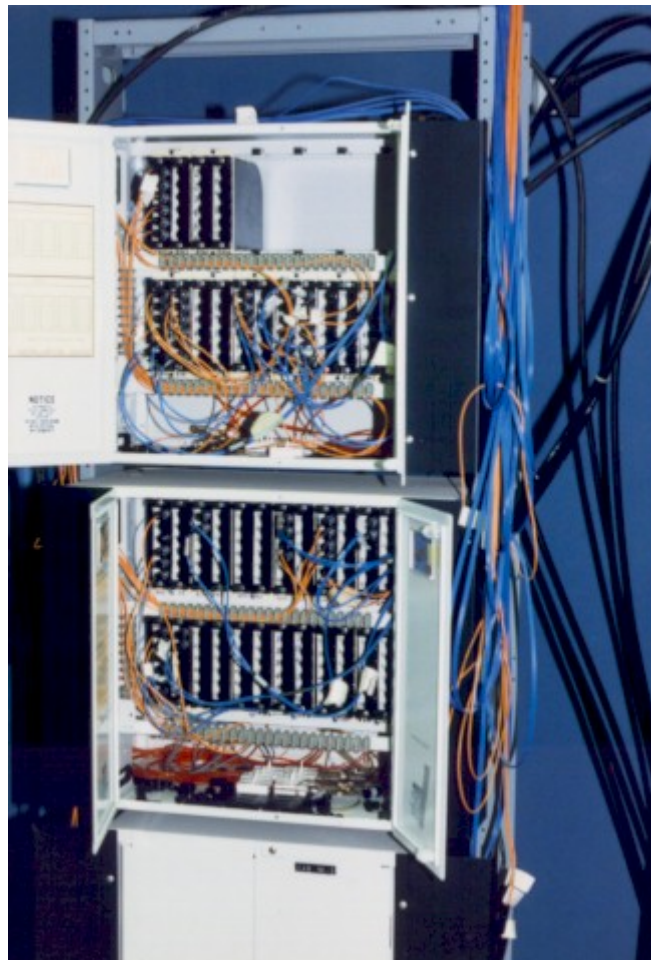


# ΚΕΦΑΛΑΙΑ VII, VIII, IX

## ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ







## Βασικά στοιχεία προγραμματισμού

Εκατοντάδες γλώσσες προγραμματισμού χρησιμοποιούνται όπως αναφέρθηκε σήμερα για την επίλυση των προβλημάτων με τον υπολογιστή, τη δημιουργία σωστών προγραμμάτων. Η επιλογή της κατάλληλης γλώσσας δεν είναι εύκολη και εξαρτάται από το είδος του προγράμματος, το διαθέσιμο εξοπλισμό και σαφώς τις γνώσεις και τις ιδιαίτερες προτιμήσεις του προγραμματιστή. Συχνά το ίδιο πρόβλημα μπορεί να λυθεί εξίσου ικανοποιητικά με πολλές γλώσσες προγραμματισμού.

Η γλώσσα προγραμματισμού που θα χρησιμοποιήσουμε ονομάζεται **ΓΛΩΣΣΑ** και περιέχει τα χαρακτηριστικά, τις δομές και τις εντολές που περιέχονται σε διάφορες σύγχρονες γλώσσες προγραμματισμού όπως η Pascal, η C, η Java και άλλες.

### Το αλφάβητο της γλώσσας

Το αλφάβητο της ΓΛΩΣΣΑΣ αποτελείται από τα γράμματα του ελληνικού και λατινικού αλφαβήτου καθώς και από ειδικά σύμβολα, που χρησιμοποιούνται για προκαθορισμένες ενέργειες.

Κεφαλαία ελληνικού αλφαβήτου (Α-Ω)

Πεζά ελληνικού αλφαβήτου (α-ω)

Κεφαλαία λατινικού αλφαβήτου (Α-Z)

Πεζά λατινικού αλφαβήτου (a-z)

Ψηφία 0-9

Ειδικοί χαρακτήρες + - \* / = ( ) . , ' ! \_ και ο κενός χαρακτήρας.

