

9.

Πίνακες



Εισαγωγή

Η χρήση των μεταβλητών με δείκτες στην άλγεβρα είναι ένας ιδιαίτερα δυναμικός τρόπος για τη διαχείριση μεγάλου αριθμού δεδομένων ιδίου τύπου. Οι γλώσσες προγραμματισμού, δανείζονται την έννοια των μεταβλητών με δείκτες και χρησιμοποιούν τους πίνακες για τον ίδιο λόγο.

Στο κεφάλαιο αυτό παρουσιάζονται οι έννοιες του πίνακα καθώς και οι βασικοί τρόποι επεξεργασίας τους από τη **ΓΛΩΣΣΑ**.

Παρουσιάζονται οι μονοδιάστατοι καθώς και οι πολυδιάστατοι πίνακες, ο τρόπος με τον οποίο ορίζονται και χρησιμοποιούνται και τέλος συζητούνται οι πλέον κοινές διαδικασίες πάνω σε πίνακες, η εύρεση μεγίστου και ελαχίστου, η αναζήτηση, η ταξινόμηση και η συγχώνευση πινάκων.



Στόχοι

Να είναι σε θέση ο μαθητής

- ⇒ Να επιλέγει το είδος του πίνακα.
- ⇒ Να ορίζει τους πίνακες σε ένα πρόγραμμα.
- ⇒ Να εισάγει, να επεξεργάζεται και να τυπώνει τα στοιχεία ενός πίνακα.
- ⇒ Να αποφασίζει αν είναι απαραίτητη η χρήση πίνακα.
- ⇒ Να αναφέρει τις βασικές επεξεργασίες σε ένα πίνακα.
- ⇒ Να αναζητά και να ταξινομεί τα στοιχεία ενός πίνακα.



Προερωτήσεις

- ✓ Πώς μπορούν να αποθηκευτούν πολλά παρόμοια δεδομένα στον υπολογιστή, για παράδειγμα τα ονόματα όλων των μαθητών μίας τάξης;
- ✓ Για ποιο λόγο χρησιμοποιούνται στην άλγεβρα οι μεταβλητές με δείκτες;
- ✓ Σε ποιες περιπτώσεις χρειάζεται η ταξινόμηση κάποιων δεδομένων;
- ✓ Είναι εύκολη η αναζήτηση ενός συγκεκριμένου δεδομένου σε μη ταξινομημένα δεδομένα;

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 επαναεισαχθούν 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 λοι-

πρόν 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.



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 ιδίου 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.



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*.

Για παράδειγμα

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Θερμοκρασία [30]

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.

**Παράδειγμα 1**

Χρησιμοποιώντας μεταβλητές με δείκτες για το προηγούμενο παράδειγμα έχουμε το εξής πρόγραμμα

ΠΡΟΓΡΑΜΜΑ Θερμοκρασίες

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Θερμοκρασία[30], Μέση, Σύνολο

ΑΚΕΡΑΙΕΣ: i, Ημέρες

ΑΡΧΗ

Σύνολο <- 0

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 30

ΓΡΑΨΕ 'Δώσε τη θερμοκρασία'

ΔΙΑΒΑΣΕ Θερμοκρασία[i]

 Σύνολο <- Σύνολο+ Θερμοκρασία[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Μέση <- Σύνολο/30

Ημέρες <- 0

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 30

ΑΝ Θερμοκρασία[i] < Μέση **ΤΟΤΕ**

 Ημέρες <- Ημέρες+1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Μέση Θερμοκρασία:', Μέση

ΓΡΑΨΕ 'Ημέρες με μικρότερη θερμοκρασία', Ημέρες

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Με τη χρήση του πίνακα, όλα τα δεδομένα καταχωρούνται κάτω από το ίδιο όνομα μεταβλητής, στο παράδειγμα Θερμοκρασία. Η ανάγνωση όλων των δεδομένων απλοποιείται, αφού είναι μόνο μία εντολή, **ΔΙΑΒΑΣΕ** Θερμοκρασία[i], η οποία βρίσκεται μέσα σε ένα βρόχο και επαναλαμβάνε-



Η ανάγνωση, η επεξεργασία και η εκτύπωση των στοιχείων των πινάκων γίνεται πάντοτε από βρόχους, οι οποίοι επαναλαμβάνονται προκαθορισμένο αριθμό φορές, όσα είναι τα στοιχεία του πίνακα και υλοποιούνται καλύτερα στον προγραμματισμό με την εντολή επανάληψης ΠΑ.

