



Επιστημονικές Εκδόσεις
ΤΖΙΟΛΑ

ΚΕΦΑΛΑΙΟ 8

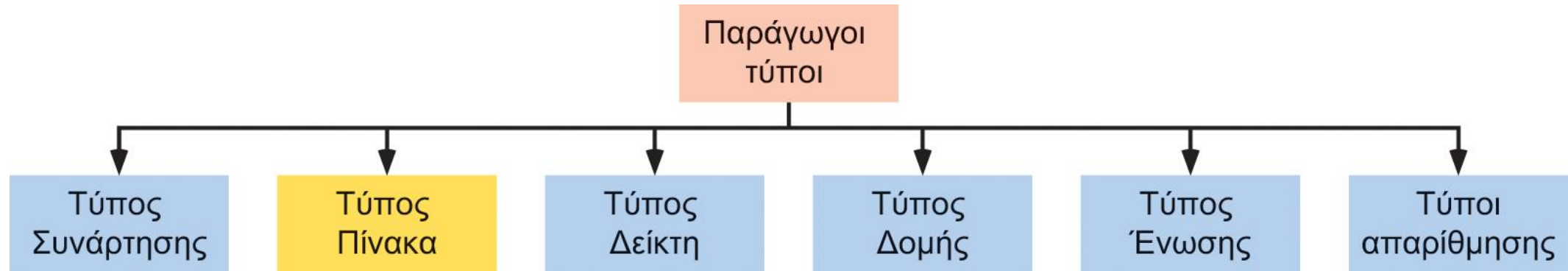
Πίνακες

Πρωτότυπο έργο: Computer Science: A Structured Programming Approach in C, 4th ed., H. Afyouni, B.A.Forouzan. Copyright 2000, 2007, 2023 Cengage Learning, Inc.
Αποκλειστικότητα για την ελληνική γλώσσα: Εκδόσεις ΤΖΙΟΛΑ. Copyright 2024.
Με επιφύλαξη παντός νόμιμου δικαιώματος.

ΘΕΜΑΤΑ ΚΕΦΑΛΑΙΟΥ

- Βασικές έννοιες και χρήσεις των πινάκων
- Δήλωση και ορισμός ενός πίνακα στη C
- Πέρασμα πινάκων και στοιχείων πινάκων σε συναρτήσεις
- Υλοποίηση πινάκων στη C χρησιμοποιώντας αριθμητικές συναρτήσεις
- Περιγραφή τριών κλασικών αλγορίθμων ταξινόμησης: ταξινόμηση με επιλογή, ταξινόμηση με παρεμβολή και ταξινόμηση φουσαλίδας
- Περιγραφή δύο κλασικών αλγορίθμων αναζήτησης: σειριακή (γραμμική) και δυαδική αναζήτηση
- Χρήση διδιάστατων και πολυδιάστατων πινάκων για την αποθήκευση δεδομένων

Βασικές έννοιες



ΣΧΗΜΑ 8-1 Παράγωγοι τύποι δεδομένων.

8.1

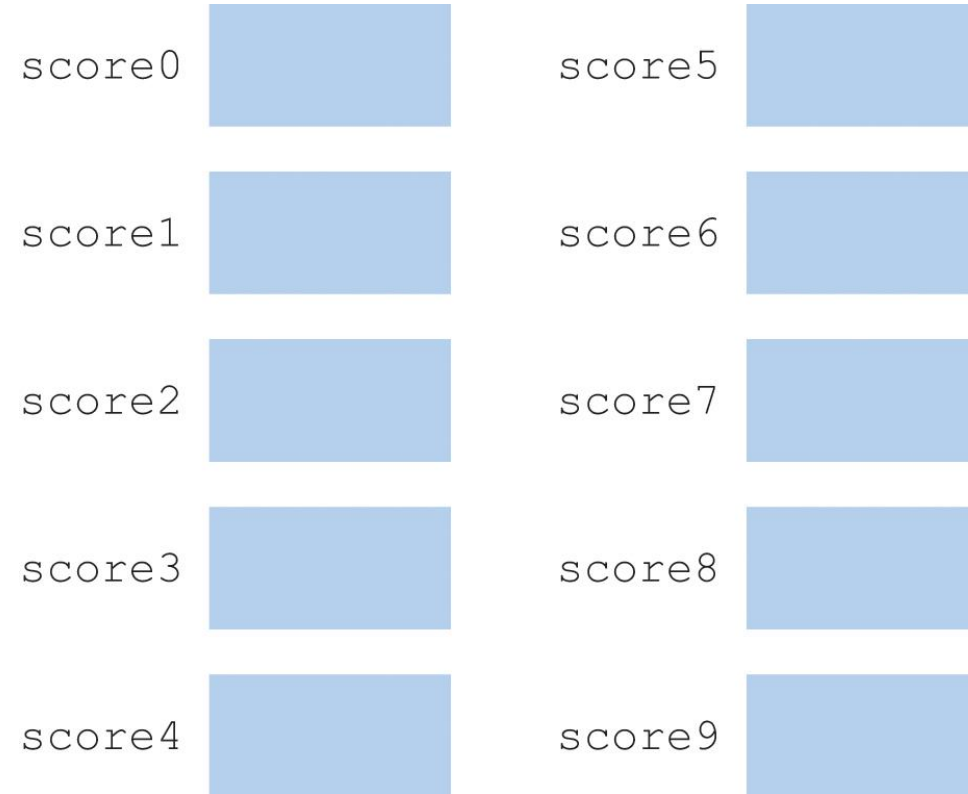
Βασικές έννοιες

Βασικές έννοιες

- Υποθέστε ότι έχουμε ένα πρόβλημα το οποίο απαιτεί να διαβάσουμε, να επεξεργαστούμε και να εκτυπώσουμε 10 βαθμολογίες, δηλαδή 10 ακέραιους αριθμούς
 - Επιπλέον, πρέπει να διατηρούμε τους υπό επεξεργασία αριθμούς στη μνήμη, καθ' όλη τη διάρκεια του προγράμματος
 - Ως πρώτη προσέγγιση, θα μπορούσαμε να δηλώσουμε/ορίσουμε 10 μεταβλητές, κάθε μία με διαφορετικό όνομα

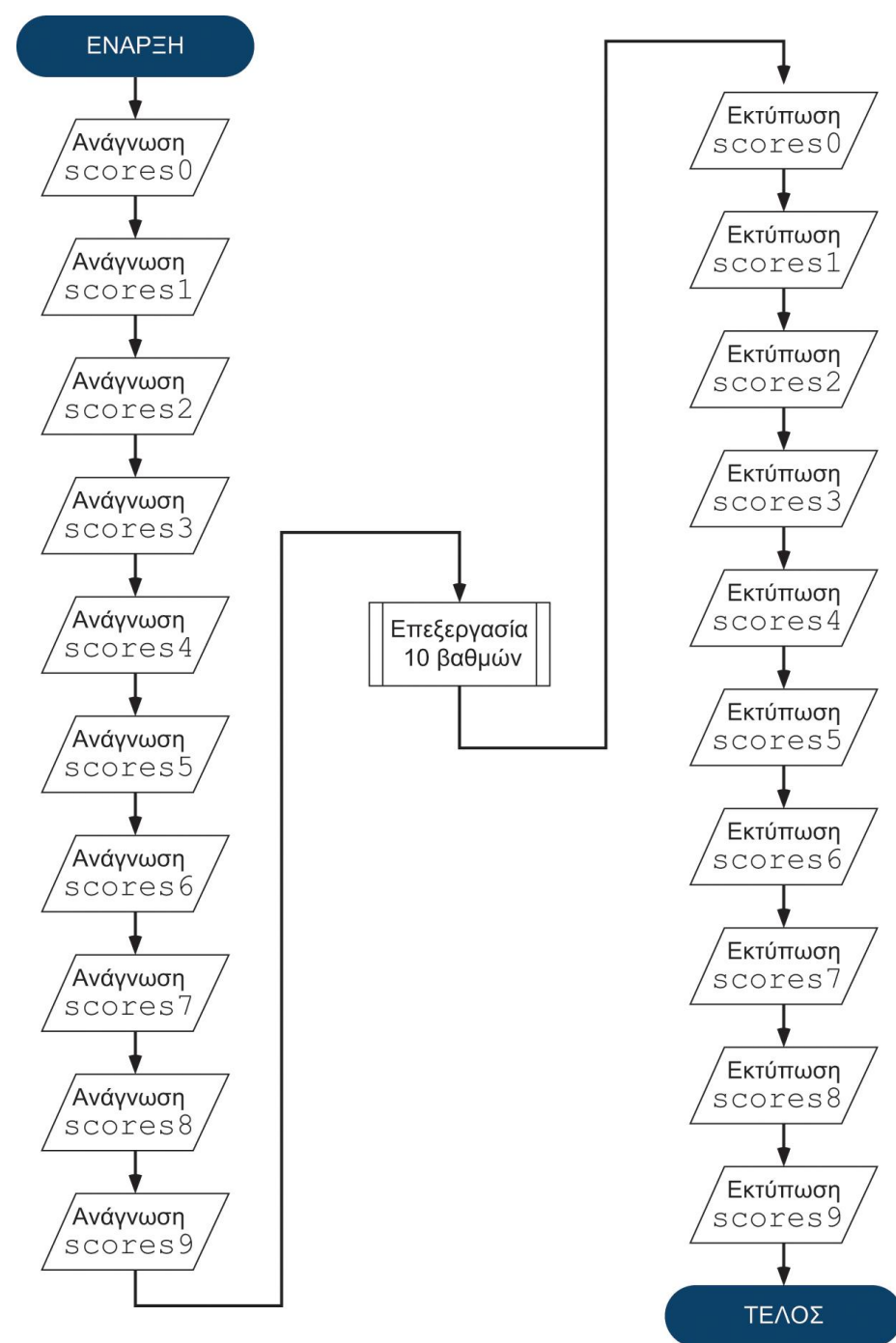
Βασικές έννοιες

- Ως πρώτη προσέγγιση, θα μπορούσαμε να δηλώσουμε/ορίσουμε 10 μεταβλητές, κάθε μία με διαφορετικό όνομα



ΣΧΗΜΑ 8-2 Δέκα μεταβλητές στη μνήμη.

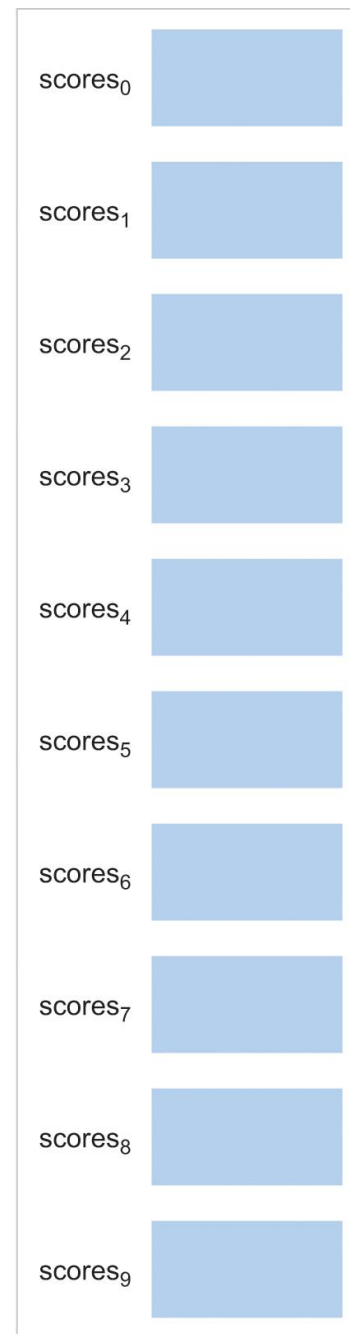
ΣΧΗΜΑ 8-3 Επεξεργασία με την προσέγγιση των 10 μεταβλητών.



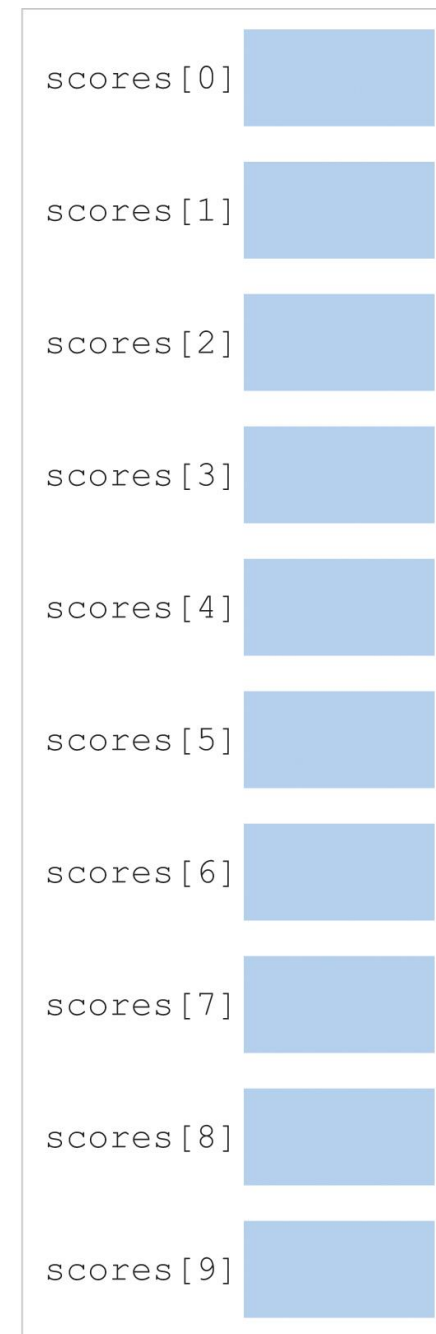
Βασικές έννοιες

- Ένας πίνακας (array) είναι μία συλλογή στοιχείων του ίδιου τύπου δεδομένων

ΣΧΗΜΑ 8-4 Συμβολισμοί για την αναφορά στα μεμονωμένα στοιχεία ενός πίνακα.

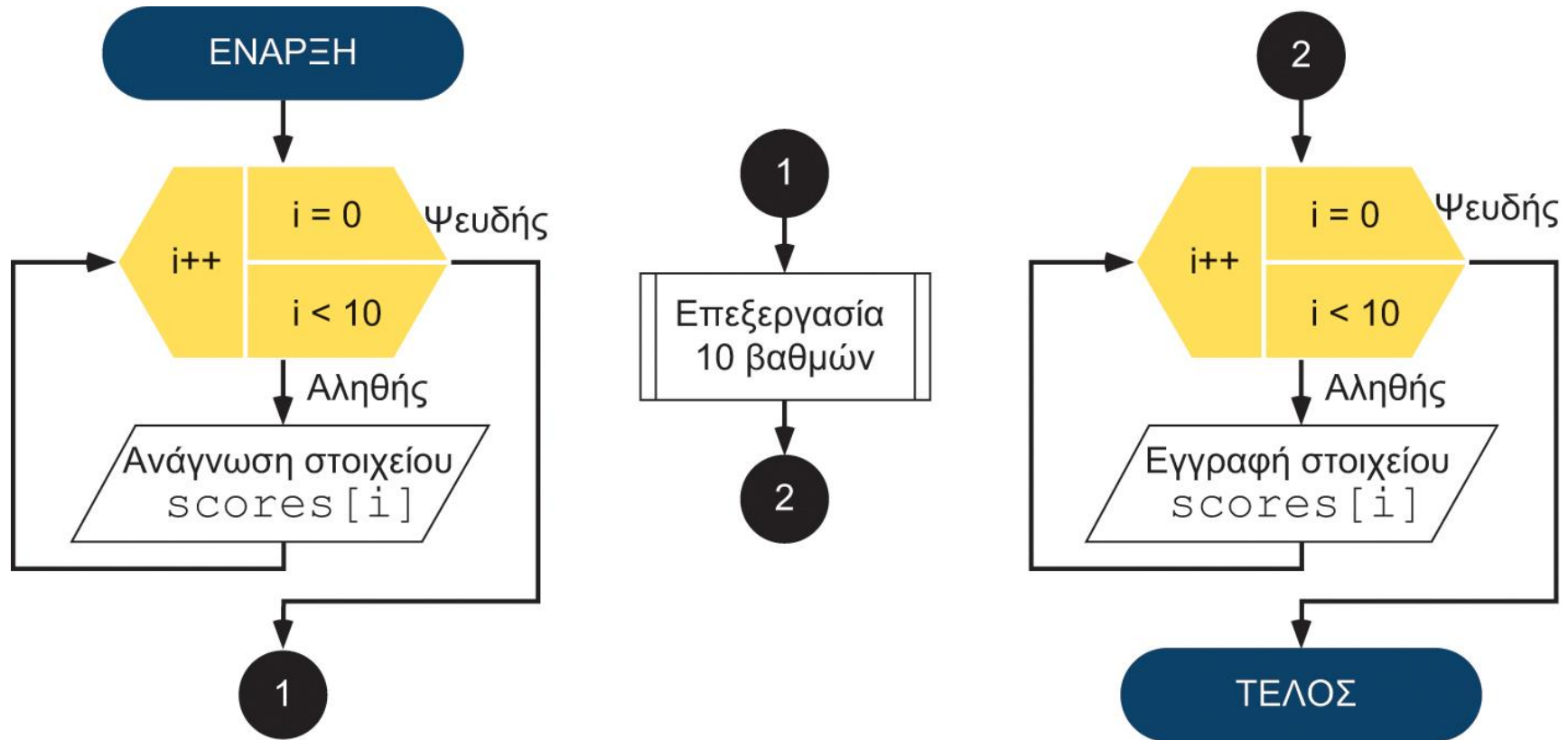


Πίνακας scores
(α) Μορφή με τυπογραφικό δείκτη

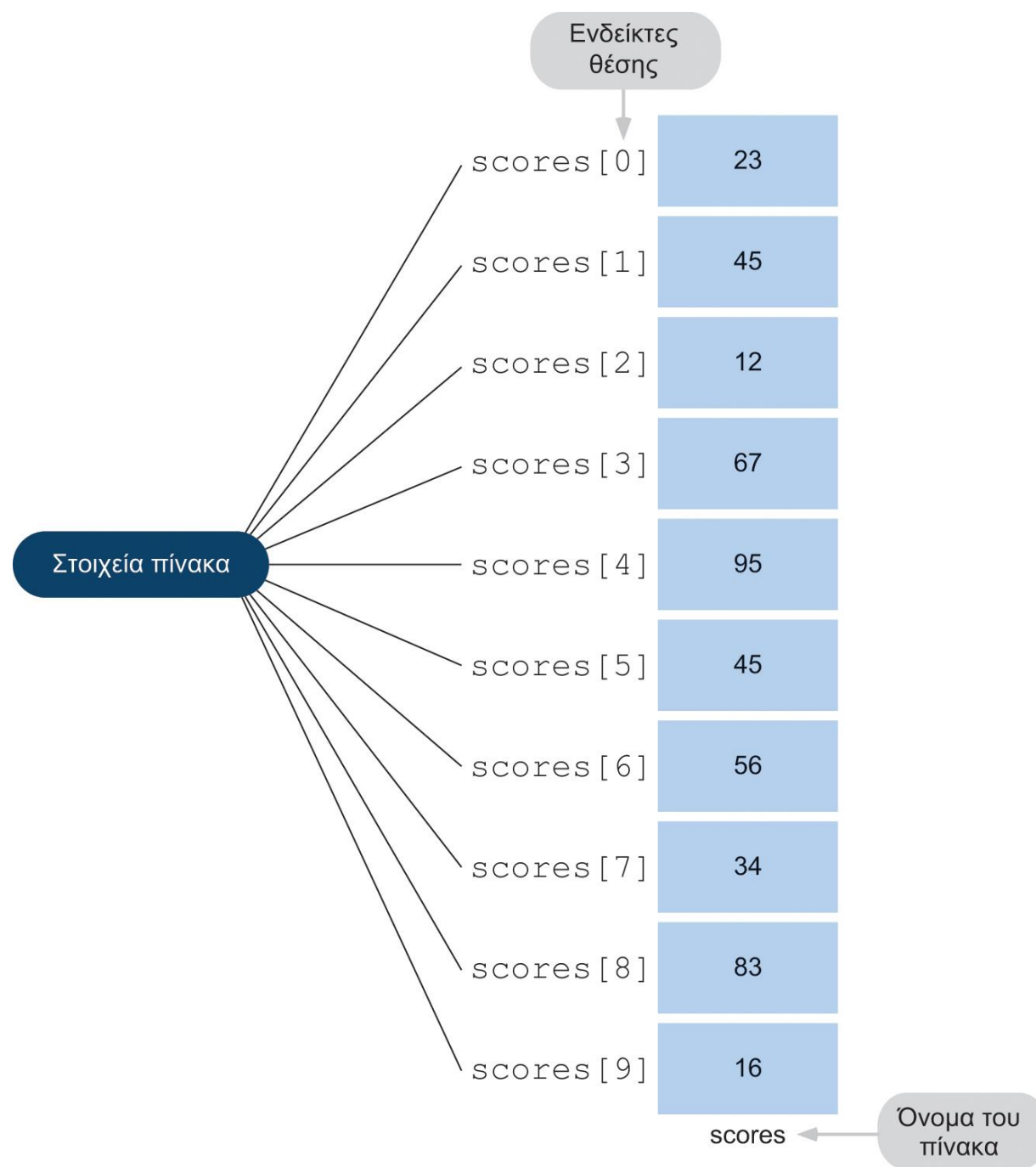


Πίνακας scores
(β) Μορφή με ενδεικτική θέση

Βασικές έννοιες



ΣΧΗΜΑ 8-5 Διάγραμμα ροής για την επεξεργασία των 10 βαθμολογιών του πίνακα `scores`.



