

Παραδείγματα με μονοδιάστατους πίνακες

Τι θα τυπώσει?

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    //Αρχικές τιμές στο πίνακα
    int i, A[] = { 5,10,15,20,25 };
    // Εμφάνιση του πίνακα στη οθόνη
    for (i=0;i<sizeof(A)/sizeof(int);i++)
        printf("%d\n",A[i]);

    system("pause");
}
```

Επεξήγηση της εντολής for στο πρόγραμμα.

Σε μια for για την επεξεργασία των στοιχείων ενός πίνακα, συνήθως ο αριθμός των επαναλήψεων προσδιορίζεται από τον αριθμό των στοιχείων του πίνακα, δηλαδή κάποιο ακέραιο N και η εντολή σχηματίζεται ως εξής:

for (i=0;i<N;i++)

Στο παράδειγμα μας χρησιμοποιούμε τον τελεστή `sizeof()` για να βρούμε το μέγεθος του πίνακα σε byte [`sizeof(A)`] που το διαιρούμε με το μέγεθος του τύπου του πίνακα [`sizeof(int)`] και με αυτό τον τρόπο βρίσκουμε τον αριθμό των στοιχείων του πίνακα $N = \text{sizeof}(A) / \text{sizeof}(\text{int})$. Συνεπώς η for γίνεται τώρα:

for (i=0;i<sizeof(A)/sizeof(int);i++)

Χρήσιμο όταν δεν γνωρίζουμε τον αριθμό των στοιχείων ενός πίνακα.

Αποδίδοντας τις τιμές 1,2,3,..N στα στοιχεία ενός πίνακα

```
#include <stdio.h>
#include <stdlib.h>
#define N 20
main()
{
    int arr[N];
    int i = 0;

    for(i=0;i<sizeof(arr)/sizeof(int);i++) { // διότι N=sizeof(arr)/sizeof(int)
        arr[i] = i; // δίνει τις τιμές 1,2,3, κλπ.
    }

    for(i=0;i<N;i++) // Εμφάνιση του πίνακα στη οθόνη
        printf("%d\n", arr[i]);

    system("pause");
}
```

Τι θα τυπώσει?

```
#include <stdio.h>
#include <stdlib.h>
#define N 10
main()
{
    int i, arr[N]={0};

    for(i=0;i<N;i++) // Δημιουργία του πίνακα...
        arr[i++] = 99; // τι θα γίνει αν βάλουμε arr[i++] = 99;

    for(i=0;i<N;i++)
        printf("%d\n", arr[i]);

    system("pause");
}
```

Εύρεση μήκους γραμμής που εισάγεται από το πληκτρολόγιο (με <enter> σταματά):

```
#include <stdio.h>
#include <stdlib.h>
#define N 5
main() {
    int n, c;
    char line[100];
    n = 0;
    while( (c=getchar( )) != '\n' ) {
        if( n < 100 ) //έλεγχος για ν' αποθηκευτούν μέχρι 100 χαρακτήρες
            line[n] = c; // βάζω τον χαρακτήρα που διάβασα σε πίνακα
            n++; //για να μη ξεπεράσω το όριο των 100 χαρακτήρων
    }
    printf("length = %d\n", n);
    system("pause");
}
```

Επεξήγηση της εντολής:

while((c=getchar()) != '\n')

- διαβάζω χαρακτήρα [getchar()],
- τον αποθηκεύω στην μεταβλητή c [c=getchar()] και
- ταυτόχρονα ελέγχω αν η τιμή της c είναι διαφορετική από τον χαρακτήρα νέας γραμμής [!= '\n'].

Εισάγω του 20 πρώτους αριθμούς της ακολουθίας Fibonacci σε πίνακα

```
#include <stdio.h>
#include <stdlib.h>
main( ) {
    int i, fibonacci[ 20 ];

    fibonacci[ 0 ] = 0;
    fibonacci[ 1 ] = 1;

    for( i = 2; i < 20; i++ )
        fibonacci[ i ] = fibonacci[ i - 2 ] + fibonacci[ i - 1 ];
    for( i = 0; i < 20; i++ )
        printf( "Fibonacci[ %d ] = %f\n", i, fibonacci[ i ] );
}
```

Εξ ορισμού, οι πρώτοι δύο αριθμοί Fibonacci είναι το 0 και το 1, και κάθε επόμενος αριθμός είναι το άθροισμα των δύο προηγούμενων:

$$F_0=0,$$

$$F_1=1,$$

$$F_n = F_{n-1} + F_{n-2} \text{ (αναδρομικός τύπος)}$$

// Το αποτέλεσμα πρέπει να είναι:

// 0 , 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765

Αντιστρέφοντας ένα πίνακα

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 10
main() {
    int i, tmp, pin[N];
    srand(time(NULL)); //αρχικοποίηση της γεννήτριας τυχαίων αριθμών
    for (i=0; i<N; i++)
        pin[i] = 1 + rand() % 100; // τυχαίες τιμές στο διάστημα 1-100
    printf("\nO pinakas: \n"); //Εμφάνιση αρχικού πίνακα
    for (i=0; i<N; i++)
        printf("%d\n", pin[i]);
    //Αντιστροφή του πίνακα
    for (i=0; i<N/2; i++) { // μέχρι τη μέση του πίνακα
        tmp=pin[i];
        pin[i]= pin[N-i-1];
        pin[N-i-1]=tmp;
    }
    printf("\nO antistrofos pinakas: \n"); //Εμφάνιση του νέου πίνακα
    for (i=0; i<N; i++)
        printf("%d\n", pin[i]);
}
```