### Ονόματα

Κάθε πρόγραμμα καθώς και τα δεδομένα που χρησιμοποιεί (συμβολικές σταθερές και μεταβλητές) έχουν ένα όνομα με το οποίο αναφερόμαστε σε αυτά. Τα ονόματα αυτά μπορούν να αποτελούνται από γράμματα του ελληνικού ή του λατινικού αλφαβήτου, πεζά ή κεφαλαία, από ψηφία και από τον ειδικό χαρακτήρα κάτω παύλα (underscore) '\_'. Πρέπει όμως υποχρεωτικά να ξεκινάνε με κάποιο γράμμα. Επειδή μερικές λέξεις χρησιμοποιούνται από την ίδια τη ΓΛΩΣΣΑ για συγκεκριμένους λόγους, όπως οι λέξεις ΠΡΟΓΡΑΜΜΑ, ΑΚΕΡΑΙΟΣ, ΑΝ, κ.λ.π., αυτές οι λέξεις δεν μπορούν να χρησιμοποιηθούν ως ονόματα. Οι λέξεις αυτές αποκαλούνται δεσμευμένες.

Παραδείγματα αποδεκτών ονομάτων: α1, temp, Μέσος\_Όρος.

Παραδείγματα μη αποδεκτών ονομάτων: 1α, temp\$, Μέση Τιμή.

### Τύποι δεδομένων

Οι υπολογιστές επεξεργάζονται δεδομένα διαφορετικών τύπων ακόμα και για τους αριθμούς. Δεδομένα διαφορετικών τύπων αποθηκεύονται και επεξεργάζονται με διαφορετικό τρόπο από τον υπολογιστή.

▶ **Ακέραιος Τύπος.** Παράδειγμα: -3, 45, 0.

▶ **Πραγματικός τύπος.** Παράδειγμα: 3.123, -4.384653, 0.45

▶ **Χαρακτήρας.** Περιλαμβάνει σειρές από έναν ή περισσότερους χαρακτήρες που μπορούν να δοθούν από ένα πληκτρολόγιο. Συχνά ονομάζονται αλφαριθμητικές. Περικλείονται υποχρεωτικά σε απλά εισαγωγικά '!'. Παράδειγμα: 'K', 'Ολυμπιακός', '#\$@!', 'GATE 7'.

▶ **Λογικός.** Δέχεται μόνο δύο τιμές ΑΛΗΘΗΣ και ΨΕΥΔΗΣ. Οι τιμές αντιπροσωπεύουν αληθείς ή ψευδείς συνθήκες.



## ΣΤΑΘΕΡΕΣ

Οι σταθερές είναι προκαθορισμένες τιμές που δεν μεταβάλλονται κατά τη διάρκεια εκτέλεσης του προγράμματος. Οι σταθερές είναι αντιστοιχού τύπου δεδομένων, δηλαδή ακέραιες, πραγματικές, αλφαριθμητικές ή λογικές.

### Συμβολικές Σταθερές.

Η γλώσσα επιτρέπει την αντιστοίχιση σταθερών τιμών με ονόματα, εφ' όσον έχουν δηλωθεί στην αρχή του προγράμματος (στο τμήμα δήλωσης σταθερών)

<i>Σύνταξη:</i>  <b>ΣΤΑΘΕΡΕΣ</b> Όνομα1 = σταθερή_τιμή1 Όνομα2 = σταθερή_τιμή2 . . . ΌνομαN = σταθερή_τιμήN	<i>Παράδειγμα:</i>  <b>ΣΤΑΘΕΡΕΣ</b> pi = 3.14159 e = 2.71828 ΦΠΑ = 0.18
-------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------

## Μεταβλητές

Μια μεταβλητή παριστάνει μία ποσότητα που η τιμή της μπορεί να μεταβάλλεται. Οι μεταβλητές που χρησιμοποιούνται σε ένα πρόγραμμα, αντιστοιχούνται από το μεταγλωττιστή σε συγκεκριμένες θέσεις μνήμης του υπολογιστή. Η τιμή της μεταβλητής είναι η τιμή που βρίσκεται στην αντίστοιχη θέση μνήμης και όπως αναφέρθηκε μπορεί να μεταβάλλεται κατά τη διάρκεια της εκτέλεσης του προγράμματος. Το όνομα της μεταβλητής καλείται αναγνωριστικό. Η ΓΛΩΣΣΑ επιτρέπει τη χρήση μεταβλητών των 4 τύπων που αναφέραμε παραπάνω. Η δήλωση του τύπου κάθε μεταβλητής που θα χρησιμοποιήσουμε στο πρόγραμμά μας γίνεται στην αρχή του προγράμματος στο τμήμα δήλωσης μεταβλητών.

