

1. A C nyelv egyszerű adattípusai, printf(), scanf()

1.1 Az első C program beírása és futtatása

```
#include <stdio.h> /* A printf függvény használata miatt szükséges */

int main (int argc, char *argv[])
{
    printf("Kupacsapataink szereplése elkészítve !\n");
    return 0; /* Jelzi az operációs rendszernek, hogy a program */
             /* szabályosan fejeződött be.*/
}
```

A szövegben szereplő '\n' az új sor jele. Az <stdio.h> a szabványos I/O műveletekkel kapcsolatos definíciókat tartalmazza. A C utasításokat a ';' zárja !

1.2 Az alapvető egyszerű adattípusok felsorolása

Minden adatot definiálni kell, azok használata előtt. Az alapvető adattípusok a következők :

| <u>Típus</u> | <u>Értelmezés</u> | <u>Byte</u> | <u>Definíció</u> |
|--------------|---|-------------|------------------|
| karakter | Az érvényes karakterkészlet egy eleme. | 1 | char |
| integer | Bináris egész szám. | 2 | int |
| float | Egyszeres pontosságú lebegőpontos szám. | 4 | float |
| double | Kétszeres pontosságú lebegőpontos szám. | 8 | double |

A definíció a tárolási osztálynak, változó típusának majd - egy vagy több szóköz után - a változó nevének megadásából áll. Ezt egy ; (pontosvessző) karakter követi. A változó neve tartalmazhat betűket, számokat és az _ (aláhúzás) karaktert, de mindenképpen betűvel vagy aláhúzással kell kezdődnie. A kis és nagybetűk között a fordító különbséget tesz ! A változó nevét célszerű "beszédesre" választani, azaz jó ha elárulja azt, hogy mire akarjuk használni ! A C nyelvben azonban nemcsak a felsorolt alapvető adattípusok használhatók, ugyanis létezik néhány minősítő szimbólum, melyek egész jellegű mennyiségek esetén használhatók, ill.egy esetben a double alaptípusra is.

A minősítő szimbólumok a következők :

| | |
|----------|------------------|
| short | "rövid" |
| long | "hosszú" |
| unsigned | "előjel nélküli" |

A short az int típust minősítheti, a long szintén az int-et, valamint a double típust is. Az unsigned minősítő a char, ill. az int típusok előtt állhat, beleértve a már - short-tal vagy long-gal - minősített int típust.

| <u>Típus</u> | <u>Értelmezés</u> | <u>Méret (byte)</u> | <u>Minősítő</u> |
|---------------|-----------------------------|---------------------|-----------------|
| short integer | "Rövid" bináris egész szám. | 2 | short |
| long integer | "Hosszú" egész szám. | 4 | long |
| long double | "Hosszú" double szám | 10 | long double |

A számábrázolási tartományok a következők, az összes lehetséges esetet figyelembe véve :

| <u>Változótípus</u> | <u>Alsó határ</u> | <u>Felső határ</u> |
|---------------------|-------------------|--------------------|
| char : | -128 | 127 |
| unsigned char : | 0 | 255 |
| int : | -32 768 | 32 767 |
| unsigned int : | 0 | 65 535 |
| long int : | -2 147 483 648 | 2 147 483 647 |
| unsigned long : | 0 | 4 294 967 295 |
| float : | 3.4E - 38 | 3.4E + 38 |
| double : | 1.7E - 308 | 1.7E + 308 |
| long double : | 1.2E - 4932 | 1.2E + 4932 |

Példa a definícióra :

```
char c;  
int i,j;  
unsigned k;  
long l;  
float f;  
double d;
```

Konstansok :

```
Egész típusú      : -123, 12  
Oktális          : 034      / Mindig a 0 (nulla) karakterrel kezdődik !/  
Hexadecimális   : 0x1f  
Long             : 123L  
Lebegőpontos    : 123.45, -23.56, 0.12345E3  
Karakteres      : 'A'
```

Láthatjuk tehát, hogy mind tízes, mind nyolcas ill. tizenhatos számrendszerben megadhatunk konstansokat. Figyelembe kell venni azt a tényt, hogy oktális számok esetén a 08 szintaktikus hibát eredményez, hiszen nyolcas számrendszerben a legnagyobb számjegy a 7-es !

1.3 A printf() függvény leírás

Általános alak : **printf("<formátum vezérlő mező>"[,<változólista>]);**
Példa : printf("\n I értéke = %d",i);
Leírás : A formátum vezérlő mező állhat szövegből, karakteres állandókból ill. formátum vezérlő karakterekből. A karakteres állandók a '\ (back slash) - sel kezdődnek.

Alapvető karakteres állandók :

```
"\n" = Új sor  
"\t" = Tabulátor  
"\f" = Lapdobás
```

A formátum vezérlő karaktereket mindig a % jel előzi meg, utána következhet több karakter, ami utal a kiíratni kívánt változó típusára, és a kiíratás formátumára. A függvény működése során a formátumvezérlő karakterek helyére rendre behelyettesíti a változólista elemeit.

Fontos : Ha több formátumvezérlő karakter van, mint ahány változót megadtunk a listában, akkor azon vezérlőkarakterek helyén, amelyekhez nem tartozik változó, véletlen érték fog megjelenni. Ha hosszabb, mint ahány formátumvezérlő karakter van, akkor a lista végén szereplő változók értékei nem jelennek meg a kiíratásnál !

Alapvető formátum vezérlő karakterek :

%d = Integer
%ld = Long integer
%x = Hexadecimális egész
%o = Oktális egész
%f = Float
%lf = Double
%c = Egy karakter
%s = Karakterlánc (string)

A változólista a kiíratni kívánt változók neveit tartalmazza, vesszővel elválasztva egymástól. Mód van a kiíratni kívánt érték mezőszélességének megadására is. Amennyiben egy integert n számjegy szélességben akarunk kiíratni, akkor a megfelelő formátumvezérlés a következő : "%nd". Ebben az esetben az n számjegynél kisebb egészek balról szóközzel lesznek feltöltve. Például : a "%6d" formátumvezérlés a 123-at a következőképpen írja ki : $\vee\vee\vee123$.

1.4 A scanf() függvény leírása

Általános alak : *scanf("<formátum vezérlő mező>",<változó cím lista>);*

Példa : scanf("%f",&f);
Leírás : A formátum vezérlő mező formátum vezérlő karakterekből állhat. Ld.: a printf() függvény leírásánál.

Mivel a scanf függvény a változó(k)-ban el akarja helyezni a beolvasott értéket, ezért a változó(k) címeit kell átadni. Ennek a ténynek a részletes magyarázatára későbbiekben térünk ki ! A változó címét a változó neve elé írt & karakter segítségével tudjuk előállítani.

Amennyiben egyszerre több változóba akarunk értékeket beolvasni, úgy kétféle módszert alkalmazhatunk :

1.) A programban a formátumvezérlő karaktereket vesszővel választjuk el egymástól, és a program futása során a bebillentyűzött értékeket is vesszővel választjuk el .

Példa : int x, y, z ;
scanf("%d,%d,%d",&x,&y,&z); esetén

a beolvasás a következőképpen történik : 2,3,4 <ENTER> .

