

Κεφάλαιο 9

Πίνακες

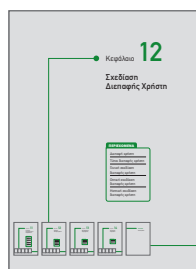
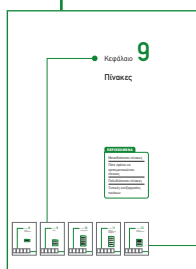
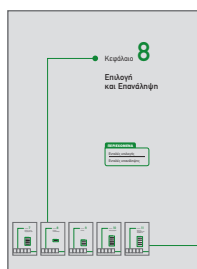
ΠΕΡΙΕΧΟΜΕΝΑ

Μονοδιάστατοι πίνακες

Πότε πρέπει να
χρησιμοποιούνται
πίνακες

Πολυδιάστατοι πίνακες

Τυπικές επεξεργασίες
πινάκων



9.1 | Μονοδιάστατοι πίνακες

Πολλά από τα προβλήματα τα οποία παρουσιάστηκαν στα προηγούμενα κεφάλαια, απλά επεξεργάζονται μία σειρά δεδομένων.

Διαβάζουν ένα δεδομένο κάθε φορά, το εκχωρούν σε μία μεταβλητή, εκτελούν τους αντίστοιχους υπολογισμούς και στη συνέχεια επαναλαμβάνεται η ίδια διαδικασία μέχρι να τελειώσουν όλα τα δεδομένα.

Για παράδειγμα ένα πρόγραμμα το οποίο διαβάζει τις θερμοκρασίες διαφόρων ημερών του μήνα, έστω 30, και υπολογίζει τη μέση θερμοκρασία, μπορεί πολύ απλά να γραφεί ως εξής:

```
...
Σύνολο <- 0
ΓΙΑ Ημέρα ΑΠΟ 1 ΜΕΧΡΙ 30
    ΔΙΑΒΑΣΕ Θερμοκρασία
    Σύνολο <- Σύνολο+θερμοκρασία
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
Μέση <- Σύνολο /30
... .
```

Χρησιμοποιώντας λοιπόν μόνο μία μεταβλητή, τη μεταβλητή *Θερμοκρασία*, το πρόβλημα λύνεται πολύ απλά και το αντίστοιχο πρόγραμμα είναι σύντομο και κατανοητό.

Αν όμως στο προηγούμενο πρόγραμμα ζητείται και ο αριθμός των ημερών που η θερμοκρασία ήταν κατώτερη της μέσης, τότε η σύγκριση αυτή πρέπει να γίνει μετά τον υπολογισμό της μέσης θερμοκρασίας. Αυτό σημαίνει ότι όλες οι θερμοκρασίες πρέπει να επανεισαχθούν για να συγκριθούν με τη μέση.

Μία άλλη λύση είναι να καταχωρηθεί κάθε θερμοκρασία σε διαφορετική μεταβλητή, έτσι ώστε κάθε τιμή που εισάγεται να διατηρείται στη μνήμη και να μπορεί να συγκριθεί με τη μέση, αφού αυτή υπολογιστεί. Τότε όμως πρέπει να δημιουργηθούν 30 διαφορετικές μεταβλητές, *Θερμοκρασία1*, *Θερμοκρασία2*,..., *Θερμοκρασία30*. Για να γραφεί το πρόγραμμα, χρειάζονται τριάντα εντολές ΔΙΑΒΑΣΕ και τριάντα εντολές ΑΝ.

Αν και αυτή η λύση είναι σωστή και πρακτική για μικρό αριθμό δεδομένων, προφανώς δεν εξυπηρετεί την επεξεργασία μεγάλου αριθμού δεδομένων.

Η καλύτερη λύση στο πρόβλημα αυτό είναι η χρήση μεταβλητής με δείκτες, έννοια που είναι γνωστή από τα μαθηματικά και υλοποιείται στον προγραμματισμό με τη δομή δεδομένων του **πίνακα**. Χρησιμοποιείται λοιπόν μόνο ένα όνομα *Θερμοκρασία*, που αναφέρεται και στις τριάντα διαφορετικές θερμοκρασίες.

