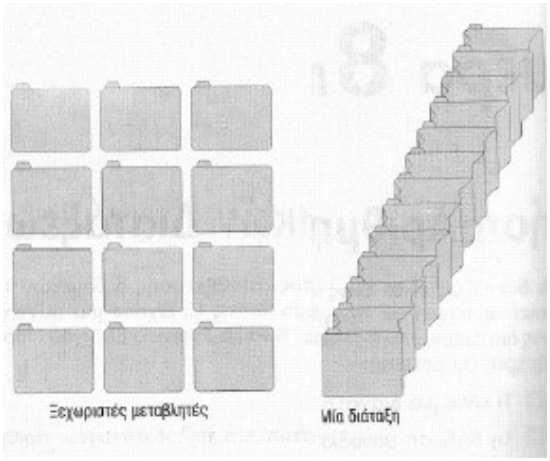
ΠΙΝΑΚΕΣ

1. ΟΙ ΠΙΝΑΚΕΣ ΣΤΗ 'C'

Οι πίνακες στη 'C' είναι μια λίστα μεταβλητών. Οι περισσότερες γλώσσες προγραμματισμού χρησιμοποιούν τέτοιες λίστες. Έστω ότι θέλετε να τηρήσετε τις πωλήσεις 50 πωλητών. Μπορείτε να ορίσετε μέχρι 50 ονόματα μεταβλητών και να εκχωρήσετε μια διαφορετική εγγραφή πωλητή ανα μεταβλητή. Ωστόσο, μια καλή πρακτική θα ήταν να χρησιμοποιηθεί μια διάταξη 50 στοιχείων αποθηκεύοντας το σύνολο για κάθε πωλητή στο αντίστοιχο στοιχείο του πίνακα. Το ακόλουθο σχήμα δείχνει τη διαφορά ανάμεσα στη χρήση των ξεχωριστών μεταβλητών και μιας διάταξης.



Σχήμα 1: Η έννοια των πινάκων

1.1 Ορισμός πινάκων

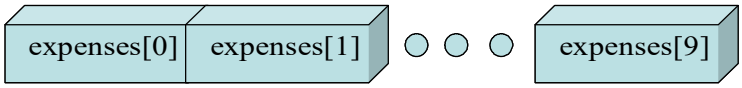
Μπορείτε να ορίσετε ένα πίνακα με οποιονδήποτε τύπο δεδομένων: ακέραιοι, πραγματικοί αριθμοί κινητής υποδιαστολής, χαρακτήρες κτλ. Όταν ορίζετε ένα πίνακα, ζητάτε από την γλώσσα 'C' να δεσμεύσει θέσεις μνήμης γι' αυτόν. Κάθε στοιχείο ενός πίνακα καταλαμβάνει το ίδιο ποσό μνήμης με μια άλλη μεταβλητή του ίδιου τύπου δεδομένων. Για παράδειγμα, κάθε στοιχείο σε έναν πίνακα χαρακτήρων καταλαμβάνει ένα byte. Το ίδιο ισχύει και για κάθε άλλο τύπο δεδομένων. Η 'C' αποθηκεύει όλα τα στοιχεία ενός πίνακα σε συνεχόμενες θέσεις στην μνήμη του υπολογιστή. Αυτό είναι σημαντικό κυρίως σε πιο προχωρημένα προγράμματα. Πάντα το πρώτο στοιχείο προηγείται του δευτέρου, το δεύτερο του τρίτου κτλ. Η μνήμη βεβαιώνει ότι δεν περιέχει κενά μεταξύ των στοιχείων ενός πίνακα. Αν μια τιμή κινητής υποδιαστολής καταλαμβάνει τέσσερα bytes, τότε το επόμενο στοιχείο ενός πίνακα κινητής υποδιαστολής βρίσκεται πάντα ακριβώς τέσσερα byres μετά το προηγούμενο του.

ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>

main ()
{
    float expenses [10];
}
```

Το παραπάνω πρόγραμμα δηλώνει έναν πίνακα τύπου float το οποίο αποτελείται από 10 στοιχεία. Καθένα από τα 10 είναι το ακριβές ισοδύναμο μιας απλής μεταβλητής float. Τα στοιχεία του πίνακα στην C αριθμούνται πάντοτε αρχίζοντας από το 0. Έτσι τα 10 στοιχεία του πίνακα αριθμούνται από το 0 μέχρι το 9 και αποθηκεύονται σε διαδοχικές θέσεις μνήμης όπως φαίνεται στο παρακάτω σχήμα.



Σχήμα 2: Διαδοχική αποθήκευση των στοιχείων ενός πίνακα

Ένα στοιχείο του πίνακα μπορεί να χρησιμοποιηθεί οπουδήποτε στο πρόγραμμα σας. Το κάθε στοιχείο του πίνακα προσπελαύνεται με τη χρήση του ονόματος της διάταξης, ακολουθούμενου από το δείκτη του στοιχείου. Όταν αναφέρεστε σε ένα στοιχείο του πίνακα, ο δείκτης του πίνακα μπορεί να είναι μια σταθερά ή μια ακέραια μεταβλητή.

1.2 Ονομασία και Δήλωση Πινάκων

Οι κανόνες για την εκχώρηση ονομάτων σε πίνακες είναι ίδιοι με τους κανόνες για τον ορισμό ονομάτων απλών μεταβλητών. Ένα όνομα πίνακα πρέπει να είναι μοναδικό. Δεν μπορεί να χρησιμοποιηθεί για άλλο πίνακα ή για οποιοδήποτε άλλο προσδιοριστή (π.χ. μεταβλητή, σταθερά κτλ). Όταν δηλώνετε ένα πίνακα, μπορείτε να καθορίσετε τον αριθμό των στοιχείων με μια σταθερά ή με την οδηγία '#define'.

ΠΑΡΑΔΕΙΓΜΑ

```
#define MAX_SIZE 10

main()
{
    int    expenses[MAX_SIZE]; /* Orismos Pinaka typoy int 10 stoixeiwn */
    long  rate[20];    /* Orismos Pinaka typoy long 20 stoixeiwn */
}
```

1.3 Μονοδιάστατος Πίνακας

Ένας μονοδιάστατος πίνακας έχει μόνο ένα δείκτη. Ένας δείκτης είναι ένας αριθμός σε αγκύλες που έπεται του ονόματος μιας διάταξης. Αυτός ο αριθμός μπορεί να προσδιορίζει τον αριθμό των ξεχωριστών στοιχείων σε μια διάταξη. Τα επιμέρους στοιχεία του πίνακα μπορούν να προσπελάσουν με τη χρήση του ονόματος του πίνακα, ακολουθούμενου από το δείκτη του στοιχείου, μέσα σε αγκύλες.

ΠΑΡΑΔΕΙΓΜΑ

```
#define MAX_SIZE 10

main ()
{
    int  expenses[MAX_SIZE]; /* Orismos Pinaka typoy int 10 stoixeiwn */
    int  i;
    expenses[0]= 4;        /* Ekxwrisi sto prwto stoixeio toy pinaka tin timi 4 */
    expenses[2]= 5*2-1;  /* Exxwrisi sto trito stoixeio toy pinaka */
}
```

Σε ένα πίνακα με n στοιχεία, η επιτρεπόμενη περιοχή δεικτών είναι από 0 μέχρι n-1. Εάν χρησιμοποιείτε την τιμή δείκτη μεγαλύτερη από n-1, μπορεί να έχετε σφάλματα στο πρόγραμμα. Ο μεταγλωττιστής (compiler) δεν αναγνωρίζει αν το πρόγραμμα χρησιμοποιεί ένα δείκτη εκτός των επιτρεπόμενων ορίων.

1.4 Αρχικοποίηση Πινάκων

Υπάρχουν δύο τρόποι αρχικοποίησης στοιχείων ενός πίνακα:

- Κατά τον ορισμό
- Μέσα στο πρόγραμμα

Χρησιμοποιώντας αγκύλες, μπορείτε να αρχικοποιήσετε οποιονδήποτε τύπο πίνακα. Για παράδειγμα για ένα πίνακα με τις ηλικίες πέντε παιδιών μπορούμε να γράψουμε το παρακάτω παράδειγμα

ΠΑΡΑΔΕΙΓΜΑ

```
main ()
{
    int  ages[5]={2,4,5,10,12};    /* Orismos kai Arxikopoisi Pinaka typoy */
                                   /* int 5 stoixeiwn */
}
```

Όταν αρχικοποιείτε ένα πίνακα οποιουδήποτε τύπου δεν χρειάζεται να ορίσετε ρητά το μέγεθος. Η 'C' γνωρίζει στην περίπτωση αυτή να δεσμεύει να δεσμεύει τις σχετικές θέσεις μνήμης.

ΠΑΡΑΔΕΙΓΜΑ

