

## Форматиран вход-изход

Функциите за форматиран вход-изход `printf` и `scanf` са функции с променлив брой аргументи, които не са част от езика C. В него няма дефиниран вход-изход. Това са просто полезни функции от стандартната библиотека на езика. Поведението им е дефинирано в ANSI стандарта.

```
int printf (const char *format [, argument, ...]);
```

```
int scanf (const char *format [, address, ...]);
```

където:

**const char \*format** - задължителен форматиращ низ; той определя броя на аргументите на двете функции, чрез форматни спецификатори задавани със символа % общият вид на форматните спецификатори е

**% [флаг] [ширна на полето] [.точност] [модификатор] конвертиращ символ**

някои конвертиращи символи за извеждане и въвеждане:

<b>%i</b> или <b>%d</b>	<b>int</b>
<b>%f</b>	<b>float</b>
<b>%c</b>	<b>char</b>
<b>%s</b>	<b>Низ от символи</b>

някои флагове:

<b>-</b>	подравнява изхода в ляво (по подразбиране е в дясно)
<b>+</b>	числата със знак се извеждат задължително с + или -
<b>space</b>	положителните числа се извеждат със <b>space</b> , отрицателните – с -
<b>0</b>	при числови стойност се запълва с 0

някои модификатори – l за long и double, h за short

**argument** - списък от аргументи с които ще се извършва обмена.

**връщан резултат:**

**print** - броя на изведените символи или константата EOF(дефинирана в `stdio.h`), при неуспешно извеждане;

**scanf** - броя на правилно обработените полета и при прочитане на край на файл - EOF (под правилно обработени полета се разбира, прочитане, преобразуване в указания формат и присвояване на преобразуваната стойност на

адреса на програмния обект от списъка за аргументи). Ако не е преобразувано нито едно поле върнатият резултат е 0 (нула).

Примери:

<pre>#include&lt;stdio.h&gt; int main(){     int a,b,n;     float c,d; char z='a';     a = 15;     b = a / 2;     n=printf("%d %3d %03d\n",b,b,b);     printf("printed %d char\n",n);     n=printf("%+d %-3d\n",b,b);     printf("printed %d char\n",n);     c = 15.3;    d = c / 3;     n=printf("%3.2f\n",d);     printf("printed %d char\n",n);     n=printf("%c %3c %%\n",z,z);     printf("printed %d char\n",n);     return 0; }</pre>	<pre>7 7 007 printed 10 char +7 7 printed 7 char 5.10 printed 5 char a a % printed 8 char</pre>
--	---

<pre>#include&lt;stdio.h&gt; int main(){     int a,n; float f;    double d;     printf("input a,f and d:");     n=scanf("%d%f%lf",&amp;a,&amp;f,&amp;d);     printf("a=%d f=%5.3f d=%6.4lf n=%d\n",a,f,d,n);     return 0; }</pre>	<pre>input a,f and d:45 67.3 86.45 a=45 f=67.300 d=86.4500 n=3</pre>
--	--

<pre>#include&lt;stdio.h&gt; int main(){     char ch; int i,n;     for(i=0;i&lt;3;i++){         printf("c:");         n=scanf("%c",&amp;ch);         printf(" %c %d n=%d\n",ch,ch,n);     }     return 0; }</pre>	<pre>c:a a 97 n=1 c: 10 n=1 c:f f 102 n=1</pre>
---	---

Внимание: & при scanf - адреси!

при разлика в % и аргументите!

при scanf роля на празен интервал в форматната спецификация – игнорира всички празни полета (интервал, табулация, нов ред) във форматни спецификации, където това има значение (например %c)