

Ο Βαρκάρης ακολουθώντας πιστά (υλοποιώντας) τις οδηγίες του αλγορίθμου μπορεί να μεταφέρει με επιτυχία τα ζώα του και το καφάσι με τα χόρτα στην απέναντι όχθη του ποταμού.

2ο Παράδειγμα

Θέλουμε να περιγράψουμε σε ένα μικρό παιδί πώς θα δημιουργήσει με τις πατούσες του ένα τετράγωνο στην άμμο. Αν το παιδί δε γνωρίζει τι σχήμα θέλουμε να αποτυπωθεί στην άμμο, ποιες είναι οι κατάλληλες οδηγίες που πρέπει να του δώσουμε;

Κατ' αρχάς πρέπει να αναλύσουμε την έννοια «τετράγωνο»:

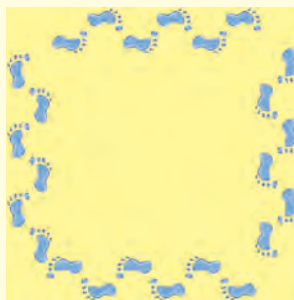
- Ένα τετράγωνο είναι ένα κλειστό γεωμετρικό σχήμα με **4 ίσες πλευρές**. Άρα, για να σχηματίσουμε τις πλευρές, πρέπει κάθε φορά να κάνουμε τον ίδιο αριθμό βημάτων.
- Ένα τετράγωνο έχει 4 ορθές γωνίες δηλαδή **4 γωνίες των 90°**. Άρα, μόλις σχηματίζουμε μία πλευρά πρέπει να γυρνάμε κατά 90° γύρω από τον εαυτό μας και πάντοτε με την ίδια φορά.

Αφού έχουμε κατανοήσει την έννοια «τετράγωνο», μπορούμε να πειραματιστούμε δίνοντας τις ακόλουθες οδηγίες στο παιδί:

Αρχή του αλγορίθμου

1. Περπάτησε 5 βήματα μπροστά.
2. Στρίψε δεξιά κατά ενενήντα μοίρες.
3. Περπάτησε 5 βήματα μπροστά.
4. Στρίψε δεξιά κατά ενενήντα μοίρες.
5. Περπάτησε 5 βήματα μπροστά.
6. Στρίψε δεξιά κατά ενενήντα μοίρες.
7. Περπάτησε 5 βήματα μπροστά.

Τέλος του αλγορίθμου



Η υλοποίηση του παραπάνω αλγορίθμου ήταν επιτυχής στο σχεδιασμό ενός τετραγώνου. Μερικές φορές, όμως, ένας αλγόριθμος μπορεί να μη μας δώσει τα προσδοκώμενα αποτελέσματα. Τότε είμαστε υποχρεωμένοι να γυρίσουμε πίσω στις εντολές που δώσαμε και να ελέγξουμε πού κάναμε λάθος. Στη συνέχεια αντικαθιστούμε τις λανθασμένες εντολές με τις σωστές και υλοποιούμε ξανά τον αλγόριθμο. Αυτή η ανατροφοδοτούμενη μορφή σχεδιασμού μας βοηθάει να καταλάβουμε καλύτερα το πρόβλημα και την επίλυσή του. Αν, για παράδειγμα, κάνουμε λάθος στο σχεδιασμό του αλγορίθμου του τετραγώνου, η διαδικασία εύρεσης του λάθους θα μας βοηθήσει να κατανοήσουμε καλύτερα την έννοια του τετραγώνου.

1.4 Υλοποίηση Αλγορίθμου με υπολογιστή – Προγραμματισμός

Τα πολλά διαφορετικά προγράμματα που μπορεί να εκτελεστούν σε έναν υπολογιστή, αποτελούν τον κύριο λόγο που χρησιμοποιούμε σήμερα τους υπολογιστές για διαφορετικές χρήσεις. Οι υπολογιστές χρησιμοποιούνται σε επιχειρήσεις-οργανισμούς, στη δημόσια διοίκηση, σε εκπαιδευτικά ιδρύματα, αλλά και σε σπίτια. Κάθε υπολογιστής γίνεται μια διαφορετική μηχανή ανάλογα με το πρόγραμμα που εκτελεί και αυτό είναι το μεγάλο του πλεονέκτημα. Τι είναι όμως ένα πρόγραμμα;