ΣΧΗΜΑ 8-6 Ο πίνακας βαθμολογιών.

8.2

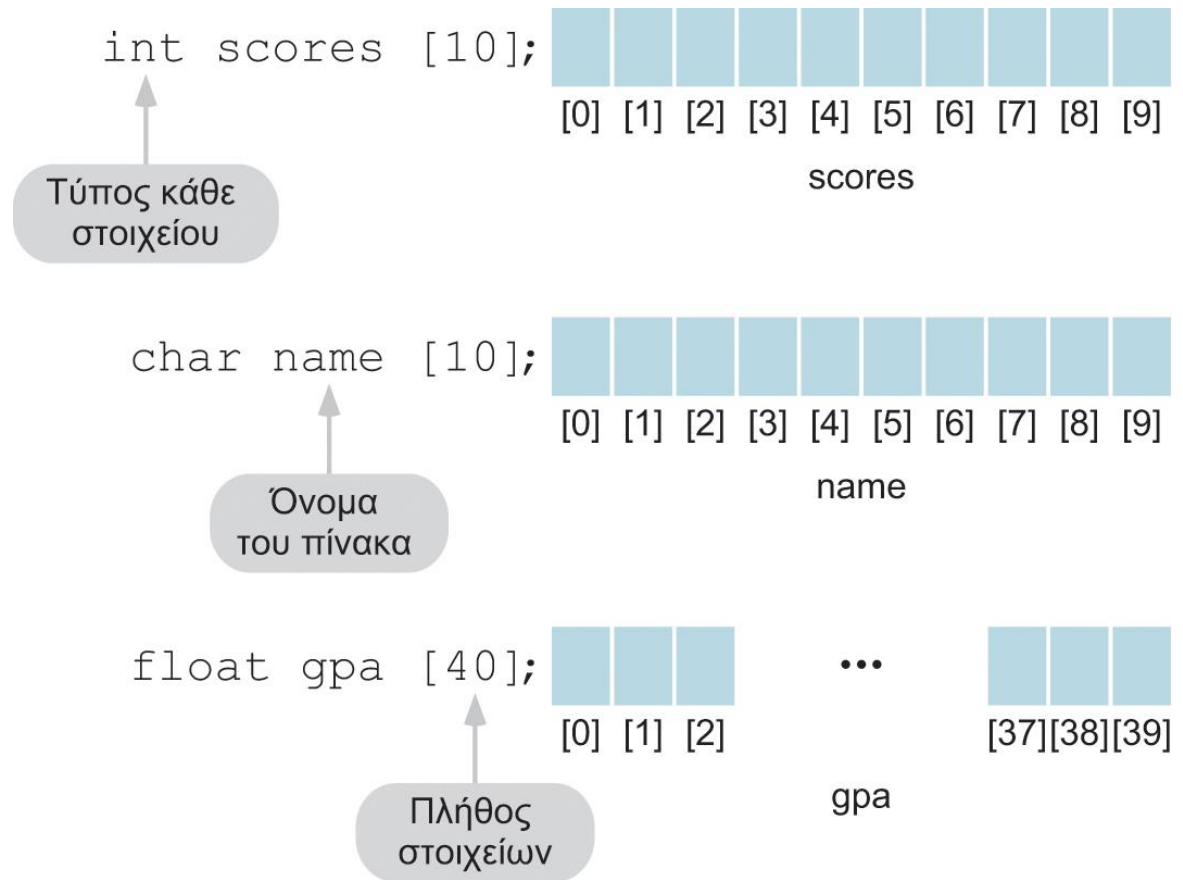
Χρήση πινάκων στη C

Χρήση πινάκων στη C

- Η C υποστηρίζει δύο διαφορετικούς τύπους πινάκων:
- Πίνακες σταθερού μεγέθους (fixed-length array)
 - Το μέγεθος του πίνακα είναι γνωστό όταν γράφεται το πρόγραμμα
- Πίνακες μεταβλητού μεγέθους (variable-length arrays)
 - Το μέγεθος του πίνακα δεν γίνεται γνωστό παρά μόνο κατά τον χρόνο εκτέλεσης του προγράμματος
 - Αυτή η δυνατότητα πρωτοπαρουσιάστηκε στη C99

Δήλωση και ορισμός

- Για να μπορεί να χρησιμοποιηθεί, ένας πίνακας πρέπει κατ' αρχάς να δηλωθεί και να οριστεί η χωρητικότητα του πίνακα
 - Η δήλωση και ο ορισμός καθορίζουν στον μεταγλωττιστή το όνομα του πίνακα, τον τύπο κάθε στοιχείου του και το μέγεθος του πίνακα (το πλήθος των στοιχείων του)



ΣΧΗΜΑ 8-7 Παραδείγματα δήλωσης και ορισμού πινάκων.

Προσπέλαση στοιχείων σε πίνακες

- Η C χρησιμοποιεί τον ενδείκτη θέσης για την προσπέλαση μεμονωμένων στοιχείων ενός πίνακα.
- Ο ενδείκτης θέσης πρέπει να είναι μία ακέραια τιμή, ή μία έκφραση που αποτιμάται σε ακέραια τιμή
 - Ο απλούστερος τρόπος προσπέλασης ενός στοιχείου πίνακα είναι μέσω μιας αριθμητικής σταθεράς
 - Ο ενδείκτης θέσης στοιχείου είναι μία έκφραση—τυπικά, η τιμή μιας μεταβλητής
- Το όνομα του πίνακα αποτελεί επίσης μία συμβολική αναφορά στη διεύθυνση του πρώτου byte που καταλαμβάνει ο πίνακας στη μνήμη
 - Κάθε φορά που χρησιμοποιούμε το όνομα του πίνακα, στην πραγματικότητα αναφερόμαστε στο πρώτο byte του πίνακα

Αποθήκευση τιμών σε πίνακες

- Η δήλωση και ο ορισμός δεν κάνουν τίποτα περισσότερο από το να δεσμεύουν χώρο για τα στοιχεία του πίνακα
 - Δεν αποθηκεύουν τιμές
 - Εάν θέλουμε να αποθηκεύσουμε τιμές στον πίνακα, πρέπει είτε να αρχικοποιήσουμε τα στοιχεία του, είτε να διαβάσουμε τιμές από το πληκτρολόγιο, είτε να αναθέσουμε τιμές σε κάθε μεμονωμένο στοιχείο

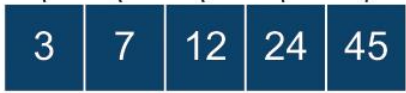
Αποθήκευση τιμών σε πίνακες

- Αρχικοποίηση
 - Όμοια με τις μεταβλητές, η αρχικοποίηση των στοιχείων ενός πίνακα σταθερού μεγέθους μπορεί να γίνει όταν αυτός ορίζεται
 - Οι πίνακες μεταβλητού μεγέθους δεν μπορούν να αρχικοποιηθούν όταν ορίζονται
 - Κατά την αρχικοποίηση, παρέχουμε μία τιμή για κάθε στοιχείο του πίνακα
 - Η μόνη διαφορά είναι ότι οι τιμές πρέπει να περικλείονται σε άγκιστρα και, εάν είναι περισσότερες από μία, να χωρίζονται με κόμματα
 - Αποτελεί σφάλμα μεταγλώττισης ο καθορισμός περισσότερων τιμών απ' όσα είναι τα στοιχεία του πίνακα

Αποθήκευση τιμών σε πίνακες

(α) Βασική αρχικοποίηση

```
int numbers[5] = {3, 7, 12, 24, 45};
```



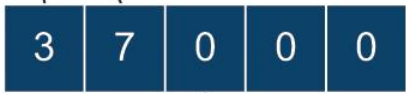
(β) Αρχικοποίηση χωρίς καθορισμό μεγέθους

```
int numbers[ ] = {3, 7, 12, 24, 45};
```



(γ) Μερική αρχικοποίηση

```
int numbers[5] = {3, 7};
```



Στα υπόλοιπα στοιχεία εισάγονται 0

(δ) Αρχικοποίηση σε μηδενικά

```
int lotsOfNumbers [1000] = {0};
```



Εισάγονται 0 σε όλα τα στοιχεία

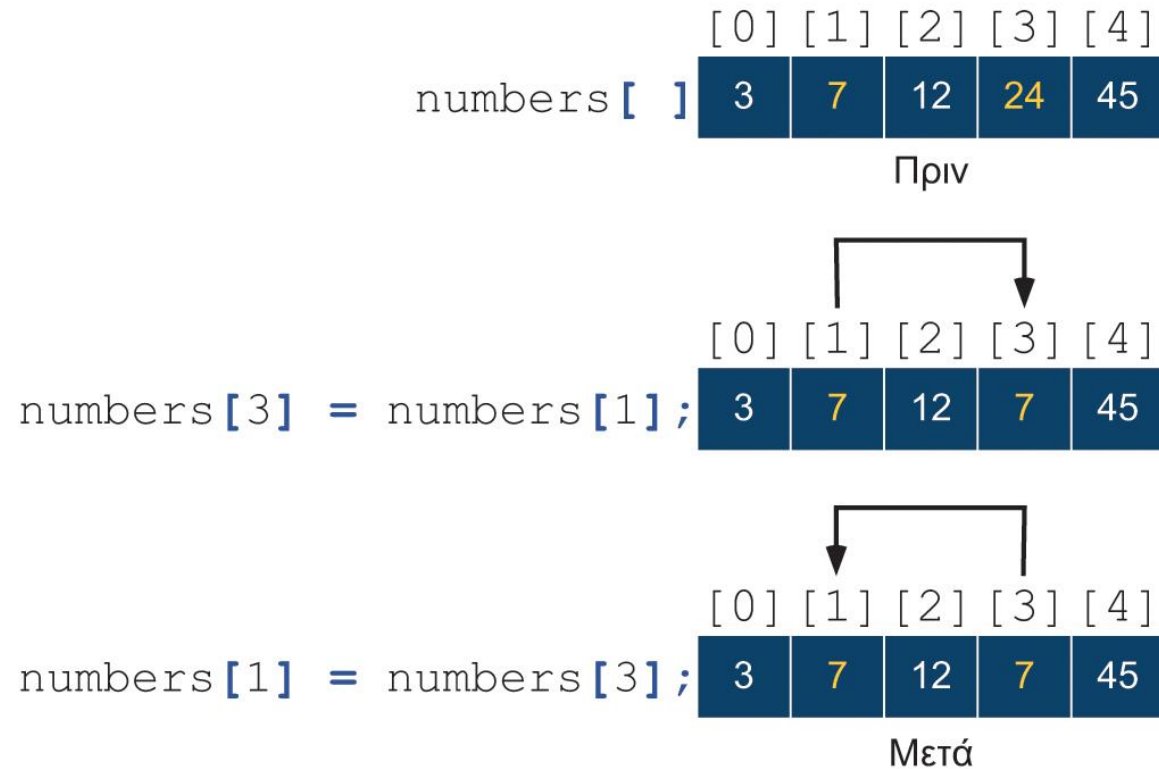
ΣΧΗΜΑ 8-8 Παραδείγματα αρχικοποίησης πινάκων.

Αποθήκευση τιμών σε πίνακες

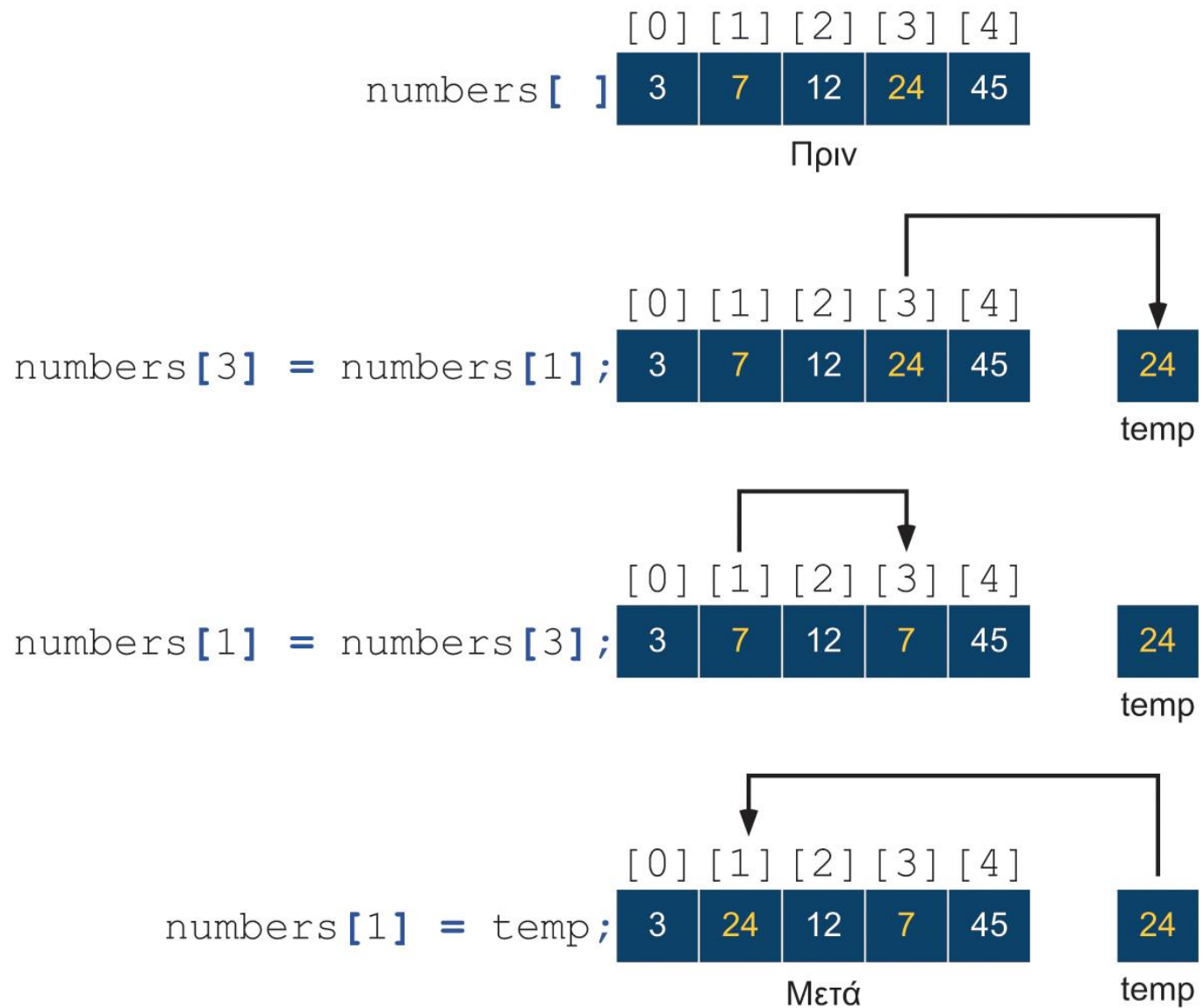
- Εισαγωγή τιμών
 - Ένας άλλος τρόπος εισαγωγής τιμών στα στοιχεία ενός πίνακα είναι η ανάγνωση των τιμών από το πληκτρολόγιο, ή από ένα αρχείο
 - Αυτή η μέθοδος εισαγωγής τιμών μπορεί να υλοποιηθεί με χρήση ενός βρόχου
 - Σε περιπτώσεις όπου ο πίνακας πρέπει να «γεμίσει» εξ ολοκλήρου, κρίνεται ως καταλληλότερος ο βρόχος for, επειδή ο αριθμός των στοιχείων είναι σταθερός και γνωστός

Αποθήκευση τιμών σε πίνακες

- Ανάθεση τιμών
 - Μπορούμε να αναθέσουμε τιμές σε μεμονωμένα στοιχεία ενός πίνακα χρησιμοποιώντας τον τελεστή ανάθεσης
 - Οποιαδήποτε τιμή μπορεί να αναχθεί στον κατάλληλο τύπο δεδομένων, μπορεί να ανατεθεί σε ένα μεμονωμένο στοιχείο πίνακα
- Εναλλαγή τιμών στοιχείων
 - Μία κοινή εφαρμογή ενός πίνακα είναι η εναλλαγή των περιεχομένων δύο στοιχείων· αυτό αναφέρεται ως εναλλαγή τιμών στοιχείων
- Εκτύπωση τιμών
 - Μία άλλη κοινή εφαρμογή είναι η εκτύπωση των περιεχομένων ενός πίνακα



ΣΧΗΜΑ 8-9 Εναλλαγή τιμών σε στοιχεία του πίνακα `scores`: ο λάθος τρόπος.



ΣΧΗΜΑ 8-10 Εναλλαγή στοιχείων ενός πίνακα με χρήση προσωρινής μεταβλητής.

