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

ΒΡΟΧΟΣ	ΠΑΡΑΔΕΙΓΜΑΤΑ
`while ( συνθήκη )` `  εντολή;`	`while ( i++ < 20 )` `  q *= 2;`

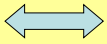
`do`
`  εντολή;`
`while ( συνθήκη );`
	`do`
	`  printf("Hello!\n");`
	`while ( i++ < 10 );`

`for(αρχικοποίηση; συνθήκη; αναπροσαρμογή)` `  εντολή;`	`for(n=0; n<10; n++)` `  printf("%d %d\n",n,2*n-1);`

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 = 0;     /* 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 = %d, 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

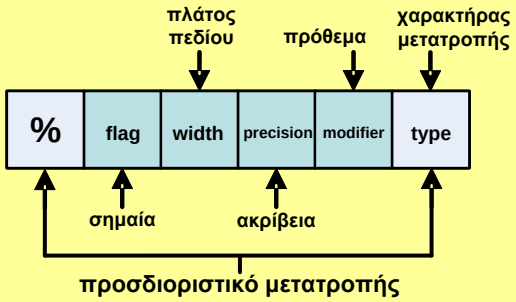
```
#include <stdio.h>
int main()
{
  float a,b,c,d;

  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

Μορφοποίηση δεδομένων: προαιρετικά πεδία προσδιοριστικών μετατροπής



Ακρίβεια (precision)

Για τους τύπους float/double καθορίζει τον αριθμό των δεκαδικών ψηφίων που θα εμφανισθούν με κατάλληλη στρογγυλοποίηση (default: 6)

Παραδείγματα:

```
printf("x = %f\n",x); /* εμφανίζει 6 δεκαδικά ψηφία */
printf("x = %.f\n",x); /* δεν εμφανίζει δεκαδικά ψηφία */
printf("x = %.3f\n",x); /* εμφανίζει 3 δεκαδικά ψηφία */
```

```
printf("x = %.*f\n",3,x); /* εμφανίζει 3 δεκαδικά ψηφία */

printf("How many decimal digits? ");
scanf("%d",&n);
printf("x = %.*f\n",n,x); /* εμφανίζει n δεκαδικά ψηφία */
```

Ακρίβεια (precision)

Για συμβολοσειρές μπορούμε να καθορίσουμε το πλήθος των χαρακτήρων που θα εμφανισθούν με την ίδια τεχνική. Ο αριθμός μετά την τελεία καθορίζει το πλήθος των χαρακτήρων προς εμφάνιση.

Παραδείγματα:

```
#define STR "This is a string"
printf("x = %s\n",STR); /* εμφανίζει όλους τους χαρακτήρες */
printf("x = %.7s\n",STR); /* εμφανίζει 7 χαρακτήρες */
x = This is
printf("How many characters to print? ");
scanf("%d",&n);
printf("x = %.*s\n",n,STR); /* εμφανίζει n χαρακτήρες*/
printf("x = %.s\n",STR); /* δεν εμφανίζει τίποτα */
```

Πλάτος πεδίου

Το πλάτος πεδίου καθορίζει, με έναν ακέραιο μετά το '%' και πριν την '.', το ελάχιστο πλήθος των χαρακτήρων που θα εμφανισθούν συμπεριλαμβανομένων και των ψηφίων ακριβείας ώστε να χωράει ο αριθμός.

Παραδείγματα:

```
int i = 100, j = -5;
float x = 1.15719, y = -145.25;
printf("|%d|,|%d|,|%f|,|%f|\n",i,j,x,y);
printf("|%4d|,|%4d|,|%8.2f|,|%8.2f|\n",i,j,x,y);
printf("|%2d|,|%1d|,|%5f|,|%5f|\n",i,j,x,y);
```

Έξοδος

```
|100|,|-5|,|1.157190|,|-145.250000|
| 100|,|  -5|,|    1.16|,| -145.25|
|100|,|-5|,|1.157190|,|-145.250000|
```

πλάτος πεδίου 8 χαρακτήρων

Πλάτος πεδίου

Στην περίπτωση που το πλάτος μπορεί να καθοριστεί αργότερα, π.χ. με χρήση της scanf, υπάρχει η δυνατότητα να χρησιμοποιηθεί το '*' αμέσως μετά το '%', και το πλάτος πεδίου να καθορισθεί στη λίστα ορισμάτων της printf.

Παραδείγματα:

```
int i = 100, d1, d2, a;   float x = 1.15719;
printf("Dose plath pediou kai akribeia: \n");
scanf("%d%d%d",&d1,&d2,&a); /* έστω: 4 8 2 <enter> */
/* printf("|%4d|,|%8.3|,|%8.2f|\n",i,x,x); */
printf("|%*d|,|%*.3f|,|%*.*f|\n",d1,i,d2,x,d2,a,x);
```

Έξοδος

```
| 100|,|   1.157|,|    1.16|
```

Πρόθεμα

Προσδιορίζει μέγεθος τύπου: h για short και l ή L για long. Το πρόθεμα τοποθετείται αμέσως πριν τον καθορισμό του τύπου.

Παράδειγμα:

```
#include <stdio.h>
int main(void)
{
  short i=10;  int j=32768;  long k=1000000;
  printf("i=%hd, j=%hd, j=%ld, k=%ld\n",i,j,j,k);
  return 0;
}
```

Ο 32768 είναι εκτός των ορίων του short int (-32768 … 32767)

Έξοδος

```
i=10, j=-32768, j=32768, k=1000000
```

Σημαίες

Οι σημαίες τροποποιούν τις εξ'ορισμού επιλογές μορφοποίησης και τοποθετούνται ακριβώς μετά το %. Παρακάτω αναφέρονται 4 από αυτές:

-	Στοίχιση στα αριστερά
0	Το πεδίο συμπληρώνεται με μηδενικά αντί για κενά
+	Όλοι οι αριθμοί εμφανίζονται με το πρόσημό τους
κενό	Οι θετικές τιμές ξεκινούν με κενό χαρακτήρα

Παραδείγματα:

```
int i = 100;   float x = 1.23;
printf("|%-10d|,|%+6.2f|,|%-+6.2f|,|% f|",i,x,x,x);
printf(",|%06.2f|\n",x);
```

Έξοδος

```
|100       |,| +1.23|,|+1.23 |,| 1.230000|,|001.23|
```