

1. Példa

Egy kis C program a program felepitesenek bemutatasa: gömb felszinenek es terfogatanak kiszamitasa:

```
#include <stdio.h> // ebben a headerfajlban van adva a scanf es printf
// fuggvenyek prototipusa, deklaracioja, a
leforditott
// alakot pedig a Cx.LIB konyvtarak tartalmazzak mar
// kozvetlenul beszerkesztheto alakban (x helyen a
// memoriamodeltol fuggo betu all).
#define PI 3.141593 // konstans definialasa
main()
{
    float sugar, felulet, terfogat; // valos valtozok definialasