

Είσοδος/έξοδος χαρακτήρων: `getchar()` , `putchar()`

- Η συνάρτηση `getchar()` παίρνει έναν χαρακτήρα από το πληκτρολόγιο και τον αποδίδει στο εκτελούμενο πρόγραμμα.
- Η συνάρτηση `putchar()` παίρνει έναν χαρακτήρα από το εκτελούμενο πρόγραμμα και τον εμφανίζει στην οθόνη.
- Παράδειγμα

```
#include <stdio.h>
int main()
{
    char ch;
    ch = getchar(); /* anti tou scanf("%c",&ch); */
    putchar(ch);    /* anti tou printf("%c",ch); */
    return 0;
}
```

```
g <enter>
g
```

- Η `getchar()` δεν έχει ορίσματα. Η τιμή που επιστρέφει είναι ο χαρακτήρας που παίρνει από το πληκτρολόγιο. Αυτόν τον χαρακτήρα στη συνέχεια αποδίδει στη μεταβλητή `ch`.
- Η `putchar()` έχει ένα όρισμα που είναι κάποιος χαρακτήρας ή ακολουθία διαφυγής ή κάποια μεταβλητή ή συνάρτηση της οποίας η τιμή είναι ένας μοναδικός χαρακτήρας.
- Π.χ. `putchar('A');` ; `putchar('\n');` ; `putchar(ch);` ; `putchar(getchar());` ;

Επαναλήψεις: οι τρεις βρόχοι της C

<u>ΒΡΟΧΟΣ</u>	<u>ΠΑΡΑΔΕΙΓΜΑΤΑ</u>
<code>while (συνθήκη)</code> <code>εντολή;</code>	<code>while (i++ < 20)</code> <code>α *= 2;</code>
<code>do</code> <code>εντολή;</code> <code>while (συνθήκη);</code>	<code>do</code> <code>printf("Hello!\n");</code> <code>while (i++ < 10);</code>
<code>for(αρχικοποίηση; συνθήκη; αναπροσαρμογή)</code> <code>εντολή;</code>	<code>for(n=0; n<10; n++)</code> <code>printf("%d %d\n",n,2*n-1);</code>

Το παρακάτω πρόγραμμα δέχεται χαρακτήρες από το πληκτρολόγιο και τους εμφανίζει στην οθόνη μέχρι να πληκτρολογήσουμε '*'.

```
#include <stdio.h>
#define STOP '*'
int main()
{
    char ch;

    ch = getchar();
    while (ch != STOP)
    {
        putchar(ch);
        ch = getchar();
    }
    return 0;
}
```

↔

```
while ((ch=getchar()) != STOP)
    putchar(ch);
```

Με τον παρακάτω βρόχο καθαρίζουμε το buffer εισόδου από υπολείμματα (π.χ. τον χαρακτήρα νέας γραμμής) πριν το διάβασμα νέων δεδομένων.

```
while( getchar() != '\n'); /* Empty input buffer */
```

Πρόγραμμα που μετράει πλήθος χαρακτήρων και λέξεων μέχρι να πληκτρολογήσουμε [enter].

```
#include <stdio.h>
#define YES 1
#define NO 0
int main()
{
    int ch, nw = 0; /* nw = word counter */
    long nc = 0L; /* nc = char counter */
    int word = NO; /* YES if ch is in a word */

    while ((ch = getchar()) != '\n')
    {
        nc++; /* count characters */

        if (ch != ' ' && ch != '\t' && word == NO)
        {
            word = YES;
            nw++; /* count words */
        }

        if ((ch == ' ' || ch == '\t') && word == YES)
            word = NO;
    }

    printf("characters = %ld, words = %d\n",nc,nw);
    return 0;
}
```

My name is Costas[enter] ---> characters = 17, words = 4

Μορφοποίηση δεδομένων κινητής υποδιαστολής

```
#include <stdio.h>
int main()
{
    float a,b,c,d;

    a = 21.234;  b = 5467.1;
    c = .99;    d = 130.25;
    printf("Numbers:\n%f %f\n",a,b);
    printf("%f %f\n",c,d);
    return 0;
}
```