```
void main ()
{
    int  ages[]={2,4,5,10,12};    /* Orismos kai Arxikopoisi Pinaka typoy */
                                  /* int 5 stoixeiwn */
}
```

Στα περισσότερα προγράμματα, σπάνια γνωρίζετε τα περιεχόμενα ενός πίνακα όταν τους ορίζετε. Συνήθως του εκχωρείτε ως τιμές δεδομένα του χρήστη ή από αρχείο δίσκου. Ο βρόχος 'for' μπορεί να χρησιμοποιηθεί για την εκχώρηση τιμών σε πίνακες.

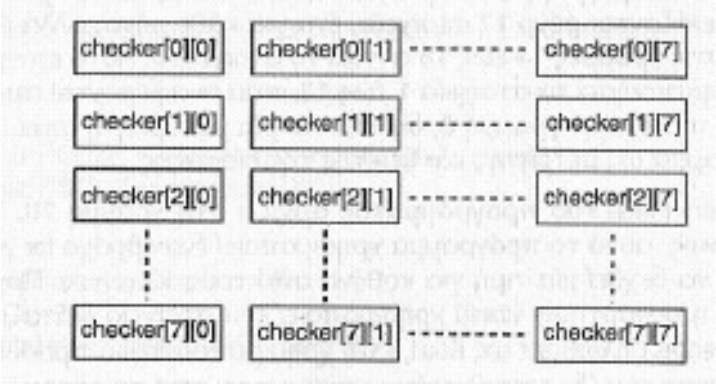
ΠΑΡΑΔΕΙΓΜΑ

```
#define ARRAY_SIZE 10
void main ()
{
    int  ages[ARRAY_SIZE]; /* Orismos Pinaka typoy int 5 stoixeiwn */
    int  index;

    for (index=0; index<ARRAY_SIZE; index++)
        ages[index]=(index+2);
}
```

1.5 Πολυδιάστατοι Πίνακες

Οι πολυδιάστατοι πίνακες είναι πίνακες με περισσότερους από έναν δείκτες. Ένας μονοδιάστατος πίνακας είναι μια λίστα τιμών. Ένας πολυδιάστατος πίνακας προσομοιώνει έναν πίνακα τιμών ή πολλούς πίνακες τιμών. Για παράδειγμα μπορείτε να γράψετε ένα πρόγραμμα που να παίζει ντάμα. Η ντάμα περιέχει 64 τετράγωνα τακτοποιημένα σε οκτώ γραμμές και οκτώ στήλες. Το πρόγραμμα αυτό θα μπορούσε να αναπαριστά την επιφάνεια της ντάμας ως μια δισδιάστατη διάταξη όπως φαίνεται στο παρακάτω σχήμα.



Σχήμα 3: Διαδοχική αποθήκευση των στοιχείων ενός πίνακα

Για να δεσμεύσετε πολυδιάστατους πίνακες, πρέπει να πληροφορήσετε την C ότι ο πίνακας έχει περισσότερες από μια διαστάσεις. Έτσι, στις παρενθέσεις γράφετε περισσότερες από μια διαστάσεις. Έτσι στις παρενθέσεις γράφετε περισσότερους από ένα δείκτες, έναν για κάθε διάσταση. Η 'C' είναι ελαστική σχετικά με τον ορισμό ενός πολυδιάστατου πίνακα κατά την αρχικοποίηση του ή κατά τον ορισμό του. Όπως με τους μονοδιάστατους πίνακες, αρχικοποιείτε τους πολυδιάστατους πίνακες με αγκύλες που ορίζουν διαστάσεις

ΠΑΡΑΔΕΙΓΜΑ

```
main()
{
    float    utilities[12][4];  /* Desmyesi 48 stoixeiwn typoy float */
    int      test[2][4] = {{14,30,12,81},
                           {15,27,40,9}};
}
```

1.6 Η συνάρτηση sizeof

Λόγω του τρόπου λειτουργίας της μνήμης, δεν θα πρέπει να επιχειρήσετε να δημιουργήσετε περισσότερα από 64 KB μεταβλητών δεδομένων. Γενικά 64 KB είναι αρκετός χώρος για δεδομένα προγραμμάτων. Το μέγεθος μιας διάταξης σε bytes εξαρτάται από τον αριθμό των στοιχείων που έχει, καθώς και από το μέγεθος κάθε στοιχείου. Για να υπολογίσετε τον απαιτούμενο αποθηκευτικό χώρο για μία διάταξη, πρέπει να πολλαπλασιάσετε τον αριθμό των στοιχείων στη διάταξη επί το μέγεθος του στοιχείου. Ένας εναλλακτικός τρόπος υπολογισμού είναι η χρήση της συνάρτησης 'sizeof'.Η συνάρτηση sizeof επιστρέφει τον αριθμό των bytes που δεσμεύει το όρισμα της. Αν ζητήσετε το μέγεθος ενός πίνακα, η sizeof() επιστρέφει τον αριθμό των bytes που δεσμεύονται για ολόκληρο τον πίνακα.

ΠΑΡΑΔΕΙΓΜΑ

```
#include <stdio.h>

void main ()
{
    int  scores[100]; /* Orismos Pinaka typoy int 100 stoixeiwn */
    int  n;

    n=sizeof(scores);
    printf("n equals to %d\n",n);
}
```

Στην οθόνη του υπολογιστή εμφανίζεται το παρακάτω αποτέλεσμα n equals to 400 bytes (σε πολύ παλιούς υπολογιστές ή και σε κάποιους καινούργιους ή μελλοντικούς αυτή η τιμή μπορεί να είναι διαφορετική).