<i>Σύνταξη:</i>  <b>ΜΕΤΑΒΛΗΤΕΣ</b> Τύπος_1: Λίστα_μεταβλητών1 Τύπος_2: Λίστα_μεταβλητών2 . . . Τύπος_N: Λίστα_μεταβλητώνN	<i>Παράδειγμα:</i>  <b>ΜΕΤΑΒΛΗΤΕΣ</b> <b>ΠΡΑΓΜΑΤΙΚΕΣ:</b> A, B <b>ΑΚΕΡΑΙΕΣ:</b> i, j <b>ΧΑΡΑΚΤΗΡΕΣ:</b> όνομα, απάντηση <b>ΛΟΓΙΚΕΣ:</b> έλεγχος
---------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Αριθμητικοί τελεστές**

Οι αριθμητικοί τελεστές που χρησιμοποιεί η ΓΛΩΣΣΑ για τις διάφορες πράξεις είναι:

Αριθμητικός τελεστής	Πράξη
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
^	Ύψωση σε δύναμη
DIV	Ακέραια διαίρεση
MOD	Υπόλοιπο ακέραιας διαίρεσης

**Συγκριτικοί τελεστές**

Στις λογικές εκφράσεις γίνεται σύγκριση της τιμής μιας έκφρασης, που βρίσκεται αριστερά από το συγκριτικό τελεστή με την τιμή μιας άλλης έκφρασης που βρίσκεται δεξιά. Οι χρησιμοποιούμενοι συγκριτικοί τελεστές παρουσιάζονται στον παρακάτω πίνακα.

Συγκριτικός τελεστής	Ελεγχόμενη σχέση
=	Ισότητα
<>	Ανισότητα (διάφορο)
>	Μεγαλύτερο από
>=	Μεγαλύτερο ή ίσο
<	Μικρότερο από
<=	Μικρότερο ή ίσο

**Λογικοί Τελεστές**

Μεταξύ δύο λογικών εκφράσεων μπορούν να γίνουν και πράξεις. Συγκεκριμένα μπορούν να χρησιμοποιηθούν οι τελεστές ΚΑΙ, Η, ΟΧΙ που έχουν επεξηγηθεί σε προηγούμενα κεφάλαια.

**Συναρτήσεις**

Πολλές γνωστές συναρτήσεις από τα μαθηματικά χρησιμοποιούνται συχνά και γι αυτό περιέχονται στη ΓΛΩΣΣΑ.

HM(x)	Υπολογισμός ημίτονου
SYN(x)	Υπολογισμός συνημίτονου
EΦ(x)	Υπολογισμός εφαπτομένης
T_P(x)	Υπολογισμός τετραγωνικής ρίζας
ΛΟΓ(x)	Υπολογισμός φυσικού λογαρίθμου
E(x)	Υπολογισμός του $e^x$
A_M(x)	Υπολογισμός ακέραιου μέρους
A_T(x)	Υπολογισμός απόλυτης τιμής

**Αριθμητικές εκφράσεις.**

Όταν μια τιμή προκύπτει από υπολογισμό, τότε αναφερόμαστε σε εκφράσεις (expressions). Για τη σύνταξη μιας έκφρασης χρησιμοποιούνται αριθμητικές σταθερές, μεταβλητές, συναρτήσεις, αριθμητικοί



τελεστές και παρενθέσεις. Οι πράξεις εκτελούνται με βάση τη γνωστή ιεραρχία (παρενθέσεις, δυνάμεις, πολλαπλασιασμοί-διαιρέσεις και προσθέσεις αφαιρέσεις).