χωρίς μορφοποίηση εξόδου

```
Numbers:
21.233999 5467.100098
0.990000 130.250000
```

Μορφοποίηση δεκαδικού μέρους

```
#include <stdio.h>
int main()
{
    float a,b,c,d;

    a = 21.234;  b = 5467.1;
    c = .99;    d = 130.25;
    printf("Numbers:\n%.3f %.3f\n",a,b);
    printf("%.3f %.3f\n",c,d);
    return 0;
}
```

3 δεκαδικά ψηφία

```
Numbers:
21.234 5467.100
0.990 130.250
```

Λεπτομερής καθορισμός τρόπου εμφάνισης πραγματικών αριθμών

```
#include <stdio.h>
int main()
{
    float a,b,c,d;

    a = 21.234;  b = 5467.1;
    c = .99;    d = 130.25;
    printf("Numbers:\n%8.3f %8.3f\n", a,b);
    printf("%8.3f %8.3f\n", c,d);
    return 0;
}
```

```
Numbers:
 21.234 5467.100
  0.990 130.250
```

χώρος 8 θέσεων: μία για υποδιαστολή, τρεις για τα δεκαδικά ψηφία και 4 για ακέραιο μέρος

ΠΙΝΑΚΕΣ

- Ένας πίνακας είναι μια συλλογή δεδομένων που αποτελείται από στοιχεία του ίδιου τύπου (π.χ. πίνακας ακεραίων).
- Τα στοιχεία ενός πίνακα αποθηκεύονται σε συνεχόμενες θέσεις μνήμης
- Στη C ο δείκτης που προσδιορίζει τη θέση του πρώτου στοιχείου ενός πίνακα N στοιχείων είναι το 0 και του τελευταίου στοιχείου το N-1. Αντιθέτως, στην Pascal και στο MATLAB, οι δείκτες ξεκινούν από το 1.
- Παράδειγμα δήλωσης πίνακα 10 ακεραίων με όνομα **a** και πίνακα 5 πραγματικών αριθμών με όνομα **weight**:

```
int a[10];
float weight[5];
```

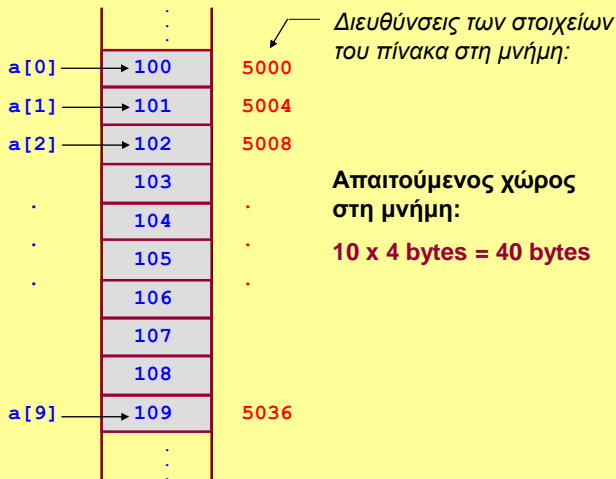
Τα στοιχεία του πίνακα a είναι τα: **a[0], ..., a[9]**.

Παράδειγμα άμεσης προσπέλασης στοιχείων: **weight[2] = 31.25;**

ΠΙΝΑΚΕΣ

- Η δήλωση ενός πίνακα δεσμεύει συνεχόμενες θέσεις στη μνήμη.
- Παράδειγμα:

```
int a[10];
```