ται όσες φορές απαιτείται όπως και ο υπολογισμός του αθροίσματος, $\text{Σύνολο} \leftarrow \text{Σύνολο} + \text{Θερμοκρασία}[i]$.

Οι τιμές μετά τον υπολογισμό της μέσης τιμής δεν χάνονται, αφού βρίσκονται στα στοιχεία του πίνακα και ξαναχρησιμοποιούνται για την εύρεση του αριθμού των ημερών, που η θερμοκρασία είναι μικρότερη από τη μέση. Ο υπολογισμός των ημερών είναι μόνο μία εντολή ΑΝ, η οποία βρίσκεται σε ένα βρόχο και επαναλαμβάνεται 30 φορές.

Παράδειγμα 2

Να γραφεί πρόγραμμα το οποίο να υπολογίζει τα βασικά στατιστικά μεγέθη, τη μέση τιμή, την τυπική απόκλιση και τη διάμεσο τιμή Ν ακεραίων αριθμών, όπου το Ν είναι από 2 μέχρι 100.

Τα δεδομένα εισάγονται από το πληκτρολόγιο και καταχωρούνται στον πίνακα Χ.

$$\text{Η μέση τιμή } \mu \text{ δίνεται από τον τύπο } \mu = \frac{\sum_{i=1}^N X_i}{N}$$

$$\text{Η τυπική απόκλιση } \sigma \text{ δίνεται από τον τύπο } \sigma^2 = \frac{\sum_{i=1}^N X_i^2}{N} - \mu^2$$

Για να βρεθεί η διάμεσος τιμή πρέπει υποχρεωτικά οι αριθμοί να ταξινομηθούν κατά αύξουσα σειρά. Τότε διάμεσος τιμή, είναι η τιμή για την οποία οι μισοί αριθμοί είναι μικρότεροι και οι άλλοι μισοί μεγαλύτεροι. Στην περίπτωση που το πλήθος των αριθμών είναι περιττό, τότε διάμεσος είναι ο μεσαίος, ενώ στην περίπτωση που είναι άρτιο, τότε διάμεσος είναι το ημίαθροισμα των δύο μεσαίων αριθμών.

Η ταξινόμηση των στοιχείων γίνεται με τη μέθοδο ταξινόμησης ευθείας ανταλλαγής, η οποία παρουσιάστηκε στο κεφάλαιο 3.

ΠΡΟΓΡΑΜΜΑ Στατιστική

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ : i, N, X[100], Διάμεσος, Άθροισμα, Άθροισμα_2, Βοηθητική

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΤ, Τυπ_Απόκλιση

ΑΡΧΗ

! Εισαγωγή δεδομένων

ΓΡΑΨΕ `Δώσε το πλήθος των αριθμών (μέγιστο 100)´

ΔΙΑΒΑΣΕ N

```

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N
    ΓΡΑΨΕ `Δώσε τον `i,`-το αριθμό`
    ΔΙΑΒΑΣΕ X[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
! Υπολογισμός αθροισμάτων
Αθροισμα <- 0
Αθροισμα_2 <- 0
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ N
    ΓΡΑΨΕ `Δώσε τον `i,`-το αριθμό`
    ΔΙΑΒΑΣΕ X[i]
    Αθροισμα <- Αθροισμα + X[i]
    Αθροισμα_2 <- Αθροισμα_2 + X[i]^2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

! Υπολογισμός μέσου όρου
ΜΤ <- Αθροισμα/N
!Υπολογισμός τυπικής απόκλισης
Τυπ_Απόκλιση <- T_P(Αθροισμα_2/N - ΜΤ^2)
!Ταξινόμηση των στοιχείων του πίνακα
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ N
    ΓΙΑ j ΑΠΟ N ΜΕΧΡΙ i ΜΕ ΒΗΜΑ -1
        ΑΝ X[j-1] > X[j] ΤΟΤΕ
            ! Αντιμετάθεση των στοιχείων j και j-1
            Βοηθητική <- X[j-1]
            X[j-1] <- X[j]
            X[j] <- Βοηθητική
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ !! j
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ !! i
!Υπολογισμός διαμέσου
ΑΝ N MOD 2 =0 ΤΟΤΕ
    Διάμεσος <- (X[N/2]+X[N/2+1])/2
ΑΛΛΙΩΣ
    Διάμεσος <- X[(N+1)/2]
ΤΕΛΟΣ_ΑΝ

! Εκτύπωση αποτελεσμάτων
ΓΡΑΨΕ `ΑΠΟΤΕΛΕΣΜΑΤΑ`
ΓΡΑΨΕ `=====`
ΓΡΑΨΕ `Πλήθος τιμών =`, N
ΓΡΑΨΕ `Μέση τιμή =`, ΜΤ
ΓΡΑΨΕ `Τυπική απόκλιση =`, Τυπ_Απόκλιση
ΓΡΑΨΕ `Διάμεσος =`, Διάμεσος
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Στατιστική