Το όνομα του πίνακα καθορίζει μία ομάδα διαδοχικών θέσεων στη μνήμη. Κάθε συγκεκριμένη θέση μνήμης καλείται στοιχείο του πίνακα και προσδιορίζεται από την τιμή ενός δείκτη, όπως φαίνεται και στο σχήμα 9.1.

		MNΗΜΗ	
		.	
		.	
		.	
		26	Θερμοκρασία[1]
		27	Θερμοκρασία[2]
		.	.
		.	.
		.	.
		21	Θερμοκρασία[29]
		25	Θερμοκρασία[30]
		.	
		.	
		.	

Σχ. 9.1. Ο πίνακας Θερμοκρασία

Οι πίνακες που χρησιμοποιούν ένα μόνο δείκτη για την αναφορά των στοιχείων τους ονομάζονται **μονοδιάστατοι πίνακες**.

Το όνομα του πίνακα μπορεί να είναι οποιοδήποτε δεκτό όνομα της **ΓΛΩΣΣΑΣ** και ο δείκτης είναι μία ακέραια έκφραση, σταθερή ή μεταβλητή που περικλείεται μέσα στα σύμβολα [και]. Το στοιχείο *Θερμοκρασία[2]* εκφράζει τη θερμοκρασία της δεύτερης ημέρας, αναφέρεται στο δεύτερο στοιχείο του πίνακα *Θερμοκρασία* και έχει την τιμή 27.

Γενικότερα το στοιχείο *Θερμοκρασία[i]* αναφέρεται στο i-στό στοιχείο του πίνακα.

Κάθε πίνακας πρέπει υποχρεωτικά να περιέχει δεδομένα του ίδιου τύπου, δηλαδή ακέραια, πραγματικά, λογικά ή αλφαριθμητικά. Ο τύπος του πίνακα δηλώνεται μαζί με τις άλλες μεταβλητές του προγράμματος στο τμήμα δήλωσης μεταβλητών. Εκτός από τον τύπο του πίνακα πρέπει να δηλώνεται και ο αριθμός των στοιχείων που περιέχει ή καλύτερα ο μεγαλύτερος αριθμός στοιχείων που μπορεί να έχει ο συγκεκριμένος πίνακας και αυτό για να δεσμευτούν οι αντίστοιχες συνεχόμενες θέσεις μνήμης.

Για παράδειγμα

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Θερμοκρασία [30]

ΟΡΙΣΜΟΣ

Πίνακας είναι ένα σύνολο αντικειμένων ίδιου τύπου, τα οποία αναφέρονται με ένα κοινό όνομα. Κάθε ένα από τα αντικείμενα που απαρτίζουν τον πίνακα λέγεται **στοιχείο** του πίνακα. Η αναφορά σε ατομικά στοιχεία του πίνακα γίνεται με το όνομα του πίνακα ακολουθούμενο από ένα δείκτη.

Ο δείκτης είναι μία μεταβλητή που μπορεί να έχει οποιοδήποτε δεκτό όνομα. Είναι σύνηθες όμως στον προγραμματισμό ως δείκτες να χρησιμοποιούνται οι μεταβλητές i, j, k.

9.2 | Πότε πρέπει να χρησιμοποιούνται πίνακες

Η χρήση πινάκων είναι ένας βολικός τρόπος για τη διαχείριση πολλών δεδομένων ιδίου τύπου, αλλά συχνά η χρήση τους είναι περιττή και επιζήμια στην ανάπτυξη του προγράμματος.

Πέρα από τα πλεονεκτήματα που αναφέρθηκαν, υπάρχουν και δύο μειονεκτήματα από τη χρήση πινάκων.

Οι πίνακες απαιτούν μνήμη. Κάθε πίνακας δεσμεύει από την αρχή του προγράμματος πολλές θέσεις μνήμης. Σε ένα μεγάλο και σύνθετο πρόγραμμα η άσκοπη χρήση μεγάλων πινάκων μπορεί να οδηγήσει ακόμη και σε αδυναμία εκτέλεσης του προγράμματος.