Ένα **πρόγραμμα** είναι η αναπαράσταση ενός αλγορίθμου γραμμένη σε γλώσσα κατανοητή για έναν υπολογιστή. Ένα πρόγραμμα, δηλαδή, αποτελείται από μία σειρά **εντολών** που δίνονται στον υπολογιστή με σκοπό να εκτελέσει κάποια συγκεκριμένη λειτουργία ή να υπολογίσει κάποιο επιθυμητό αποτέλεσμα. Η εργασία σύνταξης των προγραμμάτων ονομάζεται **προγραμματισμός**, ενώ τα άτομα που γράφουν και συντάσσουν ένα πρόγραμμα ονομάζονται **προγραμματιστές**.



Οι πύργοι του Ανόι

Ο Θρύλος: Σε κάποιους Ινδούς μοναχούς δόθηκε η δοκιμασία να μετακινήσουν 64 εύθραυστους δίσκους από μία τοποθεσία σε μια άλλη, έναν κάθε φορά, αποφεύγοντας την τοποθέτηση ενός μεγαλύτερου δίσκου πάνω σε έναν μικρότερο. Υπήρχε μόνο μια ακόμα ενδιάμεση τοποθεσία, πέρα από τις δύο, που ένας δίσκος μπορούσε να τοποθετηθεί.

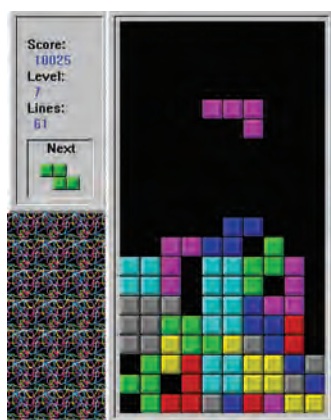
Το παιχνίδι: Υπάρχει ένα παιχνίδι βασισμένο στο μύθο. Έχετε μια μικρή συλλογή από δίσκους και τρεις πασσάλους πάνω στους οποίους μπορείτε να τους τοποθετήσετε (ο κάθε δίσκος έχει στη μέση μία τρύπα ώστε να τοποθετείται στον πασσαλό). Οι δίσκοι είναι όλοι τοποθετημένοι στον αριστερό πασσαλό σε αύξουσα σειρά ανάλογα με το μέγεθός τους. Θα πρέπει να τους μετακινήσεις στο δεξιό πασσαλό χωρίς ποτέ όμως να βάλεις έναν μεγαλύτερο δίσκο πάνω σε έναν μικρότερο. Καταγράψτε τον κατάλληλο αλγόριθμο που να περιγράφει πώς να μεταφέρετε τους δίσκους από τον αριστερό πασσαλό στον δεξιό. (Ο ελάχιστος αριθμός βημάτων του αλγορίθμου είναι: 3 βήματα για 2 δίσκους, 7 βήματα για 3 δίσκους, 15 βήματα για 4 δίσκους και 31 βήματα για 5 δίσκους).



Ο μύθος λέει πως όταν οι μοναχοί καταφέρουν να μετακινήσουν τους 64 δίσκους στη νέα τοποθεσία, τότε ο νασός τους θα καταρρεύσει και θα μετατραπεί σε σκόνη και ακόμα ο κόσμος θα καταστραφεί.

Όλα τα προγράμματα του υπολογιστή αποτελούνται από ένα πλήθος κατάλληλων εντολών, που είναι γραμμένες σε λογική σειρά. Τα παιχνίδια, ο Επεξεργαστής Κειμένου, η Ζωγραφική, το Λειτουργικό Σύστημα αποτελούνται από ένα πλήθος εντολών κατανοητών από τον υπολογιστή (Εικόνα 1.2). Κάθε φορά που χρειαζόμαστε ένα πρόγραμμα, για να εκτελέσουμε μια λειτουργία ή να επιλύσουμε κάποιο πρόβλημα, ένα σύνολο εντολών αποθηκεύονται («φορτώνονται») στη μνήμη του υπολογιστή, για να εκτελεστούν στη συνέχεια πιστά από την Κεντρική Μονάδα Επεξεργασίας. (Δείτε επίσης την Εικόνα 5.2 της Α' Γυμνασίου).