Αναφορές σε στοιχεία πινάκων και ζητήματα προτεραιότητας

- Οι αναφορές σε στοιχεία πινάκων είναι επιθεματικές εκφράσεις
 - Οι αναφορές σε πίνακες (σε στοιχεία πινάκων) έχουν πολύ υψηλή προτεραιότητα, 16
- Οι αγκύλες ([]) είναι στην πραγματικότητα τελεστές και, ουσιαστικά, δημιουργούν μία επιθεματική έκφραση από μία βασική έκφραση
- Όταν υποδεικνύουμε ένα στοιχείο πίνακα σε μία έκφραση, η τιμή του πρέπει να προσδιορίζεται αμέσως

Έλεγχος εύρους αναφορών

- Η γλώσσα C δεν διαθέτει κάποιον μηχανισμό για τον έλεγχο του εύρους των αναφορών που σχετίζονται με έναν πίνακα
 - Είναι ευθύνη του προγραμματιστή, να διασφαλίσει ότι όλες οι αναφορές που γίνονται σε στοιχεία πινάκων (χρησιμοποιώντας ενδείκτες θέσης) είναι έγκυρες και εντός των ορίων του πίνακα

Έλεγχος εύρους αναφορών

- Εάν σε μία έκφραση χρησιμοποιήσουμε άκυρο (εκτός ορίων πίνακα) ενδείκτη για να αναφερθούμε σε ένα στοιχείο πίνακα, θα λάβουμε απρόβλεπτα αποτελέσματα
 - Όταν χρησιμοποιούμε άκυρο ενδείκτη θέσης στοιχείου σε μία εντολή ανάθεσης, καταστρέφουμε τα περιεχόμενα κάποιου, μη προσδιορίσιμου τμήματος της μνήμης που έχει διατεθεί στο πρόγραμμα
 - Συνήθως, αλλά όχι πάντα, το πρόγραμμα συνεχίζει να εκτελείται και, είτε παράγει απρόβλεπτα αποτελέσματα, είτε τελικά καταρρέει

8.3

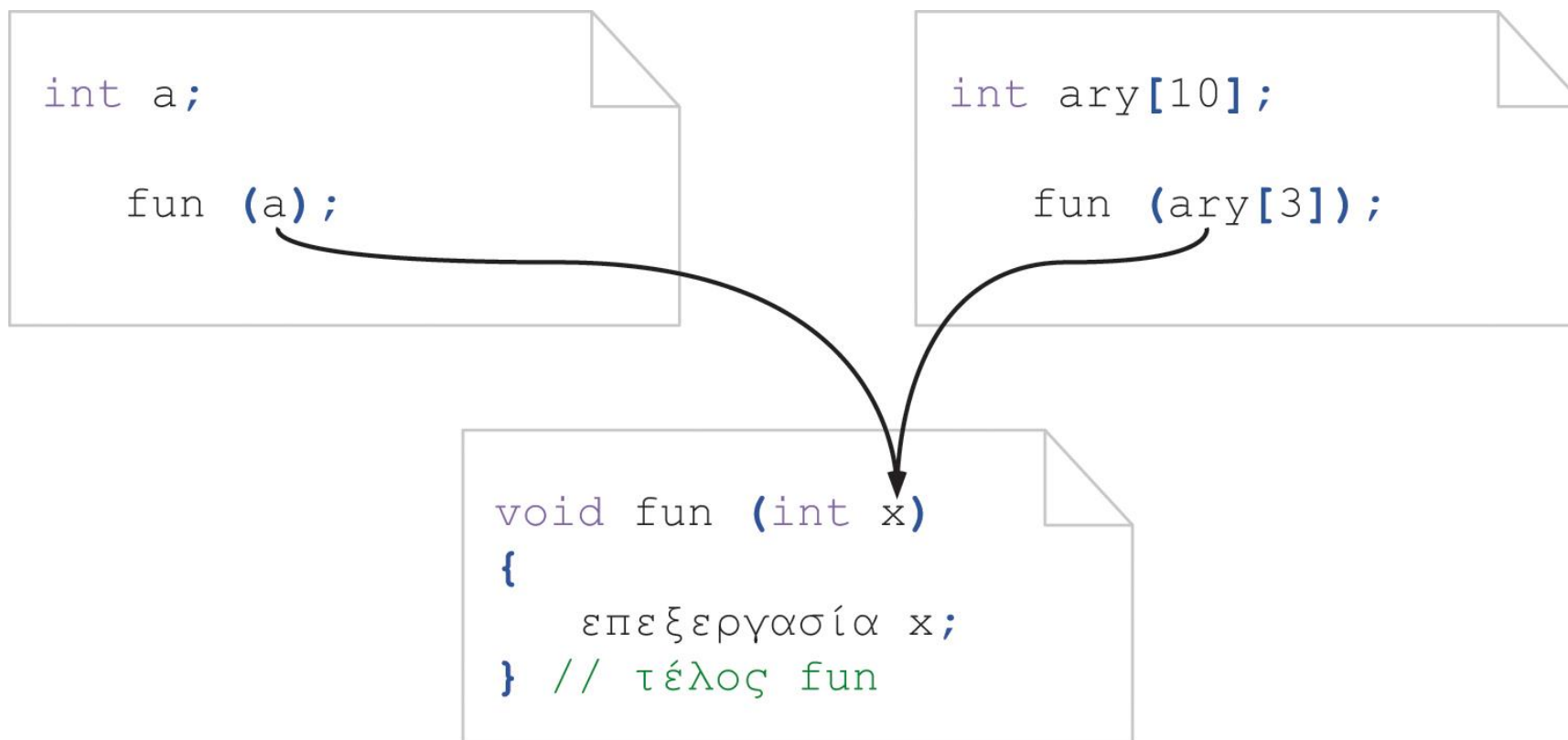
Πέρασμα πινάκων σε συναρτήσεις

Πέρασμα μεμονωμένων στοιχείων

- Πέρασμα διά τιμής

- Μπορούμε να περάσουμε μεμονωμένα στοιχεία ενός πίνακα σε μία συνάρτηση, όπως ακριβώς περνάμε οποιαδήποτε άλλη τιμή δεδομένων
 - Υπό την προϋπόθεση ότι ο τύπος του στοιχείου του πίνακα ταιριάζει με τον τύπο της αναμενόμενης παραμέτρου, το στοιχείο μπορεί να περαστεί στη συνάρτηση
 - Η καλούμενη συνάρτηση δεν μπορεί να διακρίνει εάν η τιμή που λαμβάνει προέρχεται από πίνακα, μία μεταβλητή, ή μία έκφραση

Πέρασμα μεμονωμένων στοιχείων (2/3)



ΣΧΗΜΑ 8-11 Πέρασμα στοιχείων πίνακα σε συνάρτηση.

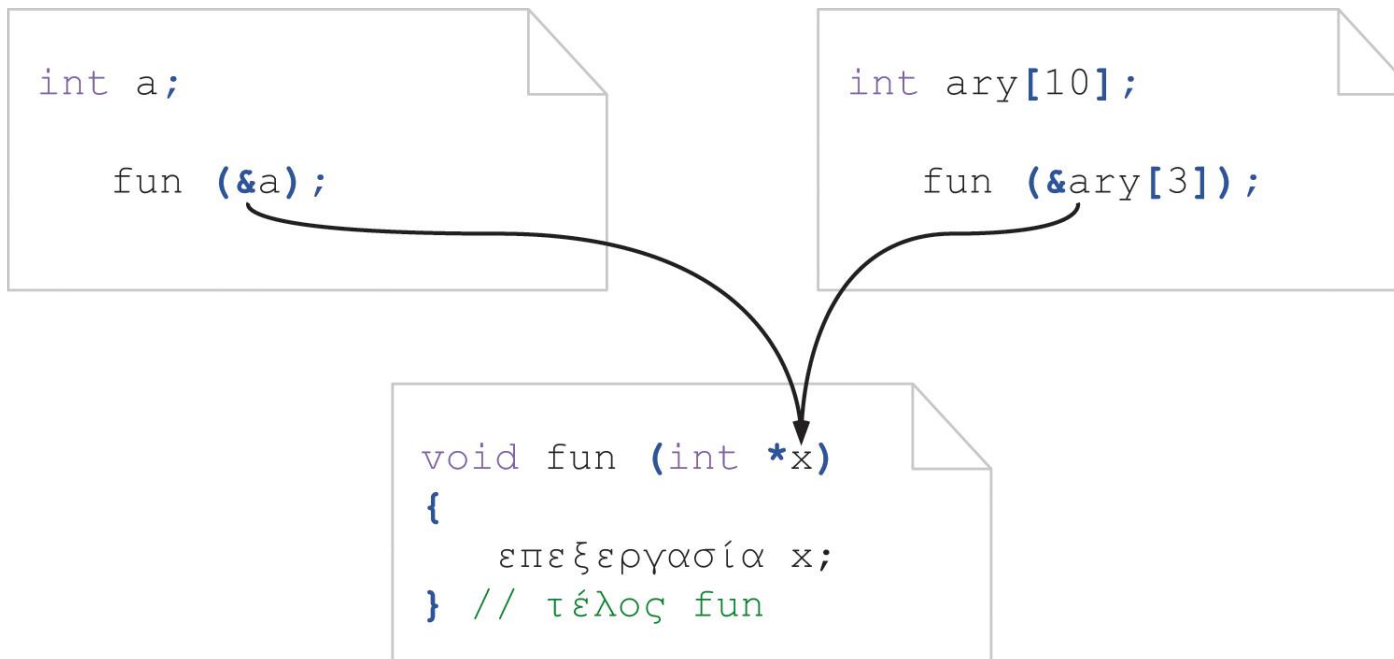
Πέρασμα μεμονωμένων στοιχείων

- Πέρασμα δι' αναφοράς

- Το πέρασμα διευθύνσεων μνήμης επιτρέπει την αμφίδρομη επικοινωνία μεταξύ συναρτήσεων
 - Μπορούμε να περάσουμε τη διεύθυνση μνήμης ενός μεμονωμένου στοιχείου ενός πίνακα, όπως ακριβώς περνάμε τη διεύθυνση μνήμης οποιασδήποτε μεταβλητής
- Για να περάσουμε τη διεύθυνση μνήμης ενός στοιχείου πίνακα, προτάσσουμε τον τελεστή διεύθυνσης στην αναφορά του στοιχείου

Πέρασμα μεμονωμένων στοιχείων

- Το πέρασμα της διεύθυνσης μνήμης ενός στοιχείου πίνακα απαιτεί δύο αλλαγές στην καλούμενη συνάρτηση.
 - η συνάρτηση πρέπει να δηλώσει ότι λαμβάνει μία διεύθυνση
 - πρέπει να χρησιμοποιηθεί ο τελεστής έμμεσης αναφοράς (*) για προσπέλαση των δεδομένων του στοιχείου



Πέρασμα διεύθυνσης πίνακα

- Στη C, το όνομα ενός πίνακα αντιμετωπίζεται ως μία στοιχειώδης έκφραση, η τιμή της οποίας είναι η διεύθυνση μνήμης του πρώτου στοιχείου του πίνακα
 - Επειδή οι δεικτοδοτούμενες αναφορές είναι απλώς υπολογιζόμενες διευθύνσεις, η μόνη πληροφορία που απαιτείται για την προσπέλαση οποιουδήποτε από τα στοιχεία του πίνακα είναι η διεύθυνση αρχής του πίνακα στη μνήμη

ΣΧΗΜΑ 8-13

Πέρασμα πίνακα
δι' αναφοράς.

```
int ary[10];  
  
fun (ary);
```

```
void fun (int fAry[ ])
{
    επεξεργασία x;
} // τέλος fun
```

Πίνακας σταθερού μεγέθους

```
int ary[size];  
  
fun (ary);
```

```
void fun (int fAry[*])
{
    επεξεργασία x;
} // τέλος fun
```

Πίνακας μεταβλητού μεγέθους

Πέρασμα διεύθυνσης πίνακα

- Πίνακες σταθερού μεγέθους
 - Για έναν πίνακα σταθερού μεγέθους, δεν χρειάζεται να καθορίσουμε τον αριθμό των στοιχείων
 - Επειδή στην πραγματικότητα ο πίνακας ορίζεται κάπου αλλού, το μόνο που έχει σημασία είναι ότι ο μεταγλωττιστής γνωρίζει ότι πρόκειται για πίνακα
- Πίνακες μεταβλητού μεγέθους
 - Σε μία δήλωση συνάρτησης, δηλώνουμε έναν πίνακα ως μεταβλητού μεγέθους χρησιμοποιώντας είτε έναν αστερίσκο, είτε μία μεταβλητή για το μέγεθος του πίνακα
 - Στον ορισμό της συνάρτησης, πρέπει να χρησιμοποιήσουμε το όνομα της μεταβλητής και, βάσει των τυπικών κανόνων σύνταξης, η μεταβλητή πρέπει να οριστεί πριν από τον πίνακα

Πέρασμα διεύθυνσης πίνακα

- Πέρασμα ενός πίνακα ως σταθερά
 - Όταν μία συνάρτηση λαμβάνει ως παράμετρο έναν πίνακα αλλά δεν αλλάζει το περιεχόμενό του, ο πίνακας πρέπει να περνιέται στη συνάρτηση ως σταθερά
 - Αυτό αποτρέπει πιθανή αλλαγή του πίνακα από τη συνάρτηση
 - Για να δηλώσουμε έναν πίνακα ως σταθερά, αναγράφουμε τον τύπο του προτάσσοντας τον προσδιοριστικό τύπου `const`, όπως φαίνεται στον ακόλουθο ορισμό συνάρτησης

```
double average (const int ary[], int size);
```

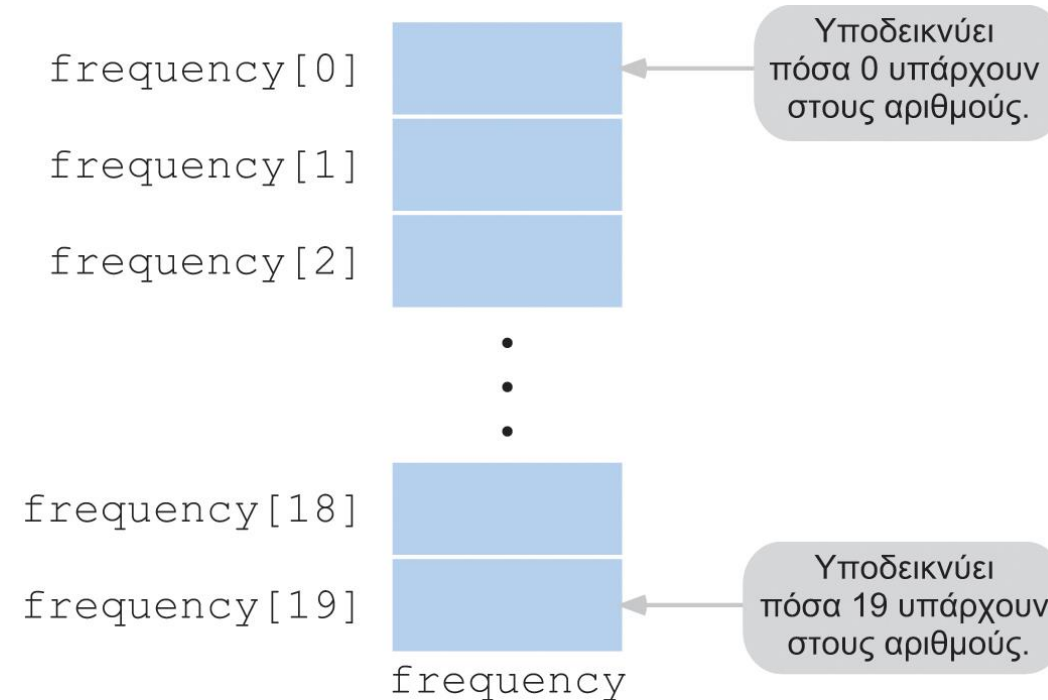
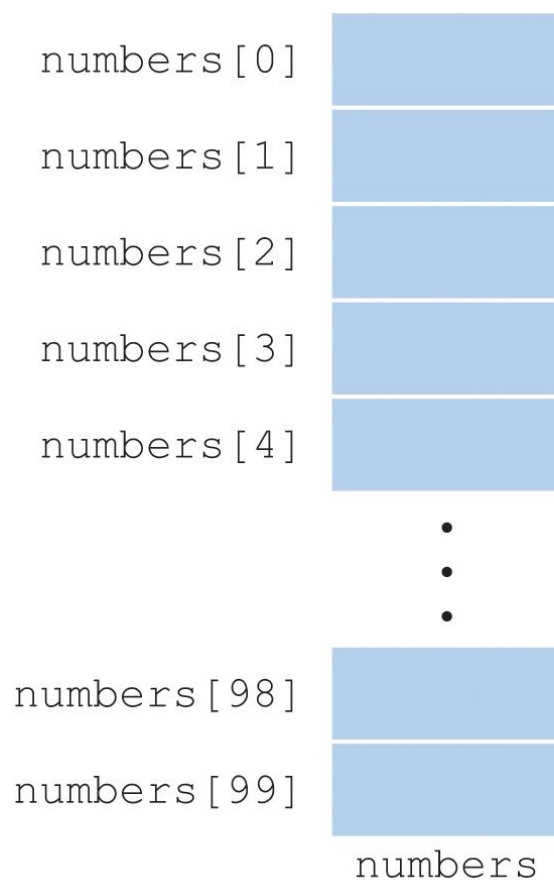
 - Αφού ένας πίνακας δηλωθεί ως `const`, ο μεταγλωττιστής θα αναφέρει σφάλμα σε οποιαδήποτε εντολή επιχειρεί να αλλάξει τα περιεχόμενά του

8.4

Χρήση πινάκων σε εφαρμογές

Πίνακες συχνοτήτων

Ένας πίνακας συχνοτήτων καταγράφει τη συχνότητα εμφάνισης πανομοιότυπων τιμών σε μία ακολουθία



ΣΧΗΜΑ 8-14 Πίνακας δείγματος δεδομένων (`numbers`) και πίνακας συχνοτήτων (`frequency`).

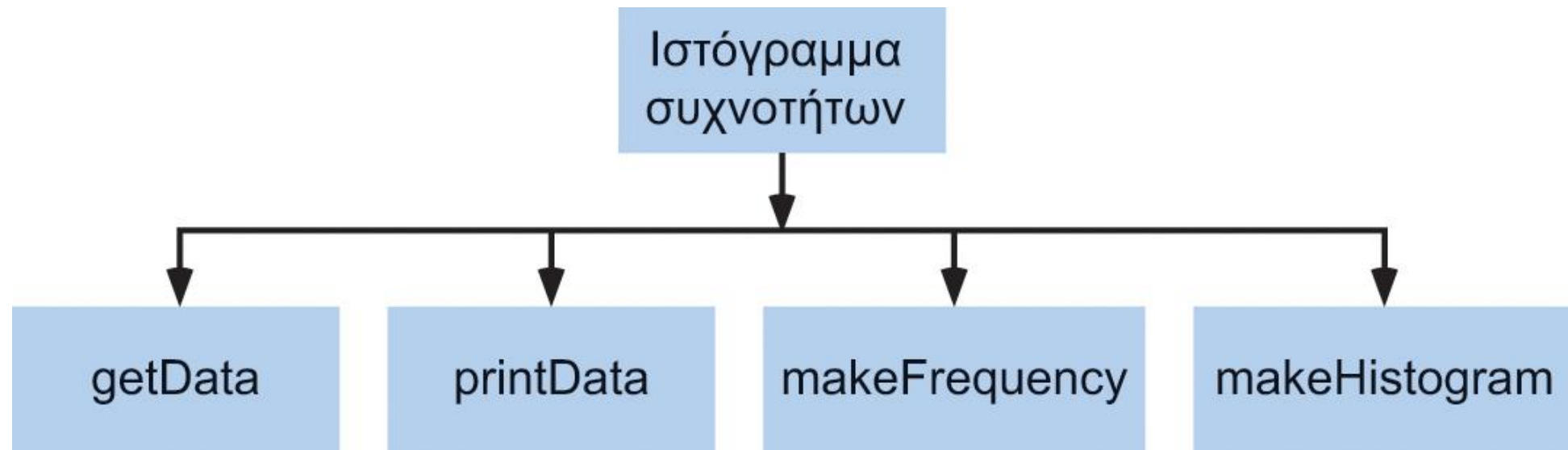
Ιστογράμματα

- Το ιστόγραμμα είναι μία διαγραμματική αναπαράσταση ενός πίνακα συχνοτήτων
- Αντί να εκτυπώνουμε τις τιμές των στοιχείων για να δείξουμε τη συχνότητα εμφάνισης κάθε τιμής δεδομένων, εκτυπώνουμε ένα ιστόγραμμα υπό μορφή ραβδογράμματος



ΣΧΗΜΑ 8-15 Ιστόγραμμα συχνοτήτων.

Ιστογράμματα



ΣΧΗΜΑ 8-16 Σχεδίαση προγράμματος για την παραγωγή ιστογράμματος συχνοτήτων εμφάνισης τιμών.