ΠΙΝΑΚΕΣ

- Με τους πίνακες μπορούμε να ομαδοποιούμε πολλές μεταβλητές με το ίδιο όνομα. Π.χ. Έστω ότι θέλουμε να διαβάσουμε 100 πραγματικούς αριθμούς σε ένα πρόγραμμα, τότε αντί να δηλώσουμε 100 μεταβλητές (`float ar1, ar2, ar3, ..., ar100;`) και να επαναλαμβάνουμε την είσοδο/έξοδο και επεξεργασία των αριθμών με πανομοιότυπο τρόπο, ορίζουμε έναν πίνακα τύπου `float` με τόσες θέσεις όσοι και οι αριθμοί: `float ar[100];` Τώρα, η είσοδος, η έξοδος και η επεξεργασία των αριθμών μπορεί να γίνει κομψά και αποτελεσματικά με χρήση των εντολών επανάληψης (βρόχων).
- Προσπέλαση στοιχείων πίνακα:

```
a[0] = 0; /* εκχώρηση τιμής στο 1ο στοιχείο του πίνακα a */
j = 5; a[j] = j; /* εκχώρηση στο 6ο στοιχείο του a */
i = 2; j = 1; a[i*3+j] = a[i] + a[j]; /* υπολογισμός δείκτη
πίνακα από αποτίμηση παράστασης */
```

Αρχικοποίηση πίνακα

```

/* Υπολογισμος plithous psifiwn pinaka apo metaglwttisth */
#include <stdio.h>
int main()
{
    int i;
    int days[]={31,28,31,30,31,30,31,31,30,31,30,31};

    for(i=0; i<12; i++)
        printf("Ο μηνas %d exei %d meres\n",i+1,days[i]);
    return 0;
}

```

```

Ο μηνas 1 exei 31 meres
Ο μηνas 2 exei 28 meres
Ο μηνas 3 exei 31 meres
...
Ο μηνas 12 exei 31 meres

```

Ταξινόμηση αριθμών

```

#include <stdio.h>
int main()
{
    int i,j;
    float m;
    float a[6] = {-1.5, 3.4, -1.6e2, 9., .2, 2.7e-1};

    for(i=0; i<5; i++)
        for(j=i+1; j<6; j++)    /* εμφωλευμένη for */
            if(a[i] > a[j])
            {
                m = a[i];
                a[i] = a[j];
                a[j] = m;
            }

    for(i=0; i<6; i++)
        printf("%7.2f ",a[i]);
    printf("\n");

    return 0;
}

```

```

-160.00   -1.50    0.20    0.27    3.40    9.00

```

Εισαγωγή δεδομένων σε πίνακες

```
#include <stdio.h>
#define TRUE 1
#define FALSE 0
int main()
{
    int i, pin1[10], pin2[10];
    int flag=TRUE;

    printf("EISAGWGH DEDOMENWN\n");
```

όχι καλός τρόπος

```
    i = 0;
    while (flag) /* while (flag == TRUE) */
    {
        printf("Input 2 integer values for ");
        printf("pin1[%d], pin2[%d]:\n",i,i);
        scanf("%d %d",&pin1[i],&pin2[i]);
        i++;
        if (i == 10)
            flag = FALSE;
    }
    ...
}
```

Εισαγωγή δεδομένων σε πίνακες

```
#include <stdio.h>
int main()
{
    int i, pin1[10], pin2[10];

    printf("EISAGWGH DEDOMENWN\n");
```

καλύτερος τρόπος

```
    i = 0;
    while (i < 10)
    {
        printf("Input 2 integer values for ");
        printf("pin1[%d], pin2[%d]:\n",i,i);
        scanf("%d %d",&pin1[i],&pin2[i]);
        i++;
    }
    ...
}
```

Εισαγωγή δεδομένων σε πίνακες

```
#include <stdio.h>
int main()
{
    int i, pin1[10], pin2[10];

    printf("EISAGWGH DEDOMENWN\n");

    i = 0;
    do
    {
        printf("Input 2 integer values for ");
        printf("pin1[%d], pin2[%d]:\n",i,i);
        scanf("%d %d",&pin1[i],&pin2[i]);
        i++;
    } while (i < 10);
    ...
}
```

Όχι καλός τρόπος. Ο βρόχος εκτελείται τουλάχιστον 1 φορά!

Εισαγωγή δεδομένων σε πίνακες

```
#include <stdio.h>
int main()
{
    int i, pin1[10], pin2[10];

    printf("EISAGWGH DEDOMENWN\n");

    for(i = 0; i < 10; i++)
    {
        printf("Input 2 integer values for ");
        printf("pin1[%d], pin2[%d]:\n",i,i);
        scanf("%d %d",&pin1[i],&pin2[i]);
    }
    ...
}
```

Ο καλύτερος τρόπος.