Οι πίνακες περιορίζουν τις δυνατότητες του προγράμματος. Στο προηγούμενο πρόγραμμα του υπολογισμού των στατιστικών μεγεθών, υπάρχει ανώτατο όριο στο πλήθος των αριθμών ίσο με 100. Αυτό γιατί οι πίνακες είναι στατικές δομές και το μέγεθός τους πρέπει να δηλώνεται στην αρχή του προγράμματος, ενώ παραμένει υποχρεωτικά σταθερό κατά την εκτέλεση του προγράμματος.

Η απόφαση για τη χρήση ή όχι πίνακα για τη διαχείριση των δεδομένων είναι κυρίως θέμα εμπειρίας στον προγραμματισμό.

Γενικά, αν τα δεδομένα που εισάγονται σε ένα πρόγραμμα πρέπει να διατηρούνται στη μνήμη μέχρι το τέλος της εκτέλεσης, τότε η χρήση πινάκων βοηθάει ή συχνά είναι απαραίτητη για την επίλυση του προβλήματος. Σε άλλη περίπτωση μπορεί να αποφεύγεται η χρήση τους.

Ας επιστρέψουμε στο προηγούμενο παράδειγμα.

Για τον υπολογισμό της μέσης τιμής και της τυπικής απόκλισης δεν είναι απαραίτητο να διατηρούνται οι τιμές στη μνήμη. Το πρόγραμμα θα δούλευε το ίδιο καλά και χωρίς τη χρήση πινάκων, αλλά με τη χρήση μίας και μόνο μεταβλητής. Ο υπολογισμός όμως της διάμεσης τιμής, που προϋποθέτει την ταξινόμηση των δεδομένων, απαιτεί τη χρήση πίνακα. Αν λοιπόν το πρόβλημα απαιτούσε μόνο τον υπολογισμό του μέσου όρου και της τυπικής απόκλισης, θα ήταν προτιμότερη μία λύση χωρίς τη χρήση πινάκων.

9.3 | Πολυδιάστατοι πίνακες

Στο προηγούμενο παράδειγμα υπήρχαν 30 τιμές της θερμοκρασίας, μία για κάθε ημέρα του μήνα, και για την επίλυσή του χρησιμοποιήθηκε ένας πίνακας 30 θέσεων, ο πίνακας *Θερμοκρασία*. Ο πίνακας αυτός περιέχει τις τιμές της θερμοκρασίας για κάθε ημέρα μίας πόλης και ο δείκτης δείχνει την ημέρα.

Έστω ότι οι θερμοκρασίες δίνονται από τον παρακάτω πίνακα.

ΗΜΕΡΑ	ΠΟΛΗ			
	1	2	...	10
1	25	21	...	32
2	26	22	...	31
...	...	...	...	...
30	27	23	...	30

Προγραμματιστικό περιβάλλον Pascal

```
PROGRAM poleis;
VAR
    temperature: array[1..30,1..10] OF REAL;
    ave:array [1..10] OF REAL;
    total:REAL;
    i,j,days:INTEGER;
BEGIN
    FOR i:=1 TO 30 DO
        FOR j:=1 TO 10 DO
            BEGIN
                write ('Δώσε τη θερμοκρασία :',i,j,' ');
                readln (temperature [i,j])
            END;
        FOR j:=1 TO 10 DO
            BEGIN
                total:=0;
                FOR i:=1 TO 30 DO
                    total:=total+temperature[i,j];
                ave[j]:=total/30
            END;
        FOR j:=1 TO 10 DO
            BEGIN
                days:=0;
                FOR i:=1 TO 30 DO
                    IF temperature[i,j] < ave[j] then
                        days:=days+1;
                writeln ('Μέση θερμοκρασία ',i,' πόλης
                        ',ave[i]:4:1);
                writeln ('Ημέρες με μικρότερη θερμοκρασία :',
                        days);
            END
        END.
END.
```

Εκτός από μονοδιάστατους και δισδιάστατους πίνακες υπάρχουν πίνακες με περισσότερες διαστάσεις, τρισδιάστατοι, τετραδιάστατοι και γενικά πολυδιάστατοι, ανάλογα με τον αριθμό των δεικτών που χρησιμοποιούνται για τον καθορισμό των στοιχείων. Ωστόσο τα περισσότερα προβλήματα αντιμετωπίζονται με τη χρήση πινάκων μονοδιάστατων ή δισδιάστατων.

