

Kiírás általános formátuma:

`%[Zászlók][Szélesség][.Pontosság][Hossz]Típus`

A [] zárójelben lévő specifikátorok elhagyhatóak. De, ha mégis használjuk őket, akkor a megadott sorrendben kell használni. Tehát a szélességet nem használhatjuk előbb mint a zászlókat, és a hossz-t nem használhatjuk előbb mint a pontosságot...

Típus	Kimenet	Példa
c	Karakter	a
d vagy i	Előjeles egész	392
e	Tudományos kiírás (mantissza és exponens) az e karaktert használva.	3.9265e+2
E	Tudományos kiírás (mantissza és exponens) az E karaktert használva	3.9265E+2
f	Lebegőpontos szám	392.65
o	Előjeles 8-as számrendszerbeli szám	610
s	Karakterekből álló string	példa
u	Előjeltelen egész	7235
x	Előjeltelen 16-os számrendszerbeli szám	7fa
X	Előjeltelen 16-os számrendszerbeli szám nagybetűkkel	7FA
p	Pointer cím	B800:0000

Zászlók	Leírás
-	balra igazítás a megadott szélességben (elhagyva az alap eset a jobbra igazítás)
+	minden számnál kiírja az előjelet, még a pozitívnál is
(szóköz)	ha nem lesz előjel kiírva, akkor egy üres szóközt tesz az érték elé
#	o, x vagy X típusnál használva a 0-tól különböző számoknál 0, 0x, 0X-t ír az érték elé e, E vagy F típusnál használva a kimenetre mindenképpen ír tizedespontot is
0	Ha meg van adva szélesség, akkor szóközök helyett 0-val egészíti ki a szám elejét

Szélesség	Leírás
(szám)	Minimum hány karakter hosszan írja ki az értéket. Ha az érték rövidebb mint ez a szám, akkor az érték szóközökkel lesz kiegészítve kiírásakor
*	A szélesség nem a formátumban van megadva, hanem egy további egész szám paraméter-ként. Példa: <code>printf("%5d", 123)</code> ekvivalens azzal, hogy <code>printf("%*d", 5, 123)</code>

.Pontosság	Leírás
.szám	d, i, o, u, x, X típusnál a minimum kiírási szélességet határozza meg. Ha az érték ennél rövidebb, akkor az érték 0-al lesz kiegészítve. Ha hosszabb az érték akkor nem lesz levágva. 0 pontosság azt jelenti, hogy a 0 számnál semmi nem lesz kiírva e, E, F típusnál a tizedespont utáni minimum tizedes jegyek számát határozza meg s típusnál a maximum kiírható karakterek számát határozza meg. Ha a string ennél hosszabb, akkor a vége le lesz vágva
.*	A pontosság nem a formátumban van megadva, hanem egy további egész paraméterként

Hossz	Leírás
h	Az érték short int vagy unsigned short int-ként lesz kiírva. (i, d, o, u, x, X típusoknál használható)
l	Az érték long int vagy unsigned long int-ként lesz kiírva (i, d, o, u, x, X típusoknál használható)
L	Az érték long double-ként lesz kiírva (e, E, f típusoknál használható)

Beolvasás általános formátuma:

%[*][Szélesség][Módosítók]Típus

	Leírás
*	Egy opcionális érték, ami azt jelzi, hogy az érték be lesz olvasva, de nem lesz eltárolva a megadott paraméterben
Szélesség	Megadja, hogy maximum hány karaktert olvashat be az aktuális olvasási folyamatban
Módosítók	h : short int (d, i típusoknál), unsigned short int (o, u, x típusoknál) l : long int (d, i típusoknál), unsigned long int (o, u, x típusoknál), double (e, f típusoknál) L : long double (e, f típusoknál)
Típus	Egy karakter ami meghatározza, hogy milyen típusú értéket szeretnénk beolvasni

Típus	Kielégítő Bemenet	Paraméter
c	Egy darab karakter. (A következő karaktert olvassa be.) Ha a szélesség nem egy, akkor szélesség mennyiségű karaktert olvas be, és a paraméterként megkapott tömbben tárolja el őket. '\0' karaktert nem teszi automatikusan a végére.	char *
d	Egész szám. Opcionálisan + vagy - lehet az elején	int *
e, E, f	Lebegőpontos szám. Opcionálisan + vagy - lehet a szám elején vagy e és E esetén az e vagy E karakter után. Példa érvényes bemenetek: -732.12 1.02E+2 9.36E-2	float *
o	8-as számrendszerbeli egész szám	int *
s	Karakterekből álló string. Karaktert olvasd addig amíg egy WhiteSpace karaktert nem talál. WhiteSpace karakterek: szóköz, '\0', új sor karakter, tab	char *
u	Előjeltelen egész szám	unsigned int *
x, X	16-os számrendszerbeli szám	int *

Fordította: Fritsi Dániel
<http://www.stud.u-szeged.hu/Fritsi.Daniel/>

Eredeti Forrás: <http://www.cplusplus.com/reference/clibrary/cstdio/printf.html>
<http://www.cplusplus.com/reference/clibrary/cstdio/scanf.html>