Στο επόμενο κεφάλαιο θα μάθουμε και εμείς να προγραμματίζουμε τον υπολογιστή, ώστε να δημιουργούμε τα δικά μας προγράμματα. Τα προγράμματα που θα αναπτύξουμε μπορεί να είναι απλά στην αρχή, αλλά οι βασικές αρχές του προγραμματισμού είναι παρόμοιες και στα πιο σύνθετα προγράμματα. Με τον καιρό θα διαπιστώσετε ότι μπορείτε να δημιουργείτε όλο και πιο σύνθετα προγράμματα, παιχνίδια, εκπαιδευτικά προγράμματα, ή ιστοσελίδες και να επιλύετε διάφορα προβλήματα με τη βοήθεια του υπολογιστή.



```
void DisplayBlock(SBlock Block)
{
    if (Block.nY < 1) return;
    RECT rcBlock = g_rcBlock;
    rcBlock.left = Block.nColor * BLOCK_DIAMETER;
    rcBlock.right = Block.nColor * BLOCK_DIAMETER + BLOCK_DIAMETER;
    g_pDisplay->Blit( (DWORD)Block.nX * BLOCK_DIAMETER - 2,
                     (DWORD)Block.nY * BLOCK_DIAMETER,
                     g_pSecondarySurface, &rcBlock );
}
```

Εικόνα 1.2. Το γνωστό παιχνίδι ΤΕΤΡΙΣ είναι ένα πρόγραμμα το οποίο περιλαμβάνει μια σειρά εντολών (ένα μικρό υποσύνολο των εντολών του μπορείτε να δείτε στα δεξιά της εικόνας).

1.5 Γλώσσες Προγραμματισμού

Διαβάζοντας τα παραπάνω μπορεί κάποιος να αναρωτηθεί σε ποια γλώσσα μπορούμε να προγραμματίσουμε έναν υπολογιστή. Οι γλώσσες που «καταλαβαίνουν» οι υπολογιστές είναι τεχνητές γλώσσες που ονομάζονται **γλώσσες προγραμματισμού**. Οι γλώσσες προγραμματισμού χρησιμοποιούνται για την επικοινωνία του ανθρώπου με τη μηχανή, όπως οι φυσικές γλώσσες (ελληνική, αγγλική, γαλλική κ.λπ.) χρησιμοποιούνται για την επικοινωνία μεταξύ των ανθρώπων.

Οι γλώσσες προγραμματισμού έχουν κι αυτές το δικό τους λεξιλόγιο και το δικό τους συντακτικό. Αν θέλουμε να προγραμματίσουμε τον υπολογιστή, για να εκτελεί πιστά τις λειτουργίες που του ζητάμε, πρέπει να μάθουμε κάποια γλώσσα προγραμματισμού. Δυστυχώς οι υπολογιστές δεν έχουν σχεδιαστεί, ώστε να καταλαβαίνουν τη γλώσσα που μιλάμε, δηλαδή τη φυσική γλώσσα. Η πρόοδος, όμως, στον τομέα αυτό είναι σημαντική και πιθανόν στο μέλλον να δίνουμε οδηγίες στον υπολογιστή με την ομιλία.

Γλώσσα Μηχανής

Όπως έχει αναφερθεί στη Β' Γυμνασίου, η λειτουργία των υπολογιστών βασίζεται στην αναπαράσταση μόνο δύο ψηφίων, των «0» και «1». Στα πρώτα βήματα της ιστορίας των υπολογιστών οι άνθρωποι, για να επικοινωνήσουν με τον υπολογιστή,

έπρεπε να χρησιμοποιούν μία γλώσσα που είχε ως αλφάβητο το «0» και το «1». Αν ήθελαν λοιπόν να δώσουν μία απλή εντολή στον υπολογιστή, π.χ. να προσθέσει το $3+5$ και να εμφανίσει το αποτέλεσμα, έπρεπε να μετατρέψουν όλη την εντολή σε μία γραμμή από 0 και 1. Η γλώσσα αυτή ονομάστηκε **γλώσσα μηχανής**. Η γλώσσα μηχανής είναι αρκετά δύσκολη για να την μάθει κάποιος, γιατί είναι πολύ διαφορετική από τη φυσική μας γλώσσα (Εικόνα 1.3). Επίσης δεν είναι ενιαία σε όλους τους υπολογιστές, μια και κάθε τύπος υπολογιστή (με διαφορετικό επεξεργαστή) έχει τη δική του γλώσσα μηχανής.