### Εντολή εκχώρησης τιμής.

Η εντολή αυτή χρησιμοποιείται για την απόδοση τιμών στις μεταβλητές κατά τη διάρκεια του προγράμματος. Μια εντολή εκχώρησης σε καμιά περίπτωση δεν πρέπει να εκλαμβάνεται σαν εξίσωση. Είναι μια εντολή που καταχωρεί τη τιμή της έκφρασης στο δεξιό μέλος στη μεταβλητή που βρίσκεται στο αριστερό μέλος. Για το λόγο αυτό η γλώσσα χρησιμοποιεί το σύμβολο  $\leftarrow$  αντί του ίσον (=). Πρέπει να σημειωθεί βέβαια ότι οι διάφορες γλώσσες προγραμματισμού χρησιμοποιούν διαφορετικά σύμβολα (π.χ. η Pascal το := ενώ η C το =).

### Εντολές Εισόδου-Εξόδου

Για την εισαγωγή δεδομένων από το πληκτρολόγιο η γλώσσα υποστηρίζει την εντολή ΔΙΑΒΑΣΕ, ενώ για την εμφάνιση αποτελεσμάτων στην οθόνη την εντολή ΓΡΑΨΕ.

Η εντολή ΔΙΑΒΑΣΕ διακόπτει την εκτέλεση του αλγορίθμου και περιμένει την εισαγωγή τιμών από το πληκτρολόγιο που θα καταχωρηθούν στις μεταβλητές που βρίσκονται δεξιά της. Η ολοκλήρωση της εισαγωγής μιας τιμής γίνεται με το πάτημα του κουμπιού enter. Μετά το τελευταίο enter η ροή του προγράμματος συνεχίζεται από την επόμενη εντολή.

Η εντολή ΓΡΑΨΕ έχει ως αποτέλεσμα την εμφάνιση τιμών στη μονάδα εξόδου (εκτυπωτή ή οθόνη). Η λίστα των στοιχείων μπορεί να περιέχει σταθερές τιμές ή και ονόματα μεταβλητών. Κατά την εκτέλεση του προγράμματος η εντολή προκαλεί την εμφάνιση των σταθερών τιμών στην οθόνη. Αν μέσα στη λίστα στοιχείων υπάρχει μια μεταβλητή η εντολή εμφανίζει την τιμή της μεταβλητής στην οθόνη.

<i>Σύνταξη</i>	<i>Σύνταξη</i>
<b>ΔΙΑΒΑΣΕ</b> λίστα-μεταβλητών	<b>ΓΡΑΨΕ</b> λίστα-στοιχείων
<i>Παράδειγμα</i>	<i>Παράδειγμα</i>
<b>ΔΙΑΒΑΣΕ</b> α, τιμή	<b>ΓΡΑΨΕ</b> α, τιμή

**ΔΟΜΗ ΠΡΟΓΡΑΜΜΑΤΟΣ**

Ένα πρόγραμμα έχει 3 τμήματα. Το πρώτο ονομάζεται επικεφαλίδα και το δεύτερο τμήμα δηλώσεων όπου δηλώνονται οι μεταβλητές που θα χρησιμοποιήσουμε στο πρόγραμμα και το τρίτο λέγεται τμήμα εντολών. Σχόλια μπορούν να γραφούν με τη χρήση του χαρακτήρα (!). Οτιδήποτε υπάρχει μετά από ένα θαυμαστικό (μέχρι το τέλος της γραμμής) αγνοείται από τη ΓΛΩΣΣΑ. Η δομή ενός προγράμματος φαίνεται στο παρακάτω σχήμα.



**Εντολές Επιλογής**

Έχουμε ήδη, στο κεφάλαιο 3, εξηγήσει πότε και με ποιο τρόπο χρησιμοποιούμε τις δομές επιλογής. Παρακάτω παρουσιάζονται οι τρεις μορφές δομών επιλογής.