```



Στο πρόγραμμα αυτό δεν γίνεται έλεγχος για μηδενική τιμή του N

Προγραμματιστικό περιβάλλον Basic

```

` Στατιστική
DEFINT A-Z
DIM x(100)
CLS
sum = 0: sum2 = 0
` Εισαγωγή δεδομένων
INPUT "N=", n
FOR i = 1 TO n
    PRINT "Δώσε τον "; i; "-το αριθμό : ";
    INPUT "", x(i)
NEXT i
` Υπολογισμός αθροισμάτων
FOR i = 1 TO n
    sum = sum + x(i)
    sum2 = sum2 + x(i) ^ 2
NEXT i
mt! = sum / n      ` Μέση τιμή
ta! = SQR(sum2 / n - mt! ^ 2) `Τυπ.Απόκλ.
` Ταξινόμηση πίνακα
FOR i = 2 TO n
    FOR j = n TO i STEP -1
        IF x(j - 1) > x(j) THEN SWAP x(j - 1), x(j)
    NEXT j
NEXT i
` Υπολογισμός διαμέσου
IF n MOD 2 = 0 THEN
    median = (x(n / 2) + x(n / 2 + 1)) / 2
ELSE
    median = x((n + 1) / 2)
END IF
PRINT "      ΑΠΟΤΕΛΕΣΜΑΤΑ"
PRINT "=====
PRINT "Πλήθος τιμών      ="; n
PRINT "Μέση τιμή          ="; mt!
PRINT "Τυπική απόκλιση    ="; ta!
PRINT "Διάμεσος           ="; median
END

```


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 μέγεθος τους πρέπει να δηλώνεται στην αρχή του προγράμματος, ενώ παραμένει υποχρεωτικά σταθερό κατά την εκτέλεση του προγράμματος.

Η απόφαση για την χρήση ή όχι πίνακα για την 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 επίλυση του χρησιμοποιήθηκε ένας πί-

νακας 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



Ο πίνακας αυτός έχει τις θερμοκρασίες για 30 ημέρες αλλά για δέκα διαφορετικές πόλεις, δηλαδή υπάρχουν συνολικά 300 τιμές θερμοκρασίας. Για να καθοριστεί κάθε στοιχείο δεν αρκεί μόνο ένας δείκτης, αλλά απαιτούνται δύο δείκτες, ο ένας για την ημέρα και ο δεύτερος για την πόλη. Για παράδειγμα η πρώτη πόλη την τριακοστή ημέρα είχε θερμοκρασία 27.

Για την επεξεργασία των θερμοκρασιών μπορεί να χρησιμοποιηθεί ένας διδιάστατος πίνακας, στον οποίο ο πρώτος δείκτης δείχνει τη γραμμή (στο παράδειγμα την ημέρα) και ο δεύτερος τη στήλη (την πόλη). Το στοιχείο *Θερμοκρασία[30,1]* έχει την τιμή 27.

Στην γενική περίπτωση κάθε στοιχείο του πίνακα είναι το στοιχείο *Θερμοκρασία[i,j]* και αναφέρεται στη θερμοκρασία στην *i* γραμμή και την *j* στήλη, όπου το *i* παίρνει τιμή από 1 έως 30 και το *j* από 1 έως 10. Ο πίνακας *Θερμοκρασία* είναι ένας διδιάστατος πίνακας 30X10.