Τυχαίες μεταθέσεις αριθμών με μοναδικότητα

- Τυχαία μετάθεση: ένα σύνολο τυχαίων αριθμών ή άλλων αντικειμένων στο οποίο δεν υπάρχουν επαναλαμβανόμενα στοιχεία
- Ουσιαστικά είναι ένα σύνολο τυχαίων αριθμών από το οποίο πρέπει να εξαλείψουμε τις διπλότυπες τιμές

randNos [0]	8	haveRand [0]	0	
randNos [1]	3	haveRand [1]	1	
randNos [2]	5	haveRand [2]	0	← 0: όχι τυχαία παραγωγή του 2
randNos [3]	1	haveRand [3]	1	
randNos [4]	7	haveRand [4]	0	
randNos [5]		haveRand [5]	1	← 1: τυχαία παραγωγή του 5
randNos [6]		haveRand [6]	0	
randNos [7]		haveRand [7]	1	
randNos [8]		haveRand [8]	1	
randNos [9]		haveRand [9]	0	
randNos		haveRand		
Μετά την παραγωγή των πρώτων πέντε τυχαίων αριθμών				

8.5

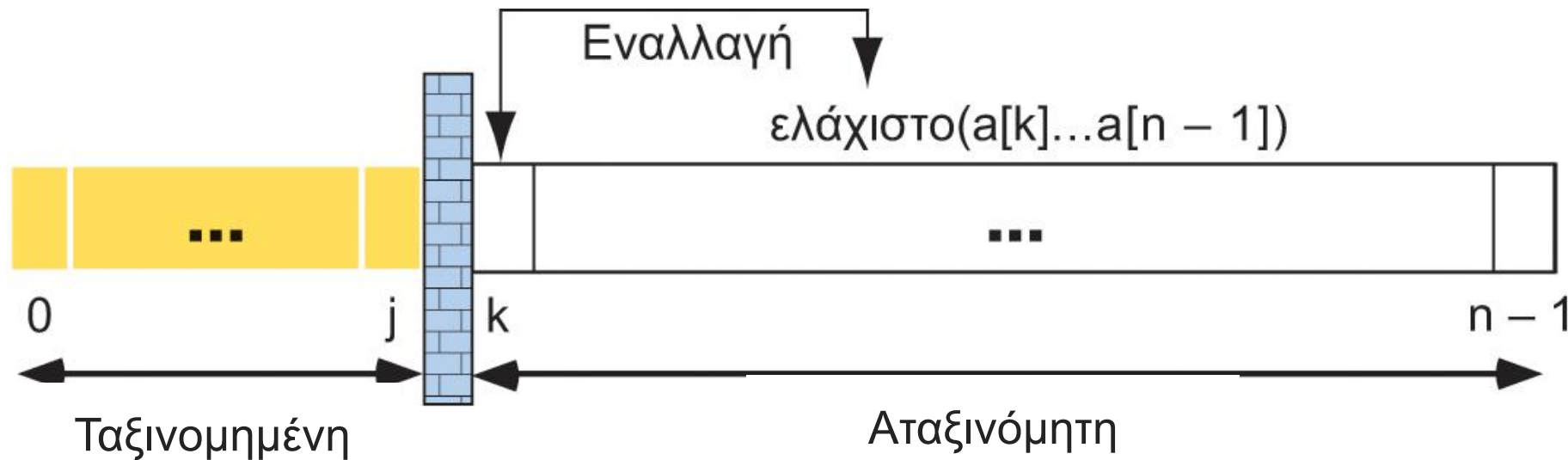
Ταξινόμηση

Ταξινόμηση

- Ζούμε σε μία εποχή όπου ο μέσος άνθρωπος αντιμετωπίζει, επί καθημερινής βάσεως, πληθώρα δεδομένων
- Εάν δεν είχαμε δυνατότητες οργάνωσης των δεδομένων, θα ξοδεύαμε ώρες προσπαθώντας να βρούμε μία πληροφορία
- Μία από τις πλέον κοινές και χρήσιμες εφαρμογές στην επιστήμη των υπολογιστών είναι η **ταξινόμηση** (sort)
- Η διαδικασία ταξινόμησης οργανώνει τα δεδομένα με κάποια σειρά, βάσει των τιμών τους

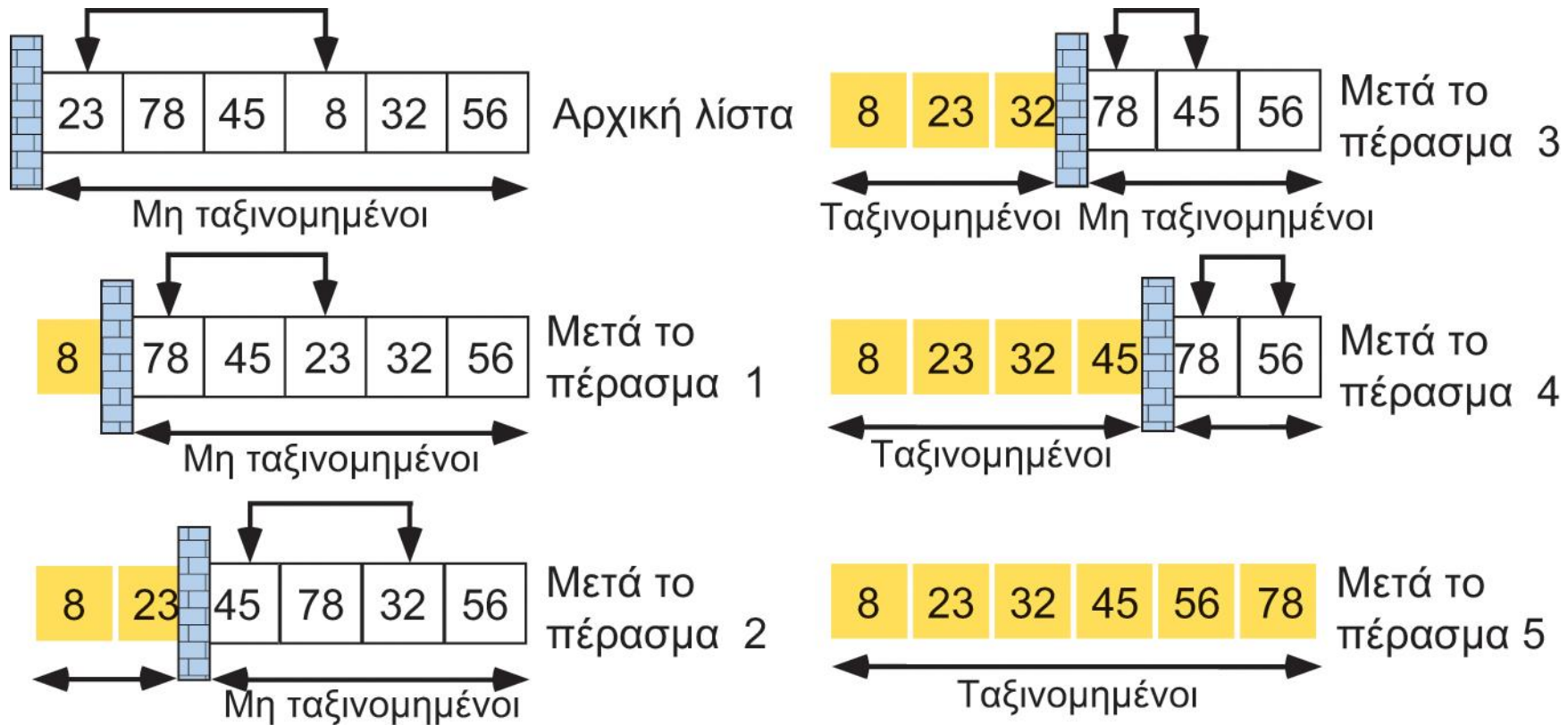
Ταξινόμηση με επιλογή (Selection Sort)

- Η λίστα δεδομένων χωρίζεται σε δύο τμήματα ή υπολίστες, μία «ταξινομημένη» και μία «αταξινομήτη» (θεωρούμε ότι υπάρχει ένα νοητό διαχωριστικό μεταξύ των δύο υπολυστών)

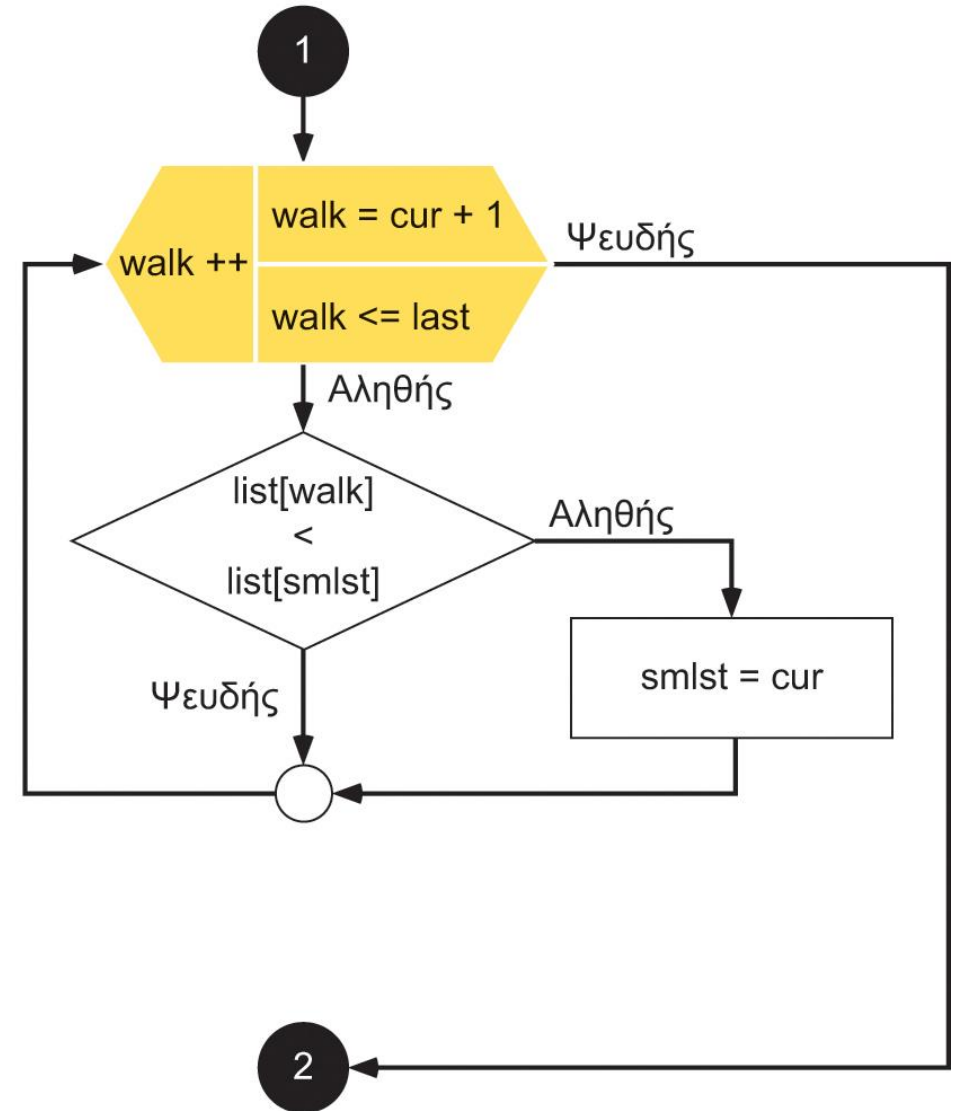
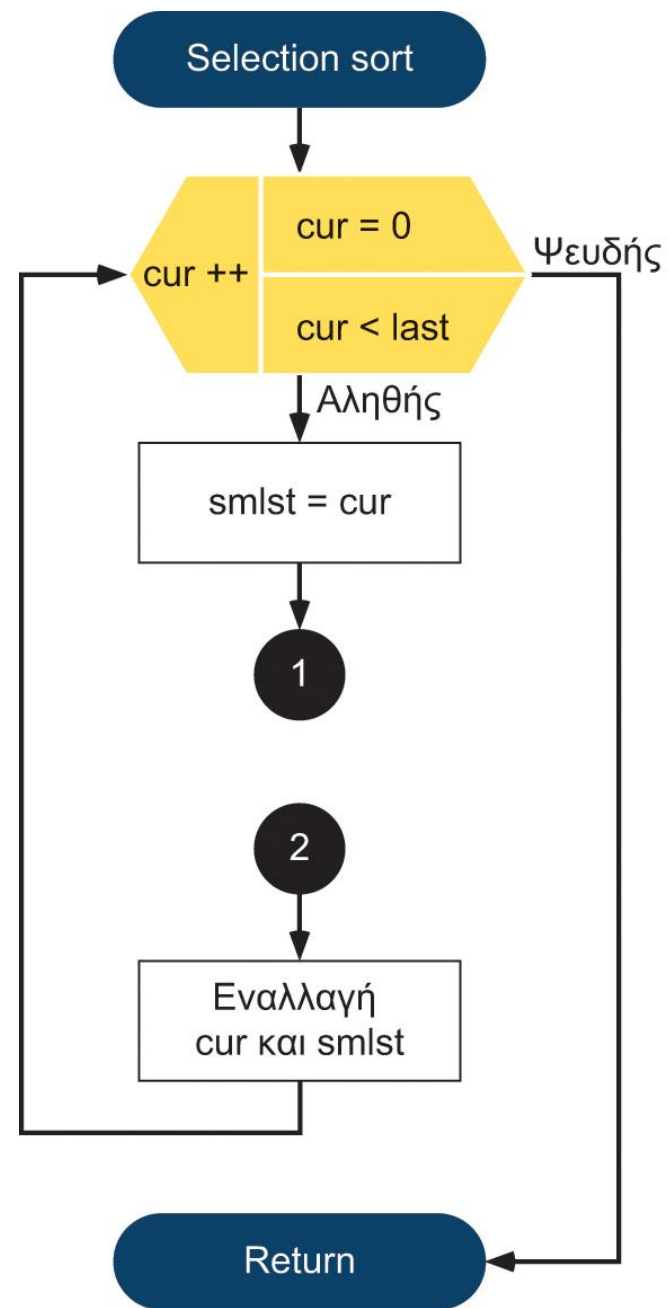


ΣΧΗΜΑ 8-18 Βασική αρχή της ταξινόμησης με επιλογή.

Ταξινόμηση με επιλογή (Selection Sort)



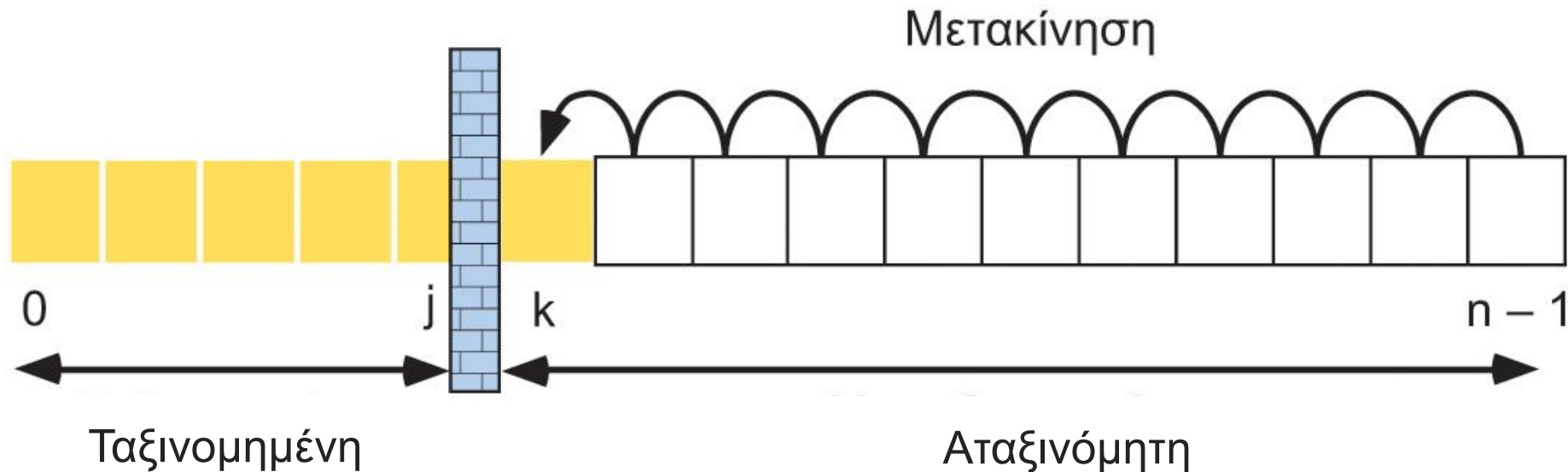
ΣΧΗΜΑ 8-19 Σχηματική αναπαράσταση ταξινόμησης με επιλογή.



ΣΧΗΜΑ 8-20 Σχεδίαση του προγράμματος ταξινόμησης με επιλογή.