Χαρακτηριστικά Γλωσσών Προγραμματισμού

Με την πάροδο των χρόνων οι γλώσσες προγραμματισμού εξελίχθηκαν, ώστε να μοιάζουν όλο και περισσότερο με τη φυσική μας γλώσσα. Στις μέρες μας υπάρχουν διάφορες γλώσσες προγραμματισμού, που χρησιμοποιούνται για την ανάπτυξη γενικών εφαρμογών, ενώ άλλες είναι πιο εξειδικευμένες και χρησιμοποιούνται για πιο ειδικά επιστημονικά προβλήματα (ανώτερων μαθηματικών, μηχανικής, προσομοίωσης πειραμάτων κ.λπ.) και εξειδικευμένες εφαρμογές (προγραμματισμός ιστοσελίδων, διαχείριση εμπορικών δεδομένων κ.λπ.). Μερικές γνωστές γλώσσες προγραμματισμού είναι η Visual Basic, η Logo, η Pascal, η C++, η Java και άλλες.

Όπως και οι φυσικές γλώσσες, έτσι και κάθε γλώσσα προγραμματισμού έχει ως βασικά χαρακτηριστικά:

- το **αλφάβητο**,
- το **λεξιλόγιο** και
- το **συντακτικό**.

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

Το **λεξιλόγιο** μιας γλώσσας είναι το σύνολο των λέξεων που αναγνωρίζει η γλώσσα και έχουν συγκεκριμένη και μοναδική σημασία. Στις γλώσσες προγραμματισμού το λεξιλόγιο είναι πολύ περιορισμένο (μερικές δεκάδες λέξεις), ώστε να μπορούμε να το μάθουμε εύκολα.

Το **συντακτικό** μιας γλώσσας προγραμματισμού είναι το σύνολο των κανόνων που πρέπει να ακολουθούμε, για να συνδέουμε λέξεις σε προτάσεις. Σε μια γλώσσα προγραμματισμού η σύνδεση λέξεων δημιουργεί ολοκληρωμένες εντολές προς τον υπολογιστή. Αν δεν ακολουθήσουμε αυστηρά το συντακτικό μιας γλώσσας, είναι αδύνατο για τον υπολογιστή να καταλάβει ποια εντολή του δίνουμε.

Για να μάθουμε λοιπόν μία γλώσσα προγραμματισμού, πρέπει να μάθουμε σταδιακά το λεξιλόγιο που χρησιμοποιεί και το συντακτικό που ακολουθεί, ώστε να γράφουμε κατάλληλα τις εντολές. Κάθε εντολή προκαλεί συγκεκριμένες ενέργειες, αν εκτελεστεί από τον υπολογιστή. Για παράδειγμα, στη γλώσσα Logo η εντολή «**ΤΥΠΩΣΕ "Καλημέρα"**» έχει ως αποτέλεσμα την εμφάνιση της λέξης «Καλημέρα» στην οθόνη του υπολογιστή.

Το ολοκληρωμένο προγραμματιστικό περιβάλλον

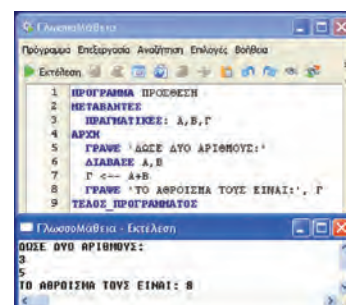
Οι σύγχρονες γλώσσες προγραμματισμού μάς προσφέρουν ένα φιλικό περιβάλλον, έτσι ώστε γρήγορα να αναπτύσσουμε τα προγράμματά μας. Ένα περιβάλλον προγραμματισμού αποτελείται από διάφορα εργαλεία που βοηθάνε τον προγραμματιστή να γράψει και να διορθώσει το πρόγραμμά του.