1 <sup>η</sup> Μορφή	2 <sup>η</sup> Μορφή	3 <sup>η</sup> Μορφή
<p><i>Σύνταξη</i></p> <p><b>ΑΝ</b> συνθήκη <b>TOTE</b>            εντολή-1            εντολή-2            .....            εντολή-ν  <b>ΤΕΛΟΣ_ΑΝ</b></p> <p><i>Παράδειγμα</i>  <b>ΑΝ</b> τιμή&gt;0 <b>TOTE</b>  <b>ΓΡΑΨΕ</b> "Ο αριθμός είναι            θετικός"            τιμή ← -τιμή  <b>ΤΕΛΟΣ_ΑΝ</b></p>	<p><i>Σύνταξη</i></p> <p><b>ΑΝ</b> συνθήκη <b>TOTE</b>            εντολή-1            εντολή-2            .....            εντολή-ν  <b>ΑΛΛΙΩΣ</b>            εντολή-1            εντολή-2            .....            εντολή-μ  <b>ΤΕΛΟΣ_ΑΝ</b></p> <p><i>Παράδειγμα</i>  <b>ΑΝ</b> τιμή&gt;=0 <b>TOTE</b>            απόλυτο ← τιμή  <b>ΓΡΑΨΕ</b> "Ο αριθμός είναι            θετικός ή μηδέν"  <b>ΑΛΛΙΩΣ</b>            απόλυτο ← -τιμή  <b>ΓΡΑΨΕ</b> "Ο αριθμός είναι            αρνητικός"  <b>ΤΕΛΟΣ_ΑΝ</b></p>	<p><i>Σύνταξη</i></p> <p><b>ΑΝ</b> συνθήκη-1 <b>TOTE</b>            εντολή-1            .....            εντολή-ν  <b>ΑΛΛΙΩΣ_ΑΝ</b> συνθηκή-2            εντολή-1            .....            εντολή-ν            .....  <b>ΑΛΛΙΩΣ</b>            εντολή-1            .....            εντολή-ν  <b>ΤΕΛΟΣ_ΑΝ</b></p> <p><i>Παράδειγμα</i>  <b>ΑΝ</b> τιμή&gt;0 <b>TOTE</b>            απόλυτο ← τιμή  <b>ΓΡΑΨΕ</b> "Ο αριθμός είναι            θετικός"  <b>ΑΛΛΙΩΣ_ΑΝ</b> τιμή&lt;0 <b>TOTE</b>            απόλυτο ← -τιμή  <b>ΓΡΑΨΕ</b> "Ο αριθμός είναι            αρνητικός"  <b>ΑΛΛΙΩΣ</b>            απόλυτο ← 0  <b>ΓΡΑΨΕ</b> "Ο αριθμός είναι            το μηδέν"  <b>ΤΕΛΟΣ_ΑΝ</b></p>



**Εντολή ΕΠΙΛΕΞΕ.**

Αν οι εναλλακτικές περιπτώσεις επιλογής είναι πολλές, μπορεί να χρησιμοποιηθεί η εντολή ΕΠΙΛΕΞΕ, η γενική μορφή της οποίας είναι:

*Σύνταξη*

**ΕΠΙΛΕΞΕ** έκφραση  
**ΠΕΡΙΠΤΩΣΗ** λίστα\_τιμών\_1  
 εντολές\_1  
 .....  
**ΠΕΡΙΠΤΩΣΗ** λίστα\_τιμών\_v  
 εντολή-v  
**ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ**  
 εντολές\_αλλιώς  
**ΤΕΛΟΣ\_ΕΠΙΛΟΓΩΝ**

*Παράδειγμα*

**ΕΠΙΛΕΞΕ** αριθμός  
**ΠΕΡΙΠΤΩΣΗ** 1, 3, 5, 7, 9  
**ΓΡΑΨΕ** "ο αριθμός είναι περιττός"  
**ΠΕΡΙΠΤΩΣΗ** 2, 4, 6, 8, 10  
**ΓΡΑΨΕ** "ο αριθμός είναι άρτιος"  
**ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ**  
**ΓΡΑΨΕ** "ο αριθμός είναι μεγαλύτερος του 10 ή αρνητικός"  
**ΤΕΛΟΣ\_ΕΠΙΛΟΓΩΝ**

**Εντολές επανάληψης**

Υπάρχουν 3 δομές επανάληψης όπως έχει διατυπωθεί στο κεφάλαιο 3. Η ΓΛΩΣΣΑ υποστηρίζει και τις τρεις, Παρακάτω δίνεται η σύνταξη κάθε μιας. Η μεταβλητή στην επανάληψη της μορφής ΓΙΑ ΑΠΟ ΜΕΧΡΙ πρέπει να είναι ακέραιου τύπου.