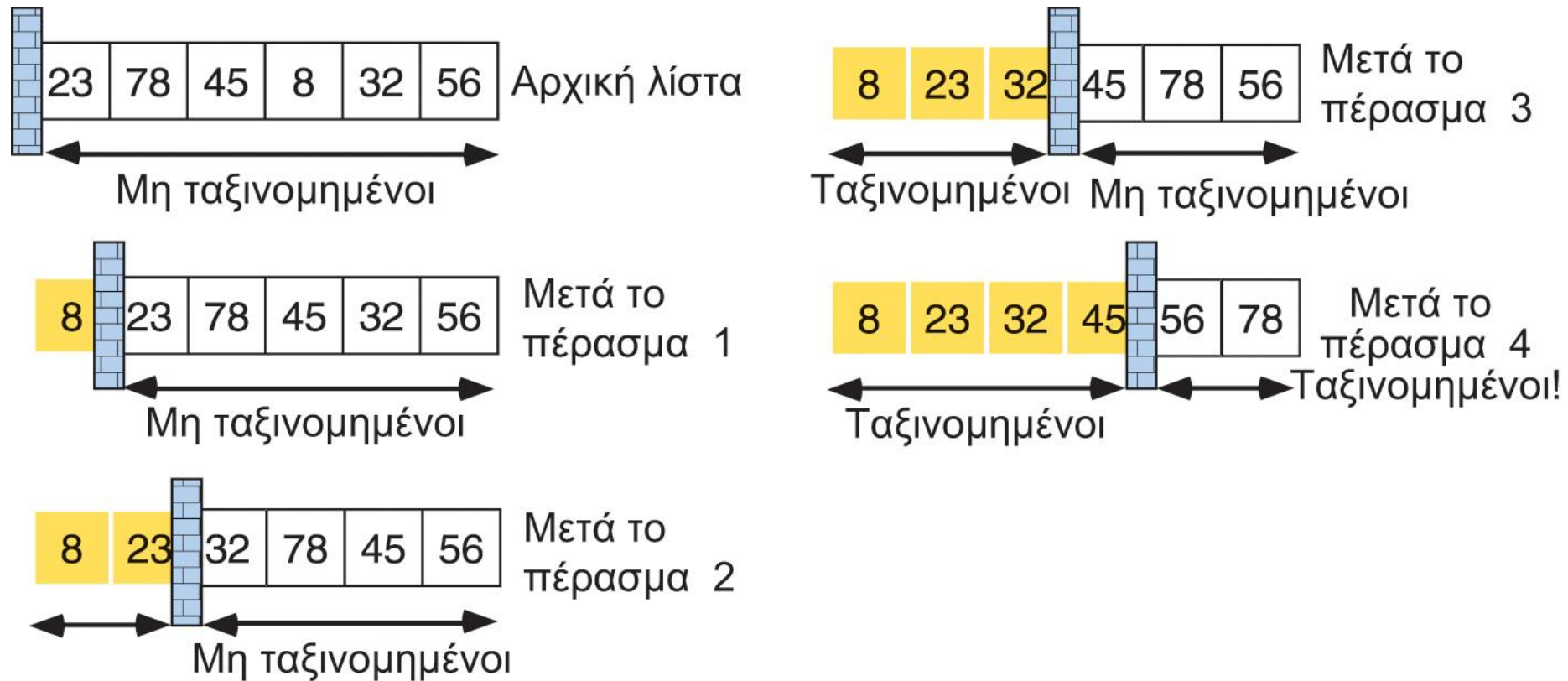
Ταξινόμηση φουσαλίδας (Bubble Sort)

- Η λίστα των προς ταξινόμηση δεδομένων χωρίζεται σε δύο υπολίστες, την «ταξινομημένη» και την «αταξινομητη»

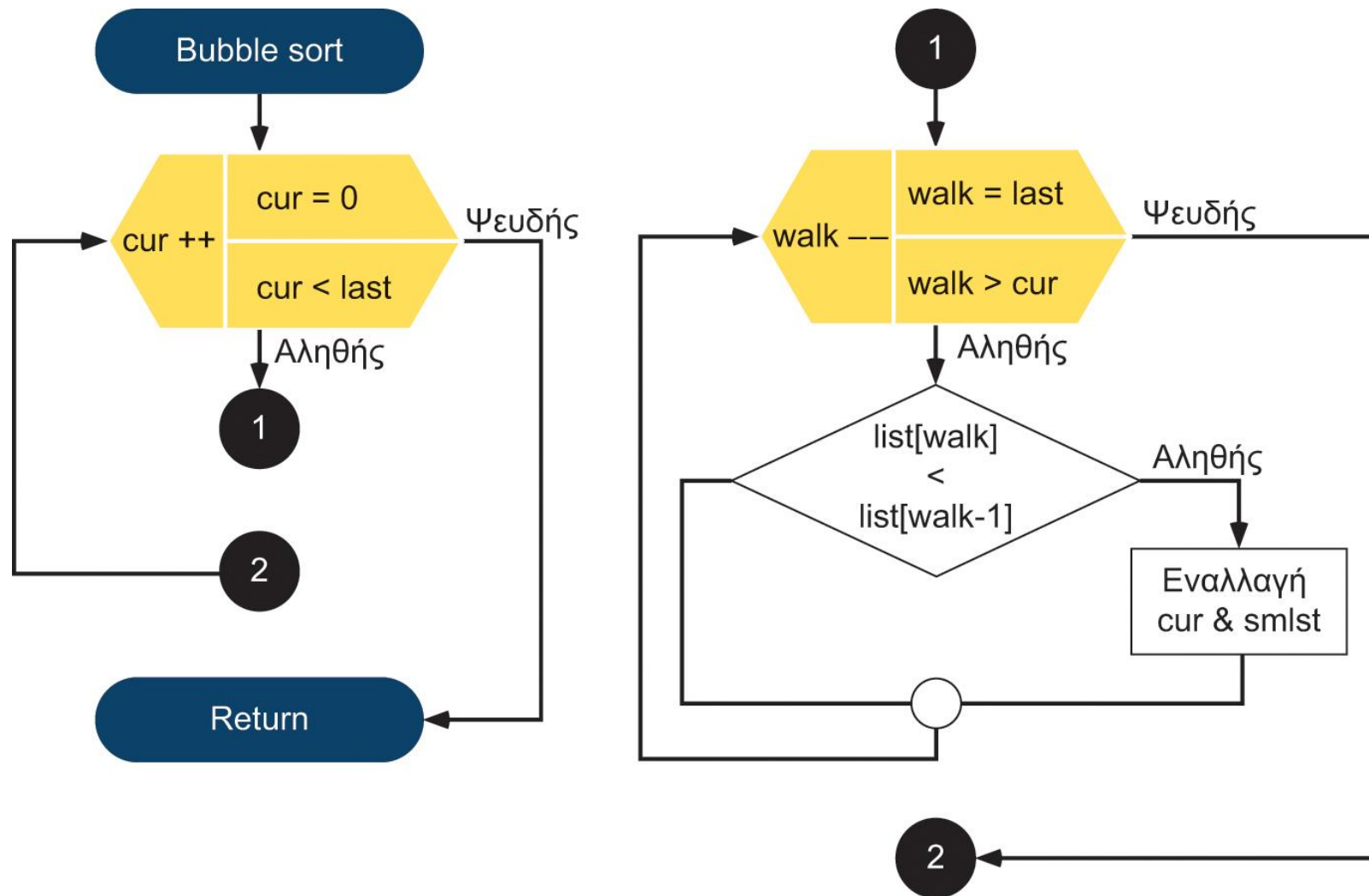


ΣΧΗΜΑ 8-21 Βασική αρχή του αλγορίθμου ταξινόμησης φουσαλίδας.

Ταξινόμηση φουσάλιδας (Bubble Sort)



ΣΧΗΜΑ 8-22 Σχηματική αναπαράσταση της ταξινόμησης φουσάλιδας.



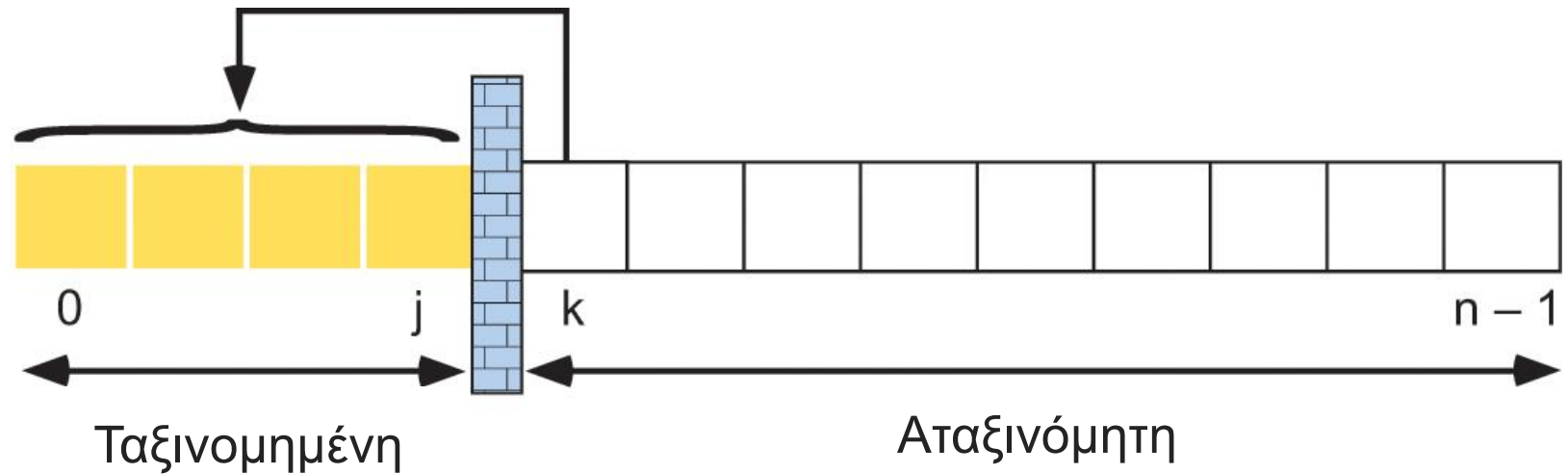
ΣΧΗΜΑ 8-23 Αρχή λειτουργίας του αλγόριθμου ταξινόμησης φυσαλίδας.

Ταξινόμηση με παρεμβολή (Insertion Sort)

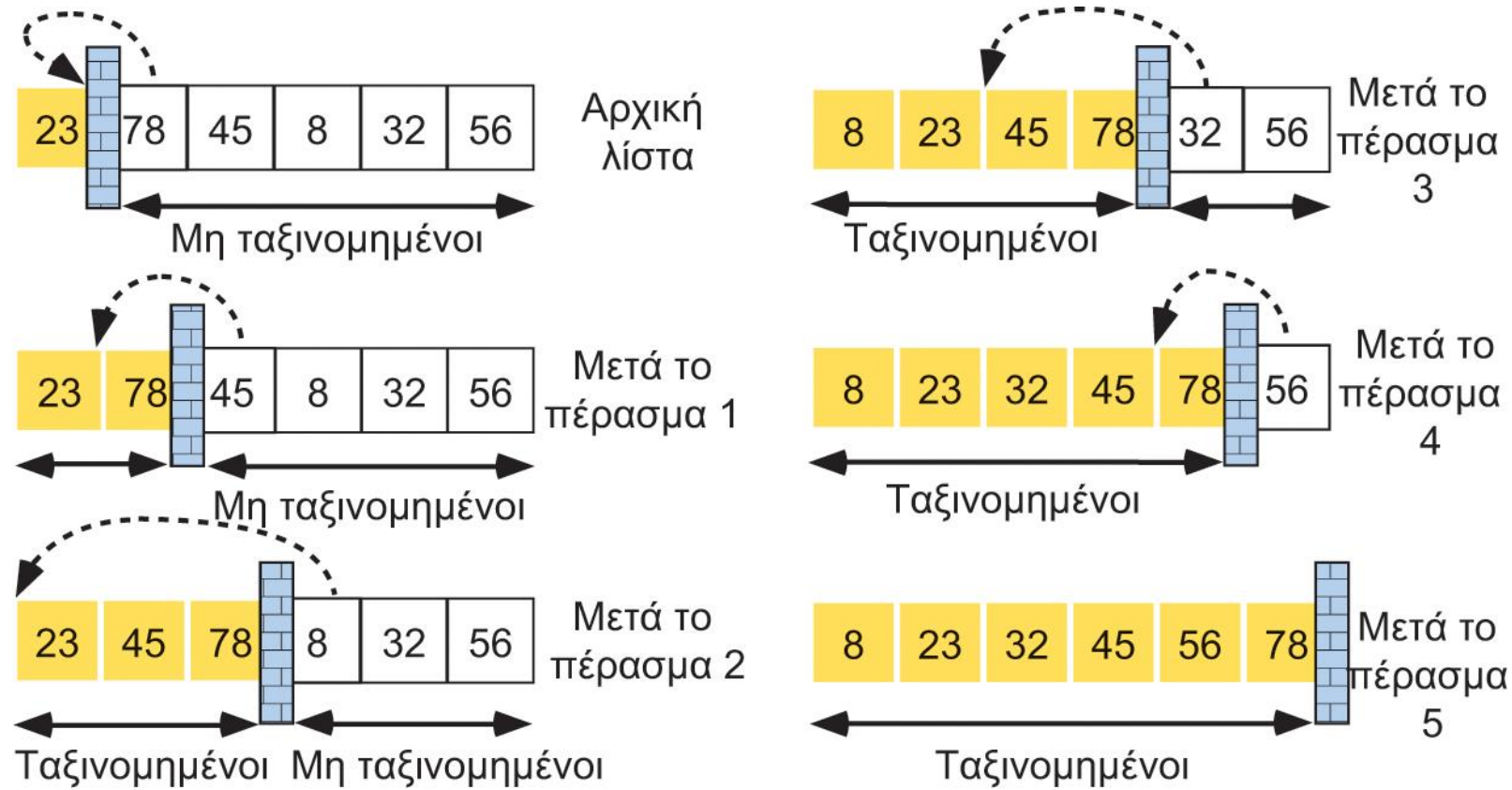
- Η λίστα των προς ταξινόμηση δεδομένων χωρίζεται σε δύο υπολίστες, την «ταξινομημένη» και την «αταξινόμητη»

ΣΧΗΜΑ 8-24

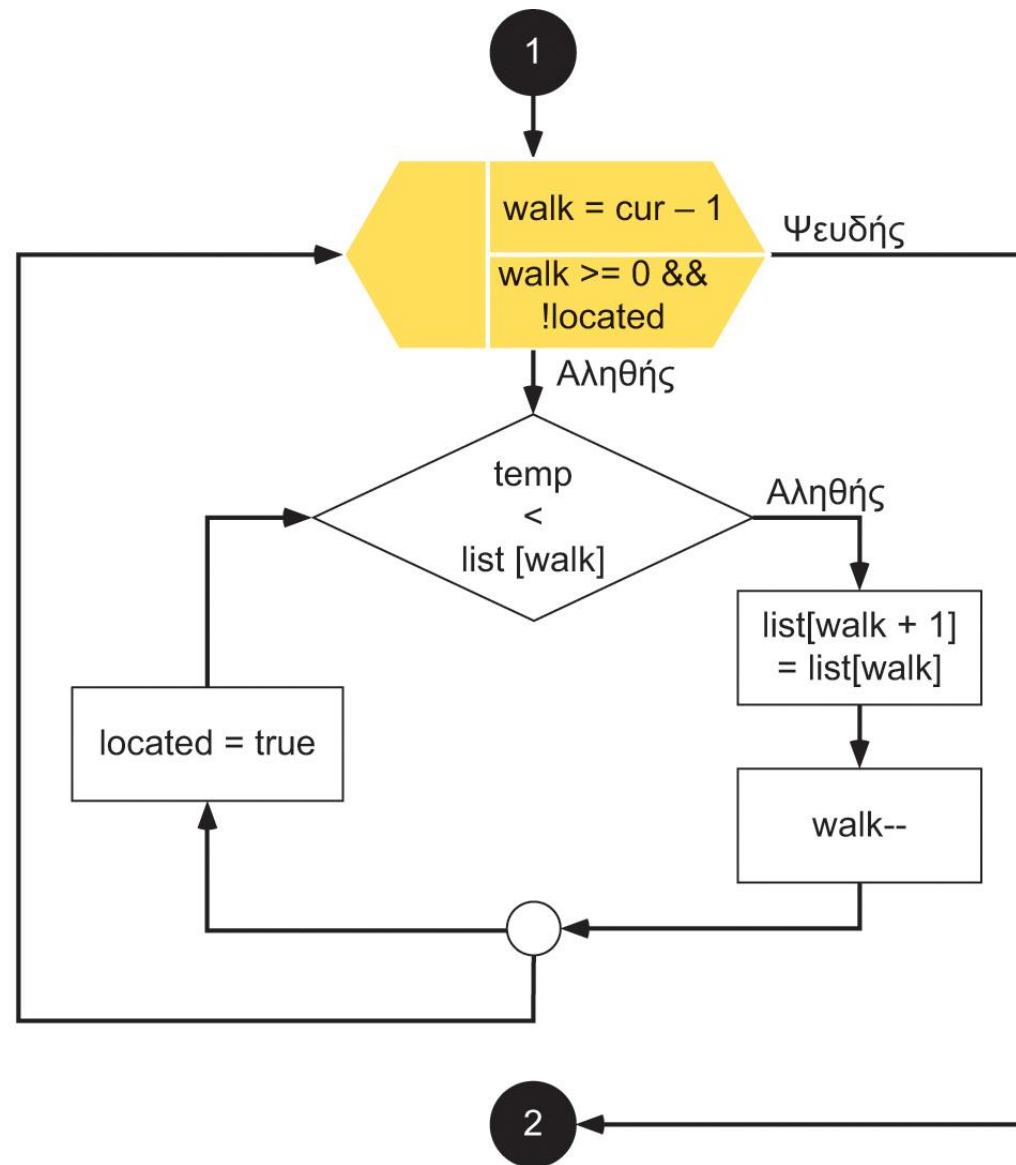
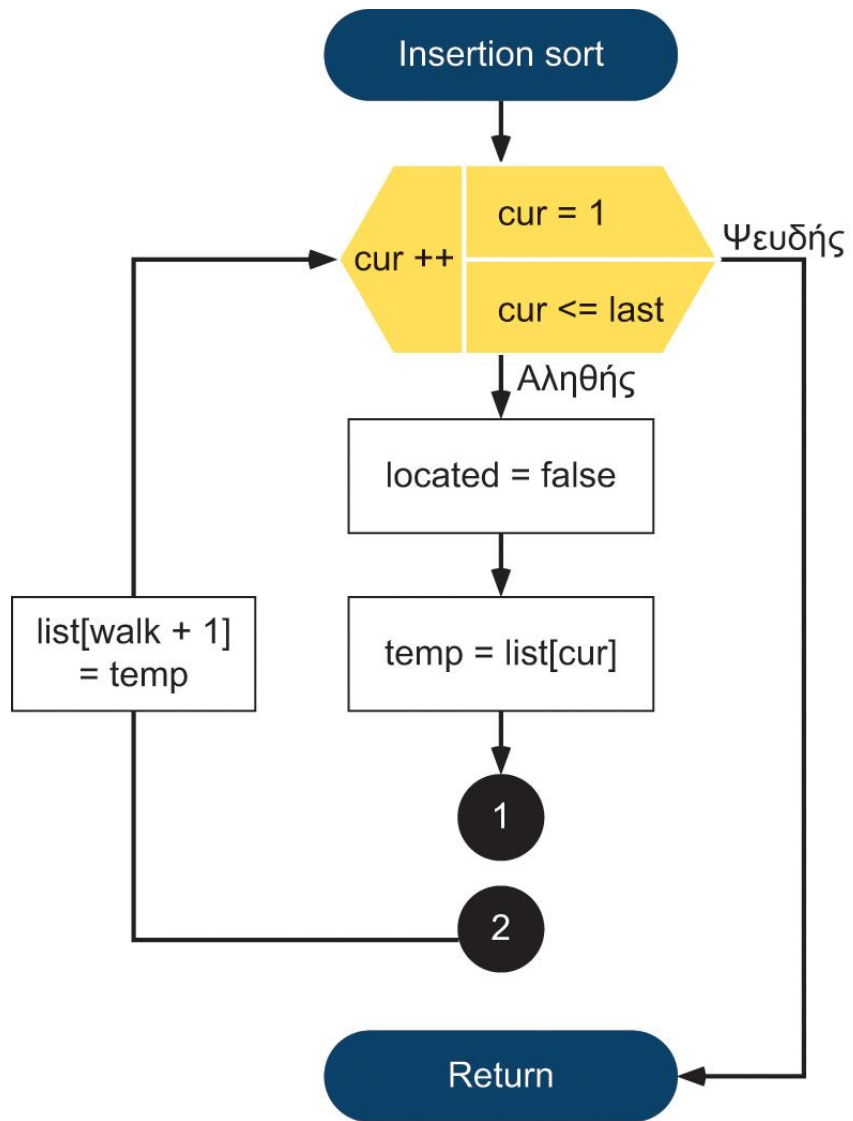
Βασική αρχή της ταξινόμησης με παρεμβολή.



Ταξινόμηση με παρεμβολή (Insertion Sort)



ΣΧΗΜΑ 8-25 Σχηματική αναπαράσταση της ταξινόμησης με παρεμβολή.



ΣΧΗΜΑ8-26 Αρχή λειτουργίας της ταξινόμησης με παρεμβολή.