Τα κύρια εργαλεία είναι:

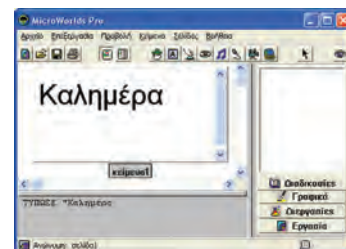
➤ ένας εξειδικευμένος κειμενογράφος, που χρησιμεύει για τη σύνταξη και τη διόρθωση του προγράμματος και

```
00000000
00000001
00000010
00000110
00000000
00100000
```

Εικόνα 1.3. Τμήμα Προγράμματος σε γλώσσα μηχανής



Εικόνα 1.4. Ο κώδικας για την άθροιση δύο αριθμών στο προγραμματιστικό περιβάλλον Γλωσσμάθεια



Εικόνα 1.5. Το αποτέλεσμα της εντολής «Τύπωσε "Καλημέρα"» στο περιβάλλον MWorldsPro

➤ ένα πρόγραμμα-μεταφραστής που μετατρέπει τις οδηγίες μας στη μορφή που τις καταλαβαίνει ο επεξεργαστής, δηλαδή σε μια σειρά από 0 και 1 (Σχήμα 1.3).

Αυτή τη μετατροπή μπορούμε να την παρομοιάσουμε με τη διαδικασία επικοινωνίας μας με ένα κάτοικο της Κίνας. Αν δεν ξέρουμε Κινέζικα και έχουμε έναν Άγγλο μεταφραστή που μιλάει Κινέζικα, μπορούμε να του μιλήσουμε Αγγλικά και αυτός να μεταφράσει αυτό που θέλουμε στα Κινέζικα. Βέβαια μια τέτοια διαδικασία προϋποθέτει ότι ξέρουμε Αγγλικά. Παρόμοια, αν θέλουμε να επικοινωνήσουμε με τον υπολογιστή, πρέπει να μάθουμε μία γλώσσα προγραμματισμού με την οποία μπορεί να γίνει η απαραίτητη μετατροπή των οδηγιών μας σε σειρά από 0 και 1 (γλώσσα μηχανής).

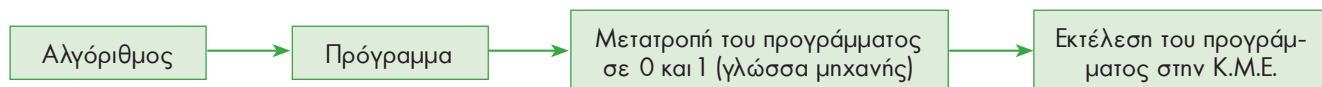
Αν σε κάποια οδηγία έχουμε κάνει λάθος στο αλφάβητο, στο λεξιλόγιο ή στο συντακτικό τότε το πρόγραμμα που μετατρέπει τις οδηγίες μας σε σειρά από 0 και 1 θα μας δώσει ένα κατάλληλο μήνυμα λάθους, ώστε να μας βοηθήσει να διορθώσουμε το λάθος μας. Τα λάθη αυτά ονομάζονται **συντακτικά λάθη**.

Τα προγράμματα που μετατρέπουν τις οδηγίες μας σε 0 και 1 μπορούν να χωριστούν σε δύο κατηγορίες:

- στους μεταγλωττιστές και
- στους διερμυνείς.

Η διαφορά τους είναι ότι οι **μεταγλωττιστές (compilers)** θα ελέγξουν όλο το πρόγραμμα για συντακτικά λάθη και μετά θα το μετατρέψουν όλο σε μια κατάλληλη σειρά από 0 και 1, ώστε να μπορεί να εκτελεστεί από τον επεξεργαστή του υπολογιστή.

Αντίθετα οι **διερμυνείς (interpreters)** ελέγχουν μία οδηγία κάθε φορά, την εκτελούν και μετά ελέγχουν την επόμενη οδηγία. Η γλώσσα προγραμματισμού Logo, που θα δούμε στο επόμενο κεφάλαιο, χρησιμοποιεί διερμυνέα.

