμα;

ΑΠΑΝΤΗΣΗ

Πρόγραμμα ονομάζουμε την αναπαράσταση ενός αλγορίθμου γραμμένη σε μια γλώσσα που καταλαβαίνει ο υπολογιστής. Ένα πρόγραμμα αποτελείται από μια σειρά **εντολών** προς τον υπολογιστή με σκοπό την εκτέλεση κάποιων λειτουργιών ή υπολογισμού.

6. Ποιο είναι το αλφάβητο της γλώσσας μηχανής του υπολογιστή;

ΑΠΑΝΤΗΣΗ

Το αλφάβητο της γλώσσας μηχανής του υπολογιστή αποτελείται από δύο χαρακτηριστικές, το μηδέν και το ένα (0 και 1).

7. Ποια είναι τα βασικά χαρακτηριστικά μιας γλώσσας προγραμματισμού;

ΑΠΑΝΤΗΣΗ

Κάθε γλώσσα προγραμματισμού έχει ως βασικά χαρακτηριστικά: **το αλφάβητο, το λεξιλόγιο και το συντακτικό.**

8. Ποια είναι τα στάδια για την εκτέλεση ενός αλγορίθμου από την Κ.Μ.Ε;

ΑΠΑΝΤΗΣΗ

Εξαρτάται από το πώς εκτελείται ο αλγόριθμος. Αν εκτελείται μέσα από ένα εκτελέσιμο αρχείο (πρόγραμμα), τότε γίνεται ως εξής:

ΠΡΟΓΡΑΜΜΑ (στο οποίο εμπεριέχεται ο αλγόριθμος) → **ΜΕΤΑΤΡΟΠΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ ΓΛΩΣΣΑ ΜΗΧΑΝΗΣ (0 ΚΑΙ 1)** → **ΕΚΤΕΛΕΣΗ ΣΤΗΝ Κ.Μ.Ε.**

Αν, όμως, ο αλγόριθμος εκτελείται μέσα από κάποιο προγραμματιστικό περιβάλλον (π.χ. μεταγλωττιστή/διερμηνέα), τότε το προγραμματιστικό περιβάλλον αναλαμβάνει τη μεταγλώττιση του αλγορίθμου σε γλώσσα μηχανής, οπότε και εκτελείται από την Κ.Μ.Ε. Αυτό που πρέπει να γνωρίζουμε είναι ότι σε κάθε περίπτωση η Κ.Μ.Ε. λαμβάνει και εκτελεί τις εντολές σε γλώσσα μηχανής, δηλαδή σε 0 και σε 1.