Σύγκριση αλγορίθμων ταξινόμησης

ΠΙΝΑΚΑΣ 8-1 Εναλλαγές και μετατοπίσεις στοιχείων για τους τρεις αλγορίθμους ταξινόμησης

ΑΛΓΟΡΙΘΜΟΣ	ΕΝΑΛΛΑΓΕΣ	ΜΕΤΑΤΟΠΙΣΕΙΣ
Ταξινόμηση με επιλογή	Μία σε κάθε πέρασμα	
Ταξινόμηση φυσαλίδας	Πολλαπλές σε κάθε πέρασμα	
Ταξινόμηση με παρεμβολή	Μερικώς, εντός του βρόχου	Μηδέν ή περισσότερες σε κάθε πέρασμα

Σχόλια σχετικά με τους αλγορίθμους ταξινόμησης

- Οι τρεις αλγόριθμοι ταξινόμησης που περιγράψαμε παραπάνω αποτελούν τη βάση των βελτιωμένων και ταχύτερων μεθόδων ταξινόμησης που μελετώνται σε μαθήματα με αντικείμενο τις δομές δεδομένων
 - Η ταξινόμηση με επιλογή αποτελεί τη βάση του αλγορίθμου ταξινόμησης Heap Sort (ταξινόμηση με χρήση σωρού)
 - Η ταξινόμηση φουσαλίδας αποτελεί τη βάση για τους αλγορίθμους Quicksort και Quickersort
 - Η ταξινόμηση με παρεμβολή αποτελεί τη βάση του αλγορίθμου ταξινόμησης Shell Sort (ταξινόμηση παρεμβολής με φθίνοντα διαστήματα)

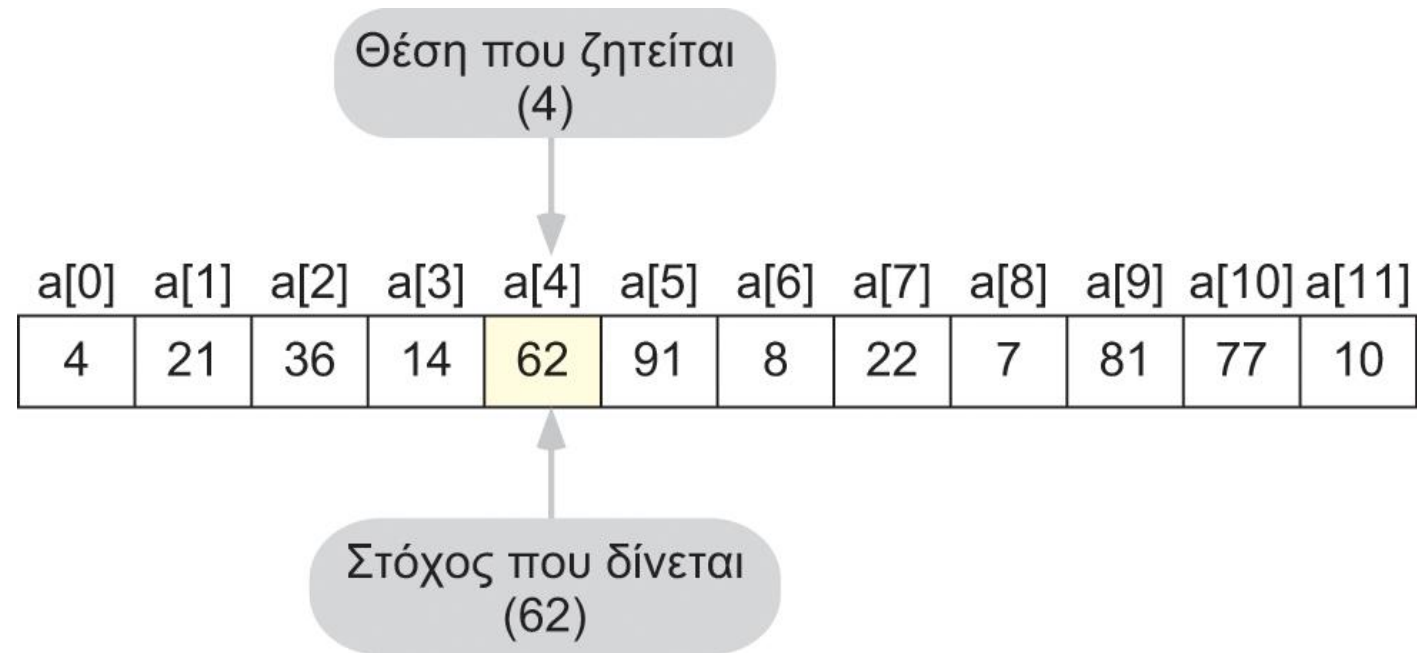
8.6

Αναζήτηση

Αναζήτηση

- Αναζήτηση (search): η διαδικασία εντοπισμού ενός στοιχείου (αντικειμένου, πεδίου, εγγραφής...) σε μία λίστα δεδομένων
- Οι αλγόριθμοι αναζήτησης συγκαταλέγονται μεταξύ των ευρύτερα χρησιμοποιούμενων και πλέον χρήσιμων στην επιστήμη των υπολογιστών

Αναζήτηση



ΣΧΗΜΑ 8-27

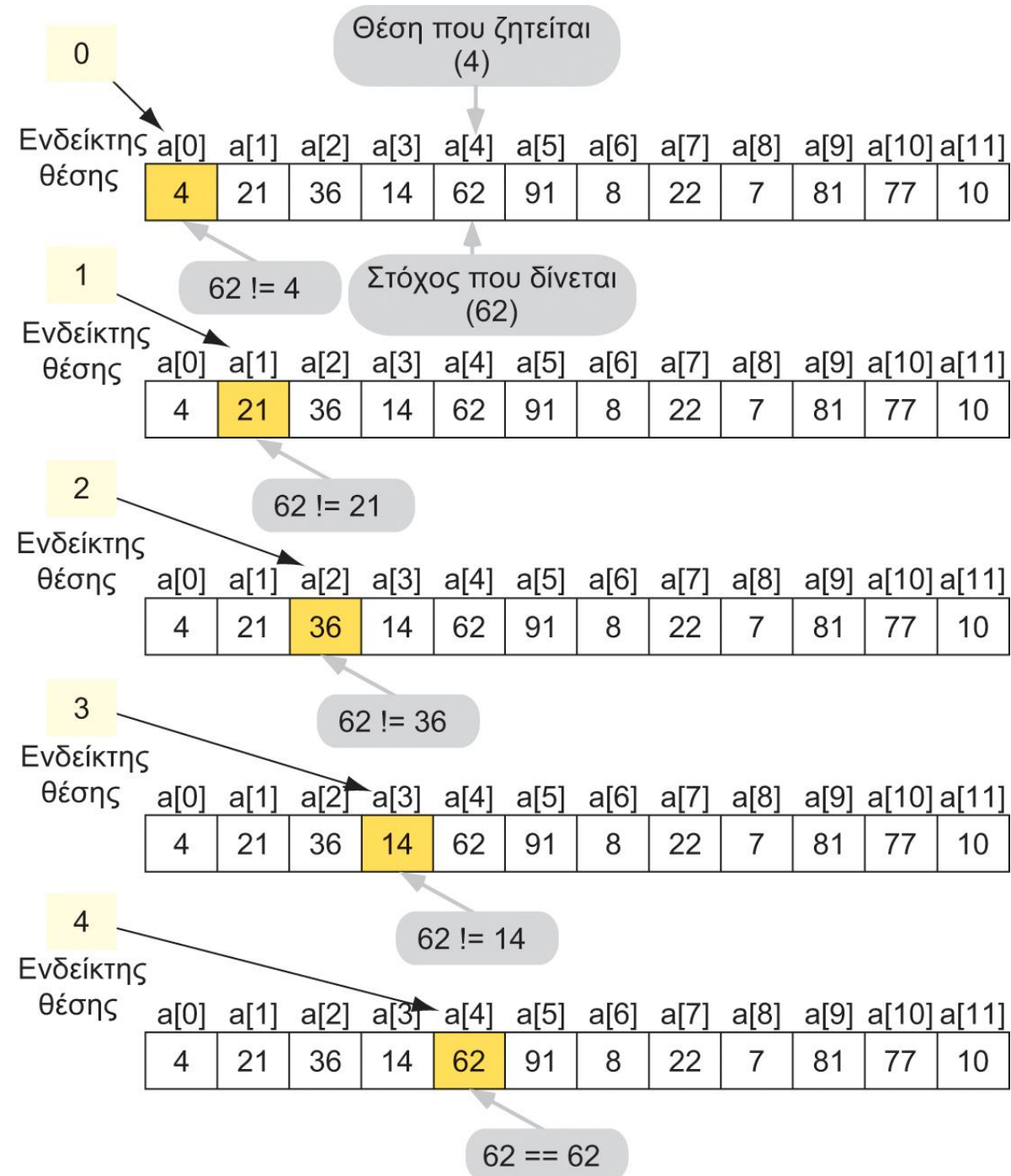
Σχηματική αναπαράσταση
της αναζήτησης σε πίνακα.

Σειριακή (γραμμική) αναζήτηση

- Η **σειριακή ή γραμμική αναζήτηση** (sequential search) χρησιμοποιείται σε περιπτώσεις όπου η λίστα δεδομένων δεν είναι ταξινομημένη
- Με την τεχνική της σειριακής-γραμμικής αναζήτησης, ξεκινάμε την αναζήτηση του στόχου από την αρχή της λίστας και συνεχίζουμε μέχρι να εντοπίσουμε το αναζητούμενο στοιχείο (τον στόχο), ή να βεβαιωθούμε ότι δεν υπάρχει στη λίστα

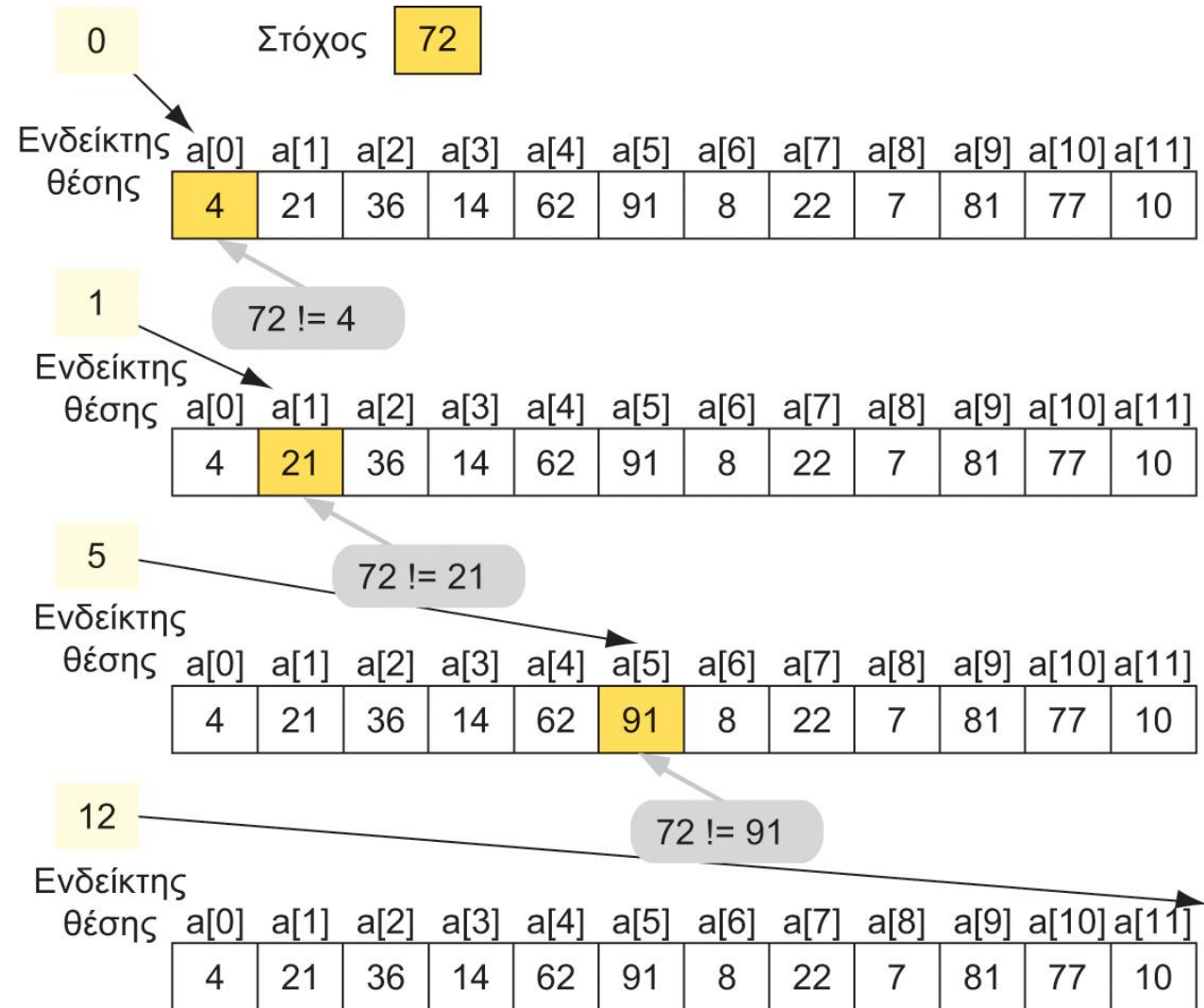
Σειριακή (γραμμική) αναζήτηση

ΣΧΗΜΑ 8-28 Αναζήτηση
σε μη ταξινομημένη λίστα.



Σειριακή (γραμμική) αναζήτηση

ΣΧΗΜΑ 8-29 Ανεπιτυχής
αναζήτηση σε μη
ταξινομημένη λίστα.

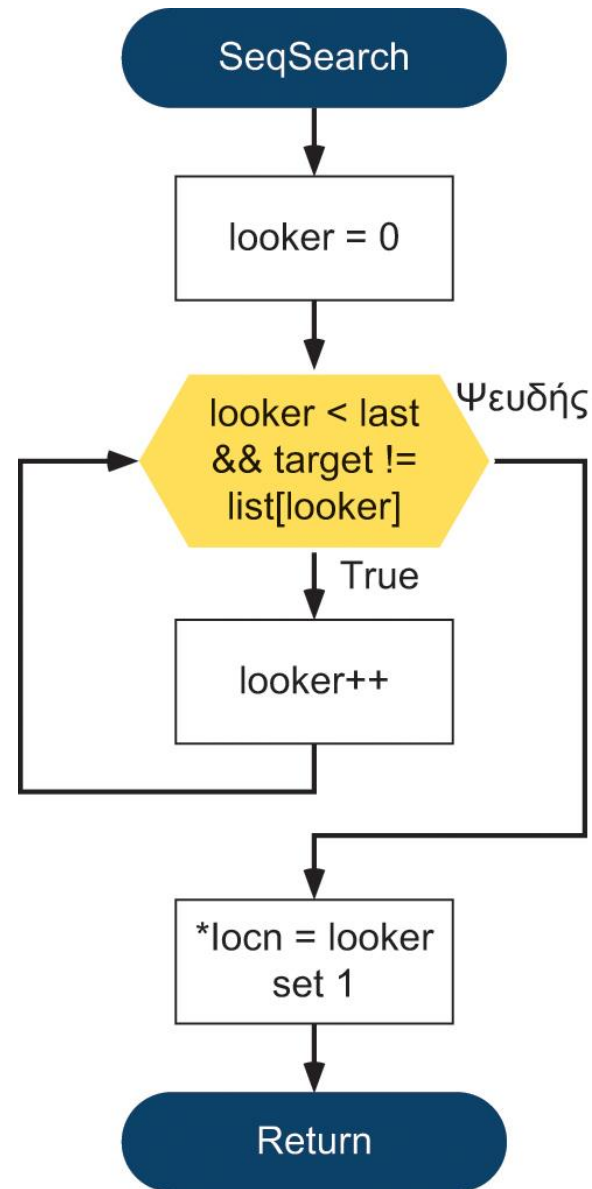


Σημείωση: Δεν παρουσιάζονται όλα τα σημεία ελέγχου.

Ενδείκτης εκτός
τέλους λίστας

Σειριακή (γραμμική) αναζήτηση

ΣΧΗΜΑ 8-30 Διάγραμμα
ροής για τη συνάρτηση
σειριακής αναζήτησης.

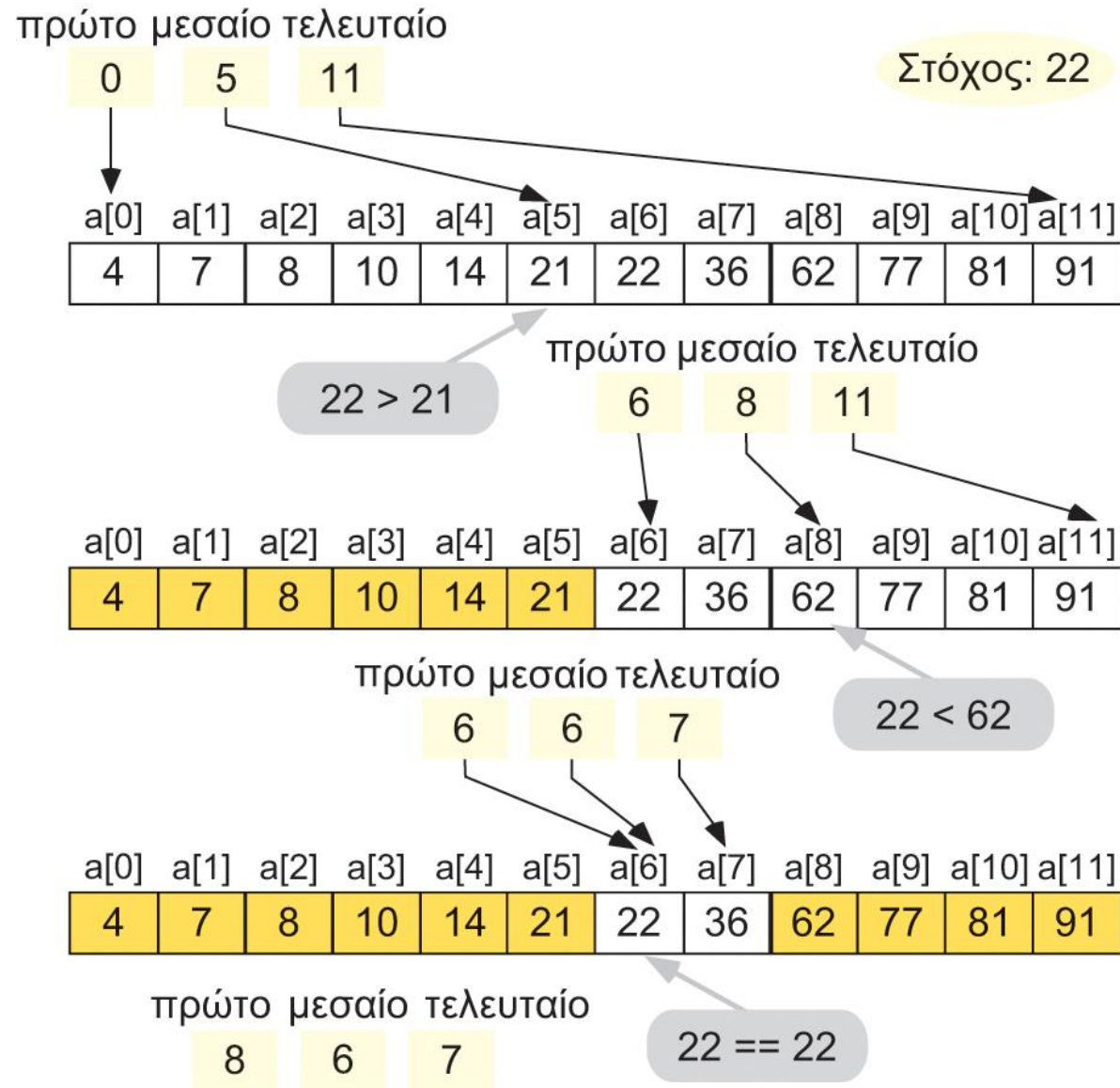


Δυαδική αναζήτηση

- Η δυαδική αναζήτηση (binary search) χρησιμοποιείται σε ταξινομημένες λίστες/πίνακες δεδομένων
- Υπερτερεί έναντι της γραμμής αναζήτησης σε εφαρμογές επεξεργασίας μεγάλων λιστών/πινάκων δεδομένων
 - προτείνουμε να χρησιμοποιείται οποτεδήποτε μία λίστα έχει πάνω από 50 στοιχεία

Δυαδική αναζήτηση

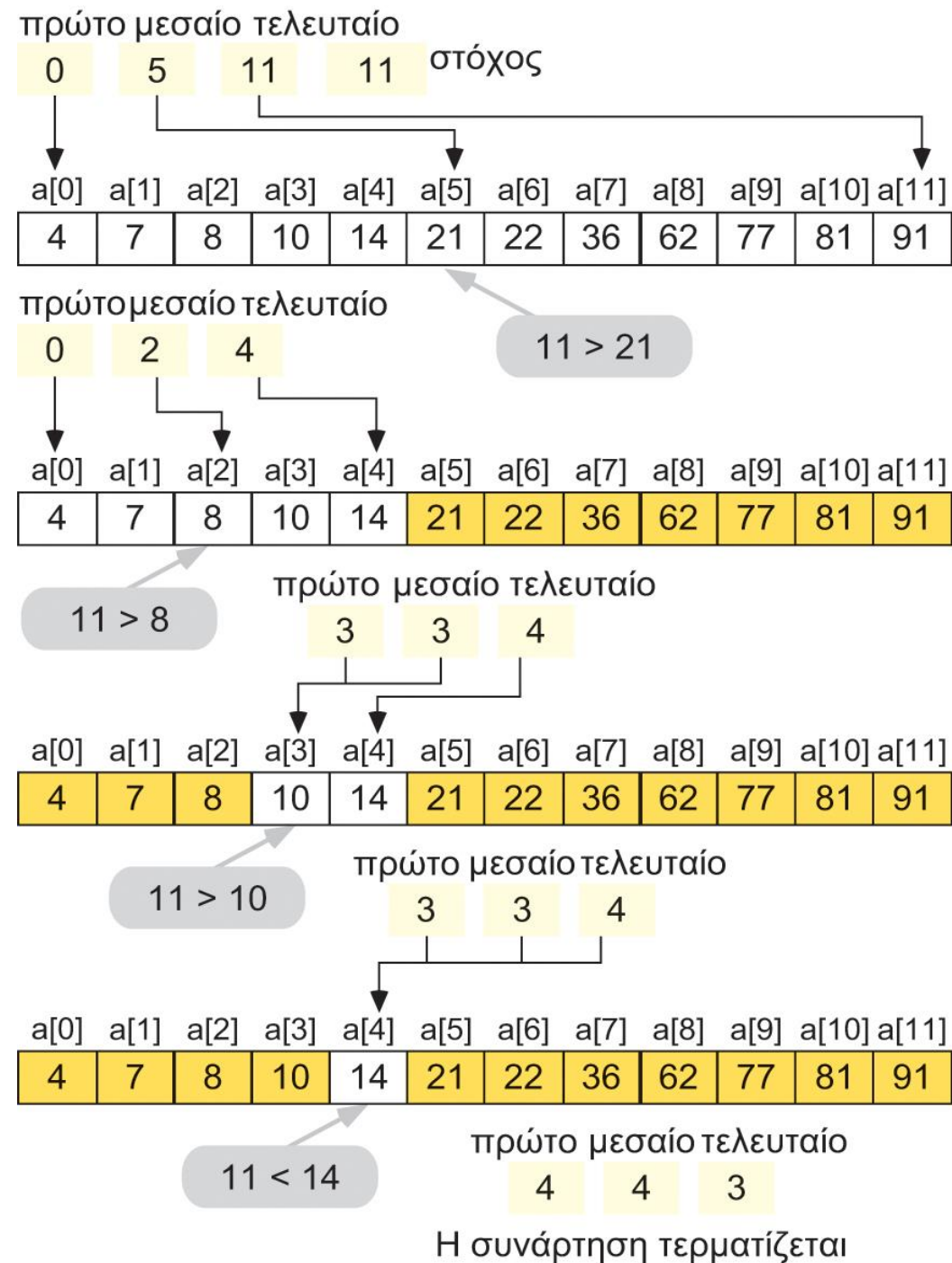
ΣΧΗΜΑ 8-31 Παράδειγμα
δυαδικής αναζήτησης.



Η συνάρτηση τερματίζεται

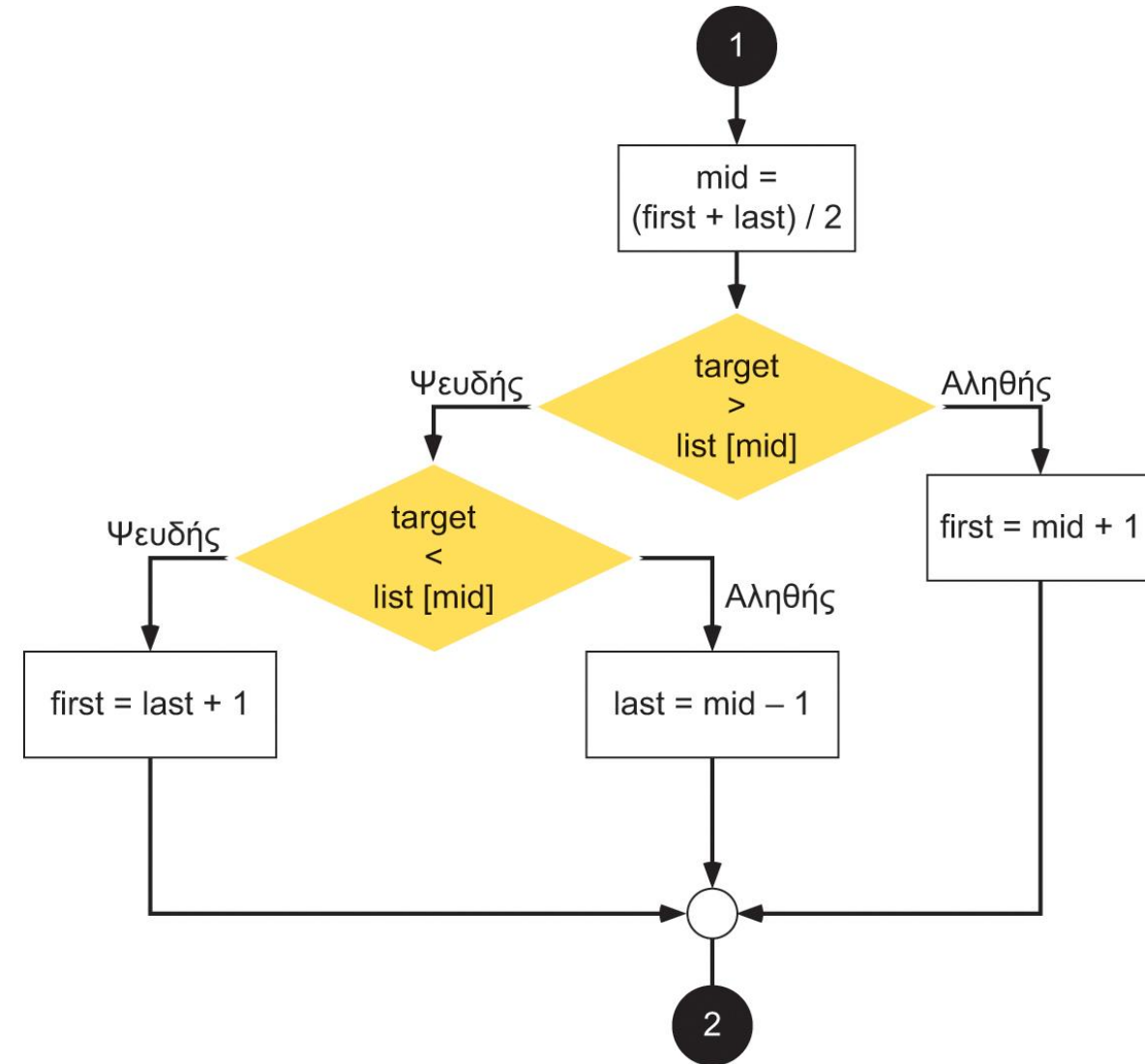
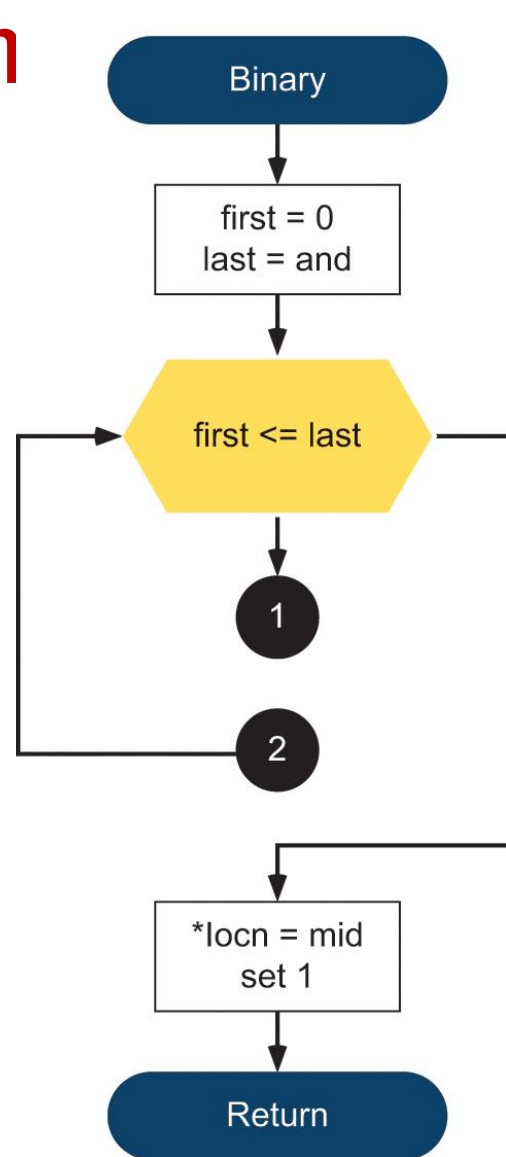
Διαδική αναζήτηση

ΣΧΗΜΑ 8-32 Παράδειγμα
ανεπιτυχούς διαδικής αναζήτησης.



Δυαδική αναζήτηση

ΣΧΗΜΑ 8-33 Διάγραμμα ροής για τον αλγόριθμο δυαδικής αναζήτησης



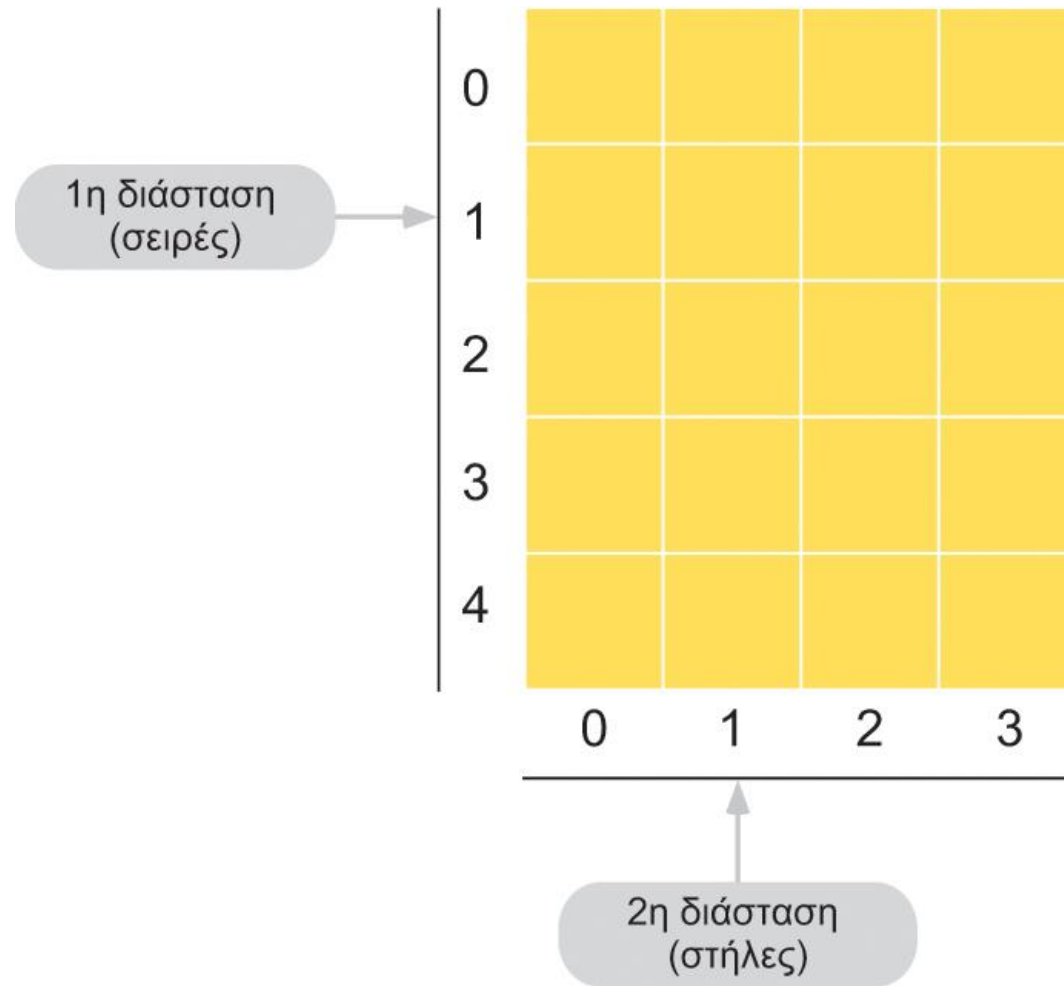
8.7

Διδιάστατοι πίνακες

Διδιάστατοι πίνακες

- Σε έναν μονοδιάστατο πίνακα (one-dimensional array) τα δεδομένα οργανώνονται γραμμικά, προς μία μόνο κατεύθυνση (μία διάσταση)
- Πολλές εφαρμογές απαιτούν την αποθήκευση δεδομένων σε περισσότερες από μία διαστάσεις
- Μία κοινή περίπτωση είναι οποιοσδήποτε τυπογραφικός πίνακας, δηλαδή οποιαδήποτε διάταξη αποτελούμενη από γραμμές και στήλες.
- Με όρους της C, μία τέτοια διάταξη είναι ένας πίνακας που εκτείνεται σε δύο διαστάσεις, δηλαδή ένας **διδιάστατος πίνακας** (two-dimensional array)

Διδιάστατοι πίνακες



ΣΧΗΜΑ 8-34 Διδιάστατος πίνακας.

Δήλωση διδιάστατων πινάκων

- Οι διδιάστατοι πίνακες, όπως και οι μονοδιάστατοι, πρέπει να δηλωθούν πριν χρησιμοποιηθούν
 - Η δήλωση πληροφορεί τον μεταγλωττιστή για το όνομα του πίνακα, τον τύπο κάθε στοιχείου και το μέγεθος κάθε διάστασης
 - Οι διδιάστατοι πίνακες μπορούν να είναι είτε σταθερού, είτε μεταβλητού μεγέθους
- Εάν δεν αρχικοποιήσουμε τον πίνακα με δεδομένα, τα περιεχόμενά του θα είναι ό,τι υπάρχει ήδη στη μνήμη που έχει δεσμευτεί γι' αυτόν—δηλαδή, απρόβλεπτα
 - Γενικά, οι πίνακες σταθερού μεγέθους πρέπει να αρχικοποιούνται

Διδιάστατοι πίνακες

ΣΧΗΜΑ 8-34 Σχηματική αναπαράσταση διδιάστατου πίνακα.



Διδιάστατος πίνακας με όνομα `table`

Επεξεργασία διδιάστατων πινάκων

- Ένας τρόπος εισαγωγής τιμών σε έναν πίνακα είναι με ανάγνωσή τους από το πληκτρολόγιο
 - Για έναν διδιάστατο πίνακα, αυτό απαιτεί συνήθως χρήση δύο βρόχων for, με τον δεύτερο ένθετο στον πρώτο
 - Ο ένας βρόχος ελέγχει την εισαγωγή τιμών στις γραμμές, ενώ ο άλλος ελέγχει την εισαγωγή τιμών στις στήλες

Επεξεργασία διδιάστατων πινάκων

- Μπορούμε να εκτυπώσουμε την τιμή όλων των στοιχείων ενός διδιάστατου πίνακα, ένα προς ένα, χρησιμοποιώντας δύο βρόχους, με τον δεύτερο ένθετο στον πρώτο
 - Ο πρώτος (εξωτερικός) βρόχος ελέγχει την εκτύπωση των γραμμών, ενώ ο δεύτερος (ένθετος) βρόχος ελέγχει την εκτύπωση των στηλών
 - Για μορφοποίηση της εκτύπωσης με διάταξη πίνακα (γραμμές και στήλες), χρησιμοποιούμε έναν χαρακτήρα νέας γραμμής ("`\n`") όταν ολοκληρώνεται η εκτύπωση κάθε γραμμής

Αρχικοποίηση διδιάστατων πινάκων

- Είναι δυνατό να αρχικοποιήσουμε μεμονωμένα στοιχεία ενός πίνακα χρησιμοποιώντας τον τελεστή ανάθεσης
 - Οι τιμές μπορούν να ανατεθούν στα στοιχεία του πίνακα χρησιμοποιώντας έναν αλγόριθμο

Αρχικοποίηση διδιάστατων πινάκων

- Οι αριθμητικοί ενδείκτες στον ορισμό ενός διδιάστατου πίνακα αντιπροσωπεύουν γραμμές και στήλες
 - Αυτή η μορφή αντιστοιχεί στον τρόπο με τον οποίο αποθηκεύονται τα δεδομένα στη μνήμη

00	01	02	03	04
10	11	12	13	14

Τι βλέπει ο χρήστης

σειρά 0					σειρά 1				
00	01	02	03	04	10	11	12	13	14
[0][0]	[0][1]	[0][2]	[0][3]	[0][4]	[1][0]	[1][1]	[1][2]	[1][3]	[1][4]

Τι περιέχεται στη μνήμη

ΣΧΗΜΑ 8-36 Διάταξη των δεδομένων ενός διδιάστατου πίνακα στη μνήμη.

Πέρασμα διδιάστατου πίνακα σε συνάρτηση

- Για διδιάστατους πίνακες, έχουμε στη διάθεσή μας τρεις επιλογές για το πέρασμα των περιεχομένων του πίνακα (ή ενός μέρους τους) σε μία συνάρτηση.
 - Μπορούμε να περάσουμε μεμονωμένα στοιχεία
 - Μπορούμε να περάσουμε μεμονωμένες (ολόκληρες) γραμμές του πίνακα
 - Μπορούμε να περάσουμε στη συνάρτηση ολόκληρο τον διδιάστατο πίνακα

```

#define MAX_ROWS 5
#define MAX_COLS 4
// Δηλώσεις συναρτήσεων
void print_square (int []);
int main (void)
{
    int table [MAX_ROWS] [MAX_COLS] =
        {
            { 0, 1, 2, 3},
            {10, 11, 12, 13},
            {20, 21, 22, 23},
            {30, 31, 32, 33},
            {40, 41, 42, 43}
        }; /* table */

    ...
    for (int row = 0; row < MAX_ROWS ; row++)
        print_square(table [row]);
    ...
    return 0;
} // τέλος της main

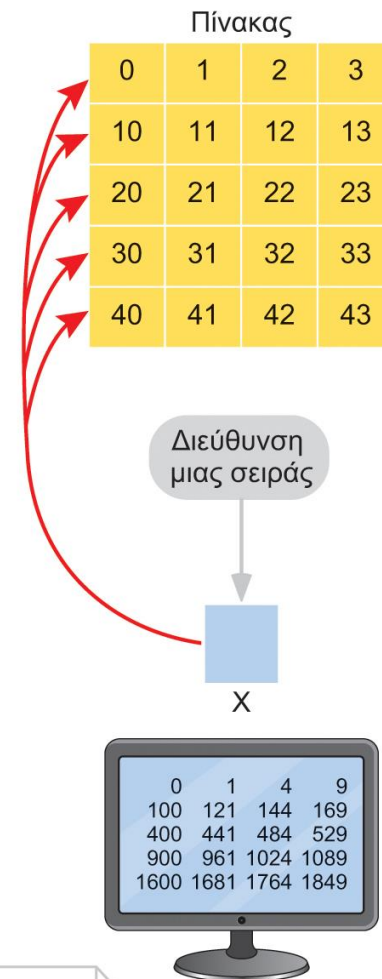
```

```

void print_square (int x[])
{
    for (int col = 0 ;col < MAX_COLS;
col++)
        printf ("%6d", x[col] * x[col]);
        printf ("\n ");

    return;
} //τέλος print_square

```



ΣΧΗΜΑ 8-37

Πέρασμα ολόκληρης γραμμής
ενός πίνακα σε συνάρτηση.

```

#define MAX_ROWS 5
#define MAX_COLS 4
// Δηλώσεις συναρτήσεων
double average (int [][][MAX_COLS]);

int main (void)
{
    double ave;
    int table[MAX_ROWS][MAX_COLS] =
        {
            { 0, 1, 2, 3},
            { 10, 11, 12, 13},
            { 20, 21, 22, 23},
            { 30, 31, 32, 33},
            { 40, 41, 42, 43}
        } ; // τέλος table

    ...
    ave = average(table);
    ...
    return 0;
} // τέλος της main

```

Πίνακας

0	1	2	3
10	11	12	13
20	21	22	23
30	31	32	33
40	41	42	43

Διεύθυνση
πίνακα



X



Άθροισμα

```

double average (int x[][MAX_COLS])
{
    double sum = 0;
    for (int i = 0; i < MAX_ROWS; i++)
        for (int j = 0; j < MAX_COLS; j++)
            sum += x[i][j];
    return (sum / (MAX_ROWS * MAX_COLS));
} // τέλος average

```

ΣΧΗΜΑ 8-38

Υπολογισμός μέσου όρου των
ακεραίων που αποθηκεύονται σε
έναν πίνακα.

0	1	1	1	1	1
-1	0	1	1	1	1
-1	-1	0	1	1	1
-1	-1	-1	0	1	1
-1	-1	-1	-1	0	1
-1	-1	-1	-1	-1	0

ΣΧΗΜΑ 8-38

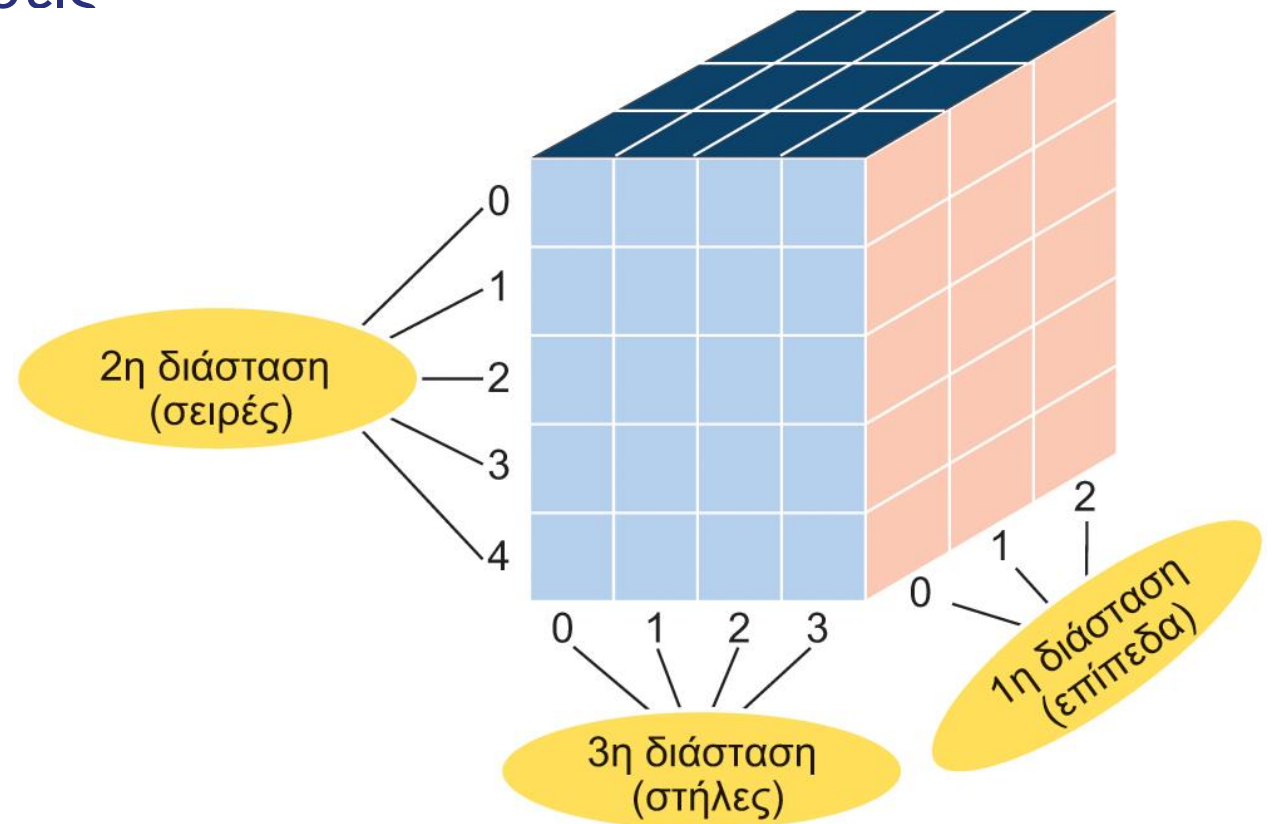
Παράδειγμα διδιάστατου πίνακα
«γεμάτου» με τιμές.

8.8

Πολυδιάστατοι πίνακες

Πολυδιάστατοι πίνακες

- Ένας πολυδιάστατος πίνακας (multidimensional array) μπορεί να έχει περισσότερες από δύο διαστάσεις
 - Η πρώτη διάσταση ονομάζεται επίπεδο (plane) και αποτελείται από γραμμές και στήλες

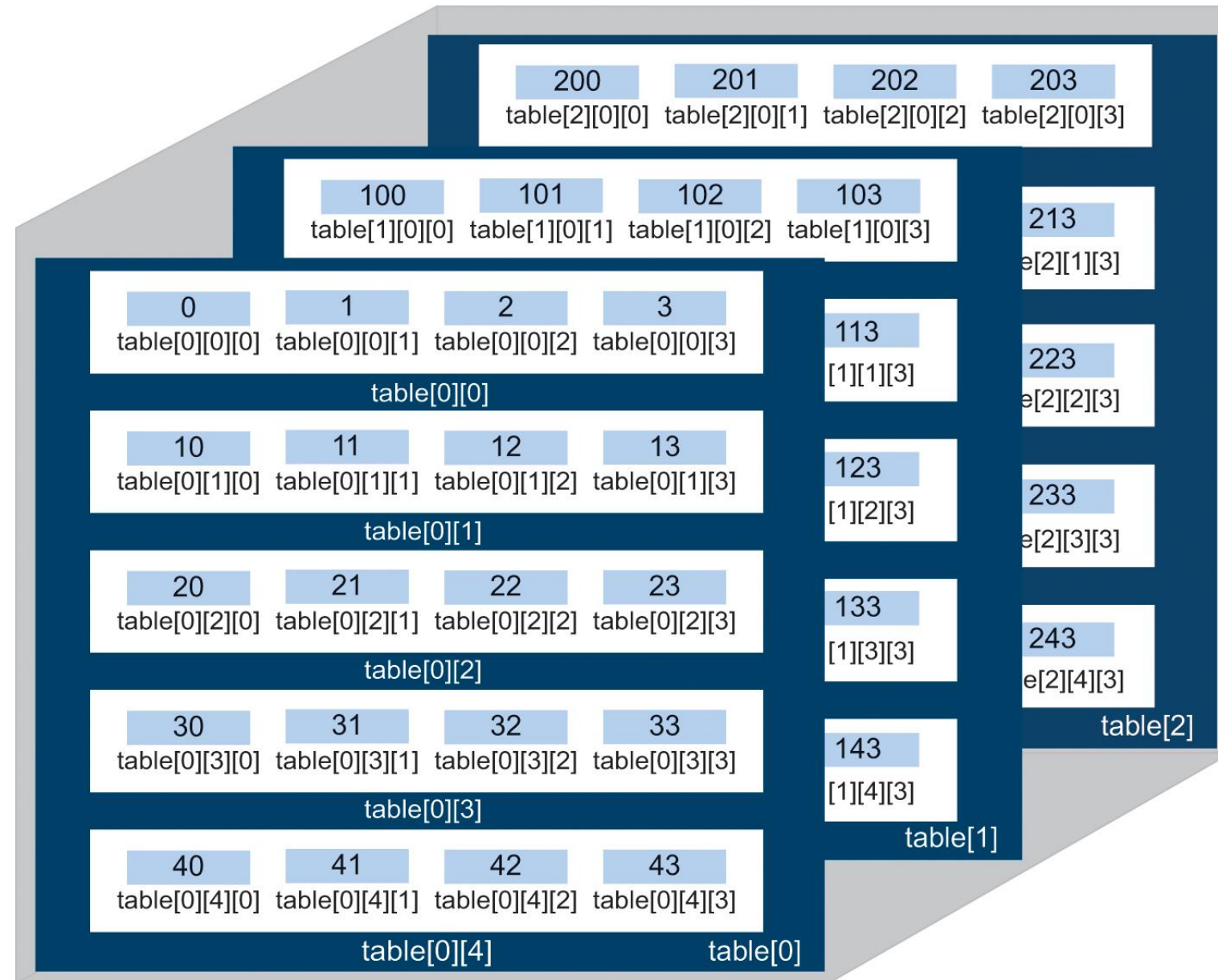


ΣΧΗΜΑ 8-40 Διάταξη τριδιάστατου πίνακα (3×5×4).

Πολυδιάστατοι πίνακες

Αν και στη C μπορούν να δημιουργηθούν και να χρησιμοποιηθούν πίνακες τεσσάρων ή περισσότερων διαστάσεων, είναι δύσκολο να απεικονιστούν σε ένα διδιάστατο μέσο όπως το χαρτί.

ΣΧΗΜΑ 8-41 Θεώρηση τριδιάστατου πίνακα στη C.



Δήλωση πολυδιάστατων πινάκων

- Οι πολυδιάστατοι πίνακες, όπως και οι μονοδιάστατοι, πρέπει να δηλωθούν πριν χρησιμοποιηθούν
 - Η δήλωση καθορίζει στον μεταγλωττιστή το όνομα του πίνακα, τον τύπο κάθε στοιχείου και το μέγεθος κάθε διάστασης
 - Για έναν πίνακα σταθερού μεγέθους, το μέγεθος είναι μία σταθερά η οποία πρέπει να έχει ήδη μία τιμή κατά τον χρόνο μεταγλώττισης

Αρχικοποίηση πολυδιάστατων πινάκων

- Η δήλωση και ο ορισμός δεν κάνουν τίποτα περισσότερο από το να δεσμεύουν χώρο για τα στοιχεία του πίνακα
 - Δεν αποθηκεύουν τιμές
- Εάν θέλουμε να αποθηκεύσουμε τιμές στον πίνακα, πρέπει είτε να αρχικοποιήσουμε τα στοιχεία του, είτε να διαβάσουμε τιμές από το πληκτρολόγιο, είτε να αναθέσουμε τιμές σε κάθε μεμονωμένο στοιχείο
 - Μόνο οι πολυδιάστατοι πίνακες σταθερού μεγέθους μπορούν να αρχικοποιούνται κατά τον ορισμό τους
 - Η αρχικοποίηση ενός πολυδιάστατου πίνακα είναι απλώς μία επέκταση της διαδικασίας αρχικοποίησης ενός διδιάστατου πίνακα

Σύνοψη

- Ένας μονοδιάστατος πίνακας είναι μία σταθερή ακολουθία στοιχείων (δεδομένων) του ίδιου τύπου
- Στη C χρησιμοποιούμε ενδείκτες θέσης για να υποδείξουμε τη θέση των στοιχείων ενός πίνακα
- Ένας πίνακας πρέπει να δηλωθεί και να οριστεί πριν χρησιμοποιηθεί. Η δήλωση και ο ορισμός καθορίζουν στον μεταγλωττιστή το όνομα του πίνακα, τον τύπο κάθε στοιχείου και το μέγεθος του πίνακα
- Η αρχικοποίηση όλων των στοιχείων ενός πίνακα σταθερού μεγέθους μπορεί να γίνει στην εντολή δήλωσής του
- Εάν ένας μονοδιάστατος πίνακας έχει αρχικοποιηθεί πλήρως κατά τη δήλωσή του, δεν είναι απαραίτητο να καθοριστεί το μέγεθός του, αλλά αυτό αποτελεί καλή πρακτική
- Όταν ένας πίνακας αρχικοποιείται εν μέρει, τα υπόλοιπα (μη αρχικοποιημένα) στοιχεία του τίθενται σε τιμή 0

Σύνοψη

- Μπορούμε να εισάγουμε δεδομένα στα στοιχεία ενός πίνακα χρησιμοποιώντας έναν βρόχο για να διαβάσουμε τα δεδομένα από το πληκτρολόγιο ή από ένα αρχείο
- Μπορούμε να προσπελάζουμε τα μεμονωμένα στοιχεία ενός πίνακα χρησιμοποιώντας το όνομα του πίνακα και τον ενδείκτη θέσης στοιχείου
- Μπορούμε να διαβάσουμε ή να γράψουμε δεδομένα στα στοιχεία ενός πίνακα χρησιμοποιώντας έναν βρόχο
- Μία αναφορά σε πίνακα είναι μία επιθεματική έκφραση, όπου οι τελεστές είναι η αριστερή και δεξιά αγκύλη ([])
- Η C δεν εκτελεί έλεγχο ορίων σε εντολές που αναφέρονται στα στοιχεία ενός πίνακα
- Μπορούμε να περάσουμε ένα μεμονωμένο στοιχείο ενός πίνακα σε μία συνάρτηση είτε διά τιμής, είτε δι' αναφοράς (διεύθυνση)

Σύνοψη

- Μπορούμε επίσης να περάσουμε έναν ολόκληρο πίνακα σε μία συνάρτηση. Σε αυτή την περίπτωση, μπορεί να περαστεί μόνο η διεύθυνση του πίνακα. Όταν συμβαίνει αυτό, η συνάρτηση έχει πρόσβαση σε όλα τα στοιχεία του πίνακα
- Ο όρος πίνακας συχνότητων αναφέρεται σε έναν πίνακα τα στοιχεία του οποίου υποδεικνύουν το πλήθος των εμφανίσεων των τιμών δεδομένων σε έναν άλλο πίνακα
- Το ιστόγραμμα είναι μία διαγραμματική αναπαράσταση ενός πίνακα συχνότητων
- Ένας διδιάστατος πίνακας της C αποτελεί μία αναπαράσταση ενός «τυπογραφικού» πίνακα με σειρές και στήλες
- Μπορούμε να περάσουμε ένα μεμονωμένο στοιχείο, μία γραμμή, ή ολόκληρο τον διδιάστατο πίνακα σε μία συνάρτηση
- Ένας πολυδιάστατος πίνακας είναι η επέκταση ενός διδιάστατου πίνακα σε τρεις, τέσσερις, ή περισσότερες διαστάσεις.

Σύνοψη

- Ένας πίνακας μπορεί να ταξινομηθεί με τη χρήση ενός αλγορίθμου ταξινόμησης
- Ο αλγόριθμος ταξινόμησης με επιλογή χωρίζει τον πίνακα σε δύο τμήματα, την ταξινομημένη και αταξινόμητη υπολίστα. Σε κάθε πέρασμα, ο αλγόριθμος επιλέγει το μικρότερο στοιχείο από την αταξινόμητη υπολίστα και το εναλλάσσει με το στοιχείο στην αρχή της αταξινόμητης υπολίστας
- Ο αλγόριθμος ταξινόμησης φουσαλίδας χωρίζει επίσης τον πίνακα σε δύο τμήματα, την ταξινομημένη και αταξινόμητη υπολίστα. Σε κάθε πέρασμα, ο αλγόριθμος μεταφέρει («ανεβάζει») το μικρότερο στοιχείο από την αταξινόμητη στην ταξινομημένη υπολίστα
- Ο αλγόριθμος ταξινόμησης με παρεμβολή χωρίζει τον πίνακα σε δύο τμήματα, την ταξινομημένη και αταξινόμητη υπολίστα. Σε κάθε πέρασμα, ο αλγόριθμος εισάγει (παρεμβάλλει) το πρώτο στοιχείο από την αταξινόμητη υπολίστα στην κατάλληλη θέση της ταξινομημένης υπολίστας

Σύνοψη

- Ο όρος αναζήτηση αναφέρεται στη διαδικασία εντοπισμού (εύρεσης της θέσης) ενός στοιχείου-στόχου σε μία λίστα δεδομένων
- Ο αλγόριθμος σειριακής αναζήτησης ξεκινά από την αρχή του πίνακα και διατρέχει τα στοιχεία του μέχρι να βρει τα αναζητούμενα δεδομένα, ή να φτάσει στο τέλος του πίνακα. Τα δεδομένα μπορεί να είναι ταξινομημένα ή όχι
- Ο αλγόριθμος δυαδικής αναζήτησης είναι πολύ ταχύτερος. Στη δυαδική αναζήτηση, κάθε έλεγχος αποκλείει τη μισή λίστα από περαιτέρω εξέταση. Σε αυτή την περίπτωση τα δεδομένα πρέπει να είναι ταξινομημένα
- Προκειμένου να περαστεί ένας ολόκληρος πίνακας μεταβλητού μεγέθους σε μία συνάρτηση, πρέπει επίσης να περαστεί και το μέγεθός του



Το παρόν συνοδευτικό έργο **παρέχεται** αποκλειστικά και μόνο στους διδάσκοντες που έχουν επιλέξει το αντίστοιχο βιβλίο μέσω του συστήματος του **Ευδόξου** ως διδακτικό σύγγραμμα για τη διδασκαλία των μαθημάτων τους και την αξιολόγηση της εκμάθησης των φοιτητών και για όσο χρονικό διάστημα διατηρείται η επιλογή του συγκεκριμένου συγγράμματος. Η **διάδοση, δημοσίευση, αναπαραγωγή ή πώληση** οπουδήποτε μέρους αυτού του έργου με οποιοδήποτε μέσο καταστρέφει την ακεραιότητα του έργου **και για σκοπούς διαφορετικούς από αυτούς για τους οποίους έχει χορηγηθεί η σχετική άδεια δεν επιτρέπεται**. Το περιεχόμενο του έργου **δεν θα πρέπει να διατίθεται** στους φοιτητές παρά μόνο όταν αυτό αναφέρεται ρητά ότι μπορεί να γίνει. Η **χρήση** του παρόντος έργου γίνεται αποκλειστικά από τον διδάσκοντα στα πλαίσια των εκπαιδευτικών καθηκόντων του και με τη χρήση των μέσων που αυτός διαχειρίζεται και έχει στη διάθεσή του από τις **επίσημες υπηρεσίες του εκπαιδευτικού ιδρύματος** όπου ανήκει, διασφαλίζοντας τη μη περαιτέρω διάδοση, δημοσίευση και αναπαραγωγή του έργου προς τρίτους. Ο διδάσκων δύναται να **προσαρμόζει** μέρος του υλικού των διαφανειών σύμφωνα με τις διδακτικές του ανάγκες, αναφερόμενος όμως πάντοτε στην αρχική πηγή αναφοράς. Το παρόν έργο προστατεύεται από την ελληνική και ευρωπαϊκή νομοθεσία περί πνευματικής ιδιοκτησίας καθώς και από τη νομοθεσία του κράτους όπου ανήκει η ξενόγλωσση πρωτότυπη έκδοση του έργου.