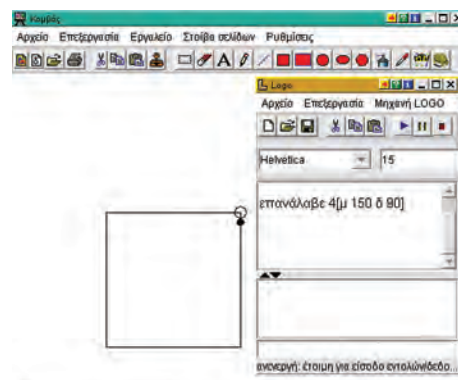


Σχήμα 1.3. Στάδια για την εκτέλεση ενός αλγορίθμου από την Κ.Μ.Ε. του υπολογιστή

Δεν πρέπει να ξεχνάμε ότι ο υπολογιστής εκτελεί πιστά, όποιες συντακτικά ορθές εντολές και αν του δώσουμε. Αν το αποτέλεσμα, που τελικά προκύπτει από την εκτέλεση του προγράμματος, δεν είναι το αναμενόμενο, τότε το πρόβλημα δε βρίσκεται στον τρόπο εκτέλεσης, αλλά στον αλγόριθμο που κατασκευάσαμε για τη λύση του προβλήματός μας. Στην περίπτωση αυτή λέμε ότι έχουμε κάνει ένα **λογικό λάθος** και πρέπει να ελέγξουμε ένα προς ένα τα βήματα-εντολές του αλγορίθμου μας, ώστε να διαπιστώσουμε, αν δίνουμε τις κατάλληλες εντολές με τη σωστή σειρά.

Ένα δεύτερο σημείο που πρέπει να γνωρίζουμε, όταν προγραμματίζουμε, είναι ότι για τον υπολογιστή **τίποτα δεν είναι αυτονόητο**. Ενώ εμείς οι άνθρωποι έχουμε την ικανότητα να συμπληρώνουμε τις οδηγίες κάποιου με τη λογική και την εμπειρία μας, ο υπολογιστής χρειάζεται να περιγράψουμε με μεγάλη ακρίβεια τις εντολές μας στον υπολογιστή, για να τις εκτελέσει. Αν, για παράδειγμα, του δώσουμε μία εντολή να υπολογίσει ένα άθροισμα, δεν είναι αυτονόητο ότι θα μας εμφανίσει και το αποτέλεσμα. Αν φαίνεται ότι οι υπολογιστές επιλύουν πολύ «έξυπνα» διάφορα προβλήματα, είναι, γιατί κάποιοι άνθρωποι τους προγραμμάτισαν γι' αυτό και όχι γιατί οι μηχανές είναι «έξυπνες». Για να φτιάξουμε λοιπόν ένα καλό πρόγραμμα, πρέπει πρώτα να έχουμε σχεδιάσει έναν καλό αλγόριθμο. Ο ρόλος του αλγορίθμου είναι θεμελιώδης.

Στο κεφάλαιο που ακολουθεί θα ασχοληθούμε με μία εκπαιδευτική γλώσσα με αρκετές δυνατότητες: τη **Logo**. Με τη γλώσσα Logo έχουμε τη δυνατότητα να μάθουμε πολύ γρήγορα πώς να δίνουμε εντολές στον υπολογιστή και να φτιάχνουμε δικά μας προγράμματα. Υπάρχουν πολλές εκδόσεις της γλώσσας Logo. Καθεμία μας προσφέρει ένα φιλικό περιβάλλον προγραμματισμού, για να γράφουμε και να δοκιμάζουμε τα προγράμματά μας. Τη «Berkeley Logo», το «Χελωνόκοσμος» (Εικόνα 1.4) και τη «MSWLogo» μπορείτε να τις βρείτε δωρεάν στο Διαδίκτυο, ενώ τη «Multi-Logo» μπορείτε να τη βρείτε στο λογισμικό Πληροφορικής Γυμνασίου (Παιδαγωγικό Ινστιτούτο 2000).



Εικόνα 1.6. Το περιβάλλον «Χελωνόκοσμος» (Παιδαγωγική Σχεδίαση: Εργαστήριο Εκπαιδευτικής Τεχνολογίας, Φ.Π.Ψ.).



ΕΡΩΤΗΣΕΙΣ

1. Γιατί πρέπει να κατανοούμε καλά ένα πρόβλημα, πριν να το επιλύσουμε;
2. Ποιες διαδικασίες μας βοηθούν στην κατανόηση ενός προβλήματος;
3. Τι είναι ένας αλγόριθμος;
4. Ποιες είναι οι βασικές ιδιότητες ενός Αλγορίθμου;
5. Τι είναι πρόγραμμα;
6. Ποιο είναι το αλφάβητο της γλώσσας μηχανής του υπολογιστή;
7. Ποια είναι τα βασικά χαρακτηριστικά μιας γλώσσας προγραμματισμού;
8. Ποια είναι τα στάδια για την εκτέλεση ενός αλγορίθμου από την Κ.Μ.Ε;