Παράδειγμα 3

Να γραφεί πρόγραμμα που να υπολογίζει τη μέση θερμοκρασία κάθε πόλης για τον προηγούμενο πίνακα θερμοκρασιών (δίδονται 30 θερμοκρασίες 10 πόλεων). Επίσης, για κάθε πόλη, να υπολογίζει πόσες ημέρες η θερμοκρασία ήταν κατώτερη από την αντίστοιχη μέση.

```

ΠΡΟΓΡΑΜΜΑ Θερμοκρασίες_2
ΜΕΤΑΒΛΗΤΕΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: Θερμοκρασία[30,10], Μέση[10]
    ΑΚΕΡΑΙΕΣ: i, j, Ημέρες, Σύνολο
ΑΡΧΗ
    ! Εισαγωγή δεδομένων
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 30
    ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
        ΓΡΑΨΕ `Δώσε τη θερμοκρασία', i, j
        ΔΙΑΒΑΣΕ Θερμοκρασία[i,j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
!Υπολογισμοί
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
    Σύνολο <- 0
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 30
        Σύνολο <- Σύνολο + Θερμοκρασία[i,j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    Μέση[j] <- Σύνολο/30
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 10
    Ημέρες <- 0
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 30
        ΑΝ Θερμοκρασία[i,j] < Μέση[j] ΤΟΤΕ
            Ημέρες <- Ημέρες+1
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ `Μέση Θερμοκρασία ', i, `Πόλης:', Μέση[j]
    ΓΡΑΨΕ `Ημέρες με μικρότερη θερμοκρασία', Ημέρες
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```



Η ανάγνωση, η επεξεργασία καθώς και η εκτύπωση των στοιχείων πολυδιάστατων πινάκων γίνεται πάντοτε από βρόχους, οι οποίοι υλοποιούνται στον προγραμματισμό με εμφωλευμένες εντολές επανάληψης ΠΑ.

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
  END

```

Εκτός από μονοδιάστατους και διδιάστατους 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 διδιάστατων.

Στο προηγούμενο παράδειγμα έστω ότι έχουμε θερμοκρασίες για κάθε μία 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]*.

		2001	ΠΟΛΗ				
		ΗΜΕΡΑ	1	2	...	10	
		1	25	21	...	32	
							31
		2000	ΠΟΛΗ				
		ΗΜΕΡΑ	1	2	...	10	...
		1	25	21	...	32	30
		1999	ΠΟΛΗ				31
		ΗΜΕΡΑ	1	2	...	10	...
		1	25	21	...	32	30
		2	26	22	...	31	
		
		30	27	23	...	30	

Παράδειγμα 4.

Να γραφεί πρόγραμμα το οποίο:

- Να διαβάζει τα ονόματα δέκα κινηματογράφων και τις αντίστοιχες εισπράξεις τους για κάθε ημέρα μίας εβδομάδας.
- Να υπολογίζει και να εκτυπώνει το άθροισμα των εισπράξεων κάθε κινηματογράφου, καθώς και τον κινηματογράφο με τη μέγιστη συνολική είσπραξη.
- Να υπολογίζει και να εκτυπώνει το άθροισμα των εισπράξεων κάθε ημέρας, καθώς και την ημέρα με τη μέγιστη συνολική είσπραξη.

Για την επίλυση του προβλήματος πρέπει να χρησιμοποιηθούν δύο πίνακες. Ο πρώτος θα περιέχει μόνο τα ονόματα των κινηματογράφων, θα είναι δηλαδή ένας μονοδιάστατος πίνακας χαρακτήρων με δέκα γραμμές. Ο δεύτερος θα περιέχει τις εισπράξεις, θα είναι ένας πίνακας δισδιάστατος ακεραίων αριθμών με δέκα γραμμές, μία για κάθε κινηματογράφο και επτά στήλες, μία για κάθε ημέρα.

Το πρόγραμμα ουσιαστικά αποτελείται από τρία τμήματα:

1. Την ανάγνωση των δεδομένων και την καταχώρηση του στους αντίστοιχους πίνακες, *Ονόματα* και *Εισπράξεις*.
2. Τον υπολογισμό του συνόλου των εισπράξεων ανά κινηματογράφο και την εύρεση της μέγιστης συνολικής εισπράξης, δηλαδή του αθροίσματος των γραμμών του πίνακα.
3. Τον υπολογισμό του συνόλου των εισπράξεων ανά ημέρα και την εύρεση της μέγιστης συνολικής εισπράξης, δηλαδή του αθροίσματος των στηλών του πίνακα.

Προσέξτε τις διαφορές του δεύτερου και του τρίτου τμήματος. Και οι δύο χρησιμοποιούν δύο εμφωλευμένους βρόχους, ένα για τις γραμμές και ένα για τις στήλες αλλά σε διαφορετική σειρά.

ΠΡΟΓΡΑΜΜΑ Κινηματογράφοι

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Εισπράξεις[10,7], i,j, Αθροισμα, Μέγιστο, Θέση

ΧΑΡΑΚΤΗΡΕΣ: Ονόματα[10]

ΑΡΧΗ

! Εισαγωγή δεδομένων και εκχώρηση τους σε δύο πίνακες

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

ΓΡΑΨΕ 'Δώσε το όνομα του ',i,' κινηματογράφου'

ΔΙΑΒΑΣΕ Ονόματα[i]

ΓΙΑ j **ΑΠΟ** 1 **ΜΕΧΡΙ** 7

ΓΡΑΨΕ 'Δώσε την ',j,'-η εισπραξη'

ΔΙΑΒΑΣΕ Εισπράξεις[i,j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

!Αθροισμα των στοιχείων του πίνακα Εισπράξεις ανά γραμμή

!και υπολογισμός του μέγιστου αθροίσματος

Μέγιστο <- 0

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

Αθροισμα <- 0

ΓΙΑ j **ΑΠΟ** 1 **ΜΕΧΡΙ** 7

Αθροισμα <- Αθροισμα+Εισπράξεις[i,j]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Συνολ. εισπραξη ',Ονόματα[i], '= ', Αθροισμα

ΑΝ Μέγιστο < Αθροισμα **ΤΟΤΕ**

Μέγιστο <- Αθροισμα

Θέση <- i

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

ΓΡΑΨΕ `Μέγ.συνολ.είσοπραξη`,Μέγιστο,`στον`,Ονόματα[Θέση]
!Αθροισμα των στοιχείων του πίνακα Εισπράξεις ανά στήλη
!και υπολογισμός του μέγιστου αθροίσματος
Μέγιστο <- 0
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ 7
    Αθροισμα <- 0
    ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 10
        Αθροισμα <- Αθροισμα + Εισπράξεις[i,j]
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ `Συνολ. είσοπραξη`,j,`-ης ημέρας =`, Αθροισμα
    ΑΝ Μέγιστο < Αθροισμα ΤΟΤΕ
        Μέγιστο <- Αθροισμα
        Θέση <- j
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ `Μέγ.συν.είσοπραξη`,Μέγιστο,`την`,Θέση,`-η ημέρα`
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

Προγραμματιστικό περιβάλλον Pascal

```

PROGRAM cinemas;
VAR
    i,j,k,max,sum:INTEGER;
    tickets:ARRAY[1..10,1..7] OF INTEGER;
    name:ARRAY [1..10] OF STRING;

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 είσοπραξη : `);
                    readln(tickets[i,j]);
                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;
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;
writeln (`Μέγιστο `, max, ` την `, k);
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 αναπτύχθη-

καν 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 αλγόριθμου 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 λιγότερη 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.



Ανακεφαλαίωση

Στο κεφάλαιο αυτό παρουσιάστηκαν οι πίνακες και πώς η ΓΛΩΣΣΑ χειρίζεται τους πίνακες. Ο πίνακας είναι μία ομάδα μεταβλητών ίδιου τύπου που αναφέρονται με ένα κοινό όνομα και αποθηκεύονται σε διαδοχικές θέσεις στη μνήμη. Για την πρόσβαση σε ένα ατομικό στοιχείο του πίνακα πρέπει να γραφεί το όνομα του πίνακα ακολουθούμενο από ένα δείκτη (μεταβλητή ή σταθερά). Οι πίνακες μπορεί να είναι μονοδιάστατοι, διδιάστατοι ή γενικότερα πολυδιάστατοι. Ο αριθμός των δεικτών καθορίζει τη διάσταση του πίνακα. Η επεξεργασία πινάκων γίνεται συνήθως με τη χρήση εντολών ΓΙΑ.

Οι τυπικές επεξεργασίες που γίνονται σε πίνακες περιλαμβάνουν την αναζήτηση, την ταξινόμηση και τη συγχώνευση πινάκων. Για αυτές τις επεξεργασίες έχουν αναπτυχθεί αρκετοί αλγόριθμοι και η μελέτη τους αποτελεί έναν από τους σημαντικούς τομείς της αλγοριθμικής.



Λέξεις κλειδιά

Μεταβλητή με δείκτη, Πίνακας, Στοιχείο πίνακα, Αναζήτηση, Ταξινόμηση, Συγχώνευση



Ερωτήσεις - Θέματα για συζήτηση

1. Τι ονομάζεται πίνακας;
2. Για ποιο λόγο χρησιμοποιούνται οι πίνακες;
3. Τι μπορεί να είναι οι δείκτες των πινάκων;
4. Ποια η διαφορά του πίνακα και του στοιχείου ενός πίνακα;
5. Πού ορίζεται η διάσταση του πίνακα;
6. Τι είδους δομή είναι ο πίνακας;
7. Ποιοι πίνακες ονομάζονται μονοδιάστατοι;
8. Δώσε ένα παράδειγμα ενός τρισδιάστατου πίνακα.

9. Πού αποθηκεύονται τα στοιχεία ενός πίνακα;
10. Ποια τα μειονεκτήματα της χρήσης πινάκων;
11. Ποιες οι τυπικές επεξεργασίες πίνακα;
12. Ποιοι είναι οι πιο διαδεδομένοι αλγόριθμοι αναζήτησης; Ποιες οι διαφορές τους;

Βιβλιογραφία

1. Κ. Αντωνικόπουλος, *Turbo Pascal 6.0 Θεωρία και Πράξη*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1993.
2. *QuickBASIC: Η εργαλειοθήκη του προγραμματιστή*, Κλειδάριθμος, Αθήνα, 1991.
3. Χρ. Κοίλιας, *Η QuickBasic και οι εφαρμογές της*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1992.
4. R. Shackelford, *Introduction to Computing and Algorithms*, Addison-Wesley, USA, 1998.
5. S. Leestma-L.Nyhoff, *Turbo Pascal, Programming and Solving*, McMillan, New York, 1990.
6. N. Wirth, *Systematic Programming: An introduction*, Prentice Hall, 1973.

Διευθύνσεις Διαδικτύου

⇒ <http://www.swcp.com/~dodrill/>

Περιέχει πληροφορίες αλλά και πολλές εκπαιδευτικές ασκήσεις για διάφορες γλώσσες προγραμματισμού.

⇒ <http://www.progsource.com>

Γενικές πληροφορίες, χρήσιμα προγράμματα, χρήσιμα βοηθητικά προγράμματα καθώς και αναφορές σε άλλες διευθύνσεις για διάφορες γλώσσες προγραμματισμού: Pascal, Delphi, C/C++, Java, Perl, Visual Basic.

⇒ www.cit.ac.nz/smac/pascal/default.htm

Πλήρης οδηγός της γλώσσας Pascal με πολλά εκπαιδευτικά παραδείγματα. Υπάρχει σε διάφορες γλώσσες όπως Αγγλικά, Γαλλικά και Γερμανικά.



⇒ <http://www.cs.vu.nl/~jprins/tp.html>

Πολλά παραδείγματα, βιβλία, εκπαιδευτικές εφαρμογές, και απαντήσεις σε ερωτήματα που δημιουργούνται συχνά σε Turbo Pascal.

⇒ <http://qbasic.com/>

Περιέχει εκπαιδευτικό οδηγό, κώδικα πολλών ασκήσεων και γενικές πληροφορίες για την Qbasic.

⇒ www.basicguru.com

Διεύθυνση που αναφέρεται αποκλειστικά στην Basic. Περιέχει πολλά έτοιμα παραδείγματα, πληροφορίες για εκδόσεις της γλώσσας, μεταφραστές για διάφορα λειτουργικά συστήματα.

Επίσης στο διαδίκτυο παρουσιάζουν ενδιαφέρον οι ακόλουθες ομάδες νέων (Usenet):

[comp.lang.pascal](#)

[comp.lang.pascal.misc](#)

Σχετικές με τη γλώσσα Pascal

[alt.lang.basic](#)

[comp.lang.basic.misc](#)

Σχετικές με τη γλώσσα Basic