<b>ΟΣΟ ... ΕΠΑΝΕΛΑΒΕ</b>	<b>ΜΕΧΡΙΣ_ΟΤΟΥ</b>	<b>ΓΙΑ ... ΑΠΟ ... ΜΕΧΡΙ</b>
<i>Σύνταξη</i> <b>ΟΣΟ</b> συνθήκη <b>ΕΠΑΝΕΛΑΒΕ</b> εντολή-1 ..... εντολή-v <b>ΤΕΛΟΣ_ΑΝ</b>	<i>Σύνταξη</i> <b>ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ</b> εντολή-1 ..... εντολή-v <b>ΜΕΧΡΙΣ_ΟΤΟΥ</b> συνθήκη	<i>Σύνταξη</i> <b>ΓΙΑ</b> μεταβλητή <b>ΑΠΟ</b> τιμή1 <b>ΜΕΧΡΙ</b> τιμή2 <b>ΜΕ_ΒΗΜΑ</b> τιμή3 εντολή-1 ..... εντολή-v <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b>
<i>Παράδειγμα</i> Σ ← 0 <b>ΟΣΟ</b> A < 1000 <b>ΤΟΤΕ</b> <b>ΔΙΑΒΑΣΕ</b> A Σ ← Σ + A <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b>	<i>Παράδειγμα</i> Σ ← 0 <b>ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ</b> <b>ΔΙΑΒΑΣΕ</b> A Σ ← Σ + A <b>ΜΕΧΡΙΣ_ΟΤΟΥ</b> A >= 1000	<i>Παράδειγμα</i> Σ ← 0 <b>ΓΙΑ</b> i <b>ΑΠΟ</b> 1 <b>ΜΕΧΡΙ</b> 100 <b>ΔΙΑΒΑΣΕ</b> A Σ ← Σ + A <b>ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ</b>



## ΠΙΝΑΚΕΣ

Στο κεφάλαιο 3 είδαμε τις βασικές εφαρμογές των πινάκων. Ένας πίνακας μπορεί να περιέχει στοιχεία του ίδιου τύπου μόνο. Ένας πραγματικός πίνακας 50 θέσεων με το όνομα Πωλήσεις δηλώνεται ως εξής:

ΠΡΑΓΜΑΤΙΚΕΣ: Πωλήσεις[50]

Η χρήση πινάκων είναι ένας βολικός τρόπος για την διαχείριση πολλών δεδομένων του ίδιου τύπου, αλλά συχνά η χρήση πίνακα είναι περιττή και επιζήμια στην ανάπτυξη του προγράμματος. Υπάρχουν, λοιπόν, και μερικά μειονεκτήματα από τη χρήση πινάκων.

- ▶ **Οι πίνακες απαιτούν μνήμη.** Κάθε πίνακας δεσμεύει από την αρχή του προγράμματος πολλές θέσεις μνήμης. Σε ένα πολύ μεγάλο πρόγραμμα η άσκοπη χρήση πινάκων μπορεί να οδηγήσει σε αδυναμία εκτέλεσης του προγράμματος.
- ▶ **Οι πίνακες περιορίζουν τις δυνατότητες του προγράμματος.** Οι πίνακες είναι στατικές δομές και το μέγεθός τους παραμένει σταθερό κατά την εκτέλεση του προγράμματος.

Στο κεφάλαιο 3 είδαμε πω υλοποιούνται οι πιο βασικές λειτουργίες σε ένα πίνακα (μονοδιάστατο ή δισδιάστατο). Υπάρχουν μερικές ακόμα βασικές λειτουργίες που δεν θα εξετάσουμε εξονυχιστικά.

- ▶ **Ταξινόμηση.** Όπως ξέρουμε η πιο αργή μέθοδος (αλλά συγχρόνως και η απλούστερη) είναι η φυσαλίδα. Υπάρχουν όμως πολλές μέθοδοι που είναι αρκετά γρηγορότερες.
- ▶ **Αναζήτηση.** Στο κεφάλαιο 4 είδαμε την σειριακή αναζήτηση στοιχείου σε ένα πίνακα. Υπάρχει και μια μέθοδος που είναι αποδοτικότερη αλλά χρησιμοποιείται μόνο σε ταξινομημένους πίνακες. Αυτή είναι η δυαδική αναζήτηση.
- ▶ **Συγχώνευση δύο πινάκων.** Σκοπός της συγχώνευσης είναι, από δύο ή περισσότερους ταξινομημένους πίνακες, η δημιουργία ενός μόνο πίνακα που είναι επίσης ταξινομημένος.