Πίνακες

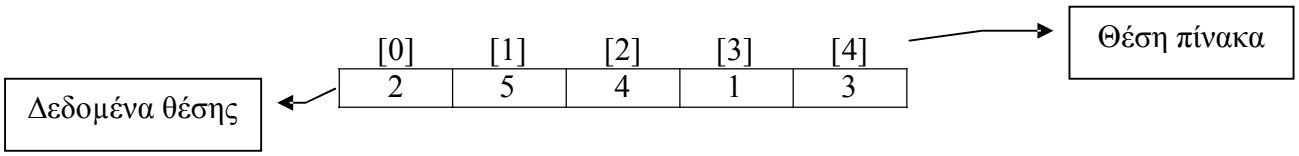
Εισαγωγή

Όπως ήδη έχει αναφερθεί, για τη διαχείριση δεδομένων, χρησιμοποιούνται μεταβλητές. Επειδή όμως συχνά καλούμαστε να λύσουμε προβλήματα με δεδομένα που ομαδοποιούνται, είναι πολύ πιο εύκολο και αποτελεσματικό να χρησιμοποιούνται οι πίνακες η οποίοι είναι μια μορφή δομών δεδομένων. Για παράδειγμα, θα ήταν πολύ πιο εύκολο ένας προγραμματιστής / καθηγητής να δημιουργήσει ένα πίνακα 30 θέσεων για να καταχωρεί τους βαθμούς των φοιτητών του παρά να δημιουργήσει 30 ξεχωριστές μεταβλητές.

Μπορείτε να φανταστείτε ένα πίνακα να αποτελείτε από διαδοχικές θέσεις κελιών, όπου αποθηκεύονται δεδομένα του ίδιου όμως τύπου, για παράδειγμα ακέραιοι αριθμοί όπως φαίνεται πιο κάτω:

2	5	4	1	3

Κάθε θέση του πίνακα αριθμείται έτσι ώστε να μπορούμε να διαχειριζόμαστε τα περιεχόμενα της. Η αρίθμηση αυτή ξεκινά από το μηδέν (0) όπως φαίνεται πιο κάτω για ένα πίνακα πέντε θέσεων:



Ορισμός Πινάκων

Οι πίνακες στη c ορίζονται στο χώρο που ορίζουμε και τις απλές μεταβλητές. Ο τρόπος που ορίζουμε ένα πίνακα στη γλώσσα c είναι ο εξής:

```
data_type  name_of_table[size_of_table];
```

όπου **data_type**: τύπος δεδομένων όπως int, double, char κλπ.
name_of_table: ένα οποιοδήποτε όνομα όπως ισχύει με τις μεταβλητές.
size_of_table: ένας οποιοδήποτε **ακέραιος θετικός** αριθμός.

Πιο κάτω φαίνεται ένα παράδειγμα ορισμού ενός πίνακα με το όνομα `myArray`:

```
int myArray[10];
```

Ο πίνακας `myArray` είναι μεγέθους 10 θέσεων και είναι τύπου ακεραίων αριθμών, μπορεί δηλαδή να φυλάει 10 ακέραιους αριθμούς ανά πάσα στιγμή.

Όπως τις μεταβλητές, έτσι και τους πίνακες μπορούμε να τους αρχικοποιούμε με τιμές κατά τον ορισμό τους. Συνεπώς, αν θέλουμε να αρχικοποιήσουμε τον πιο πάνω πίνακα με τις τιμές 1,2,3,.....10 το επιτυγχάνουμε με τον εξής τρόπο:

```
int myArray[] = {1,2,3,4,5,6,7,8,9,10};
```

Προσέξτε ότι το μέγεθος του πίνακα δε χρειάζεται να δηλωθεί καθώς αυτό μπορεί να υπολογιστεί από το πλήθος των στοιχείων που αρχικοποιούμε.

Επεξεργασία Πινάκων

Ο τρόπος που επεξεργαζόμαστε τους πίνακες είναι παρόμοιος με αυτός των μεταβλητών. Η διαφορά είναι ότι στους πίνακες πρέπει να καθορίζουμε κάθε φορά και τη θέση του πίνακα στην οποία αναφερόμαστε. Η πρόσβαση σε μια θέση του πίνακα πραγματοποιείται γράφοντας το όνομα του πίνακα και τη θέση. Στο προηγούμενο πίνακα, αν θέλουμε να **αναφερθούμε** στα δεδομένα στη θέση 6 τότε γράφουμε:

```
myArray[6]
```

Με αυτό το τρόπο, το `myArray[6]` αναφέρεται στα δεδομένα που υπάρχουν στη θέση 6 του πίνακα (αρχίζοντας από το 0). Αν ισχύει η αρχικοποίηση πιο πάνω, η τιμή αυτή είναι το 7. Αν τώρα επιθυμούμε να φυλάξουμε στη θέση 6 του πίνακα τη τιμή 145 τότε γράφουμε:

```
myArray[6] = 145;
```

Η τιμή 7 που είχε προηγούμενος ο πίνακας θα χαθεί και ο αριθμός 145 θα φυλαχτεί στη θέση 6 του πίνακα.

Παράδειγμα 1ο

Δημιουργείστε ένα πρόγραμμα που να ορίζει ένα πίνακα 7 θέσεων και αρχικοποιείται με τις τιμές 4,2,6,7,7,6,4. Ακολούθως οι τιμές αυτές να τυπώνονται στην οθόνη, η μια κάτω από την άλλη με τη χρήση επανάληψης.

Λύση

```
#include <stdio.h>



void main(void)
{

     int myTable[] = {4,2,6,7,7,6,4};
     int i;


     for (i=0; i<7; i++)
     {
          printf("\n%d", myTable[i]);
     }

}
```

Παράδειγμα 2^{ο}

Δημιουργείστε ένα πρόγραμμα που να ορίζει ένα πίνακα 3 θέσεων. Στη συνέχεια ο χρήστης θα καταχωρεί στις 3 αυτές θέσεις τρεις χαρακτήρες δικής του προτίμησης.

Λύση

```
#include <stdio.h>



void main(void)
{

     char myTable[3];
     int i;


     for (i=0; i<3; i++)
     {
          printf("Please enter a character:");
```

```
            scanf("%c", &myTable[i]);

            fflush(stdin);
      }


      for (i=0; i<3; i++)
      {
            printf("%c", myTable[i]);

      }

}
```

Σημείωση: Η εντολή `fflush(stdin);` διορθώνει τυχόν προβλήματα που προκύπτουν από την επεξεργασία χαρακτήρων. Μπορείτε να την αγνοήσετε.