Στο προηγούμενο παράδειγμα έστω ότι έχουμε θερμοκρασίες για κάθε πόλη, για κάθε ημέρα αλλά και για κάθε έτος. Τότε κάθε στοιχείο χρειάζεται τρεις δείκτες. Έναν που θα δείχνει το έτος, έναν που θα δείχνει την πόλη και έναν που θα δείχνει την ημέρα. Για παράδειγμα, η θερμοκρασία της πρώτης ημέρας, της πρώτης πόλης και του πρώτου έτους είναι 25 και είναι το στοιχείο *Θερμοκρασία [1, 1, 1]*.

```

BEGIN
  FOR i:=1 TO 10 DO
    BEGIN
      write ('Δώσε το όνομα : ');
      readln (name[i]);
      FOR j:=1 TO 7 DO
        BEGIN
          write ('Δώσε την ', j, ' η είσπραξη : ');
          readln(tickets[i,j]);
        END;
      END;
    END;
    max:=0;
    FOR i:=1 TO 10 DO
      BEGIN
        sum:=0;
        FOR j:=1 TO 7 DO
          sum:=sum+tickets[i,j];
        writeln('Σύνολο ', name[i], ' = ', sum);
        IF max<sum THEN
          BEGIN
            max:=sum;
            k:=i;
          END;
        END;
      END;
      writeln ('Μέγιστο ', max, ' στον ', name[k]);

      max:=0;
      FOR j:=1 TO 7 DO
        BEGIN
          sum:=0;
          FOR i:=1 TO 10 DO
            sum:=sum+tickets[i,j];
          writeln('Σύνολο ', j, ' ημέρας = ', sum);
          IF max<sum THEN
            BEGIN
              max:=sum;
              k:=j;
            END;
          END;
        END;
      END;
      writeln ('Μέγιστο ', max, ' την ', k);
    END
  END

```

9.4 | Τυπικές επεξεργασίες πινάκων

Τα προγράμματα τα οποία χρησιμοποιούν πίνακες πολύ συχνά απαιτούν συγκεκριμένες επεξεργασίες στα στοιχεία του πίνακα. Οι τυπικές αυτές επεξεργασίες είναι:

Υπολογισμός αθροισμάτων στοιχείων του πίνακα.

Εύρεση του μέγιστου ή του ελάχιστου στοιχείου.

Ταξινόμηση των στοιχείων του πίνακα.

Αναζήτηση ενός στοιχείου του πίνακα.

Συγχώνευση δύο πινάκων.

Μερικές από αυτές τις επεξεργασίες παρουσιάστηκαν ήδη στα παραδείγματα αυτού του κεφαλαίου. Για τις διαδικασίες της ταξινόμησης και της αναζήτησης έχουν παρουσιαστεί μερικοί αλγόριθμοι, οι οποίοι αναπτύχθηκαν στο αντίστοιχο κεφάλαιο των αλγορίθμων, όπου συζητήθηκαν τα πλεονεκτήματα και τα μειονεκτήματά τους.

Συνοπτικά έχουμε:

Υπολογισμός αθροισμάτων στοιχείων του πίνακα

Πολύ συχνά απαιτείται ο υπολογισμός του αθροίσματος στοιχείων του πίνακα που έχουν κοινά χαρακτηριστικά, για παράδειγμα βρίσκονται στην ίδια στήλη ή στην ίδια γραμμή. Τα παραδείγματα που παρουσιάστηκαν σε αυτό το κεφάλαιο απαιτούσαν τον υπολογισμό των αθροισμάτων των στοιχείων του πίνακα τόσο ανά γραμμή όσο και ανά στήλη.

Εύρεση του μέγιστου ή του ελάχιστου στοιχείου

Η εύρεση του μέγιστου ή του ελάχιστου στοιχείου ενός πίνακα παρουσιάστηκε στα προηγούμενα παραδείγματα. Αν ο πίνακας δεν είναι ταξινομημένος, τότε πρέπει να συγκριθούν τα στοιχεία ένα προς ένα, για να βρεθεί το μέγιστο ή το ελάχιστο. Αν ο πίνακας είναι ταξινομημένος, τότε προφανώς το μέγιστο και το ελάχιστο βρίσκονται στα δύο ακριανά στοιχεία του πίνακα.

Ταξινόμηση των στοιχείων του πίνακα

Στο κεφάλαιο 3 αναφέρθηκε η μέθοδος ταξινόμησης της ευθείας ανταλλαγής, η οποία χρησιμοποιήθηκε και στο παράδειγμα 2.

Η μέθοδος αυτή είναι από τις απλούστερες αλλά δεν είναι η πιο αποδοτική. Υπάρχουν πολλές άλλες μέθοδοι ταξινόμησης καθώς και παραλλαγές αυτών.

Η επιλογή του καλύτερου αλγορίθμου εξαρτάται κυρίως από το πλήθος των στοιχείων του πίνακα και την αρχική τους διάταξη, αν δηλαδή ο πίνακας είναι τελείως αταξινομήτος ή μερικώς ταξινομημένος.

Αναζήτηση ενός στοιχείου του πίνακα

Δύο είναι οι πλέον διαδεδομένοι αλγόριθμοι αναζήτησης:

Η σειριακή αναζήτηση

Η δυαδική αναζήτηση

Η σειριακή μέθοδος αναζήτησης είναι η πιο απλή, αλλά και η λιγότερο αποτελεσματική μέθοδος. Χρησιμοποιείται όμως υποχρεωτικά για πίνακες που δεν είναι ταξινομημένοι. Αντίθετα η δυαδική αναζήτηση χρησιμοποιείται μόνο σε ταξινομημένους πίνακες και είναι σαφώς αποδοτικότερη από τη σειριακή μέθοδο.

Συγχώνευση δύο πινάκων

Η συγχώνευση είναι μία από τις βασικές λειτουργίες σε πίνακες.

Σκοπός της είναι η δημιουργία από τα στοιχεία δύο (ή περισσότερων) ταξινομημένων πινάκων ενός άλλου, που είναι και αυτός ταξινομημένος.

1. Τι είναι πίνακας (ορισμός)

Πίνακας είναι ένα σύνολο αντικειμένων ίδιου τύπου, τα οποία αναφέρονται με κοινό όνομα. Κάθε ένα από τα αντικείμενα αυτά λέγεται **στοιχείο** του πίνακα. Η αναφορά σε ατομικά στοιχεία του πίνακα γίνεται με το όνομα του πίνακα ακολουθούμενο από έναν ή περισσότερους δείκτες.

2. Τι καθορίζει το όνομα ενός πίνακα;

Το όνομα ενός πίνακα καθορίζει μια σειρά **διαδοχικών** θέσεων στη μνήμη.

3. Πότε ένας πίνακας λέγεται (είναι) μονοδιάστατος;

Ένας πίνακας είναι μονοδιάστατος όταν για την αναφορά (ή προσπέλαση) των στοιχείων του αρκεί ένας δείκτης.

4. Τι είναι οι δείκτες που χρησιμοποιούνται για την αναφορά σε συγκεκριμένα στοιχεία ενός πίνακα;

Οι δείκτες μπορεί να είναι ακέραιες εκφράσεις, σταθερές ή μεταβλητές.

5 Μπορεί ένας πίνακας να φιλοξενεί ταυτόχρονα δεδομένα διαφορετικών τύπων;

Όχι. Οι πίνακες είναι δομές δεδομένων που περιέχουν στοιχεία ίδιου τύπου.

6. Τι καθορίζει τις διαστάσεις ενός πίνακα;

Οι διαστάσεις ενός πίνακα καθορίζονται από το πλήθος των δεικτών που απαιτούνται για την πρόσβαση των στοιχείων του.

7. Πότε επιβάλλεται να χρησιμοποιούνται πίνακες μεταξύ των μεταβλητών ενός προγράμματος;

Οι πίνακες είναι γενικά ένας βολικός τρόπος να διαχειριζόμαστε πολλά στοιχεία ίδιου τύπου. Αυτό ωστόσο δεν σημαίνει ότι πρέπει να χρησιμοποιούμε πίνακες κάθε φορά. Η χρήση πίνακα επιβάλλεται (είναι αναγκαία) όταν πρέπει τα δεδομένα να διατηρηθούν μέχρι το τέλος της εκτέλεσης του προγράμματος.

8. Ποια είναι το/τα πλεονεκτήματα της χρήσης πινάκων;

Οι πίνακες είναι ένας βολικός τρόπος για να τον προγραμματιστή όταν θέλει να διαχειριστεί πολλά δεδομένα ίδιου τύπου.

9. Ποια είναι το/τα μειονεκτήματα της χρήσης πινάκων;

(α) **Ασκοπη χρήση μνήμης:** συχνά δηλώνουμε μεγάλους πίνακες σε ένα πρόγραμμα. Αυτό όμως μπορεί να οδηγήσει σε προβλήματα κατά την εκτέλεση του προγράμματος.

(β) **Περιορισμός των δυνατοτήτων του προγράμματος:** το μέγεθος των πινάκων πρέπει να δηλώνεται από την φάση του προγραμματισμού και παραμένει υποχρεωτικά σταθερό κατά την διάρκεια εκτέλεσης του προγράμματος.

Τα δύο αυτά μειονεκτήματα της χρήσης πινάκων οφείλονται στο γεγονός ότι οι πίνακες είναι στατικές δομές.

10. Ποιες είναι οι τυπικές επεξεργασίες πινάκων.

(α) **υπολογισμός αθροισμάτων** (ανά γραμμή, ανά στήλη, συνολικά, ή με βάση κάποιο κριτήριο)

(β) **εύρεση μέγιστου ή ελάχιστου στοιχείου**

(γ) **ταξινόμηση**

(δ) **αναζήτηση** (σειριακή ή δυαδική, η δυαδική όμως απαιτεί ταξινομημένο πίνακα)

(ε) **συγχώνευση** (είναι η συνένωση δύο ή περισσότερων ταξινομημένων πινάκων ώστε να προκύψει ένας ενιαίος ταξινομημένος πίνακας)

11. (κ) Σε μια εντολή ΓΙΑ μέσα στην οποία επεξεργαζόμαστε διάφορους πίνακες, επιτρέπεται οι τιμές ΑΠΟ, ΜΕΧΡΙ και ΜΕ_ΒΗΜΑ να είναι πραγματικοί αριθμοί;

Αν έστω μία από τις τιμές ΑΠΟ, ΜΕΧΡΙ, ΜΕ_ΒΗΜΑ είναι πραγματικός αριθμός, τότε ο μετρητής της ΓΙΑ δεν μπορεί να χρησιμοποιηθεί σαν δείκτης σε κάποιο πίνακα. Όμως, μέσα σε αυτή τη ΓΙΑ μπορεί να γίνει επεξεργασία πινάκων αρκεί να χρησιμοποιούμε άλλες μεταβλητές για δείκτες στους πίνακες.

12. (κ) Να γίνει αντιστοίχιση ανάμεσα στις τυπικές επεξεργασίες πινάκων και στις βασικές λειτουργίες των δομών δεδομένων.

Τυπικές Επεξεργασίες Πινάκων	Βασικές Λειτουργίες Δομών Δεδομένων
Υπολογισμός αθροισμάτων	Προσπέλαση
Εύρεση μέγιστου, ελάχιστου	Προσπέλαση
Ταξινόμηση	Ταξινόμηση
Αναζήτηση	Αναζήτηση
Συγχώνευση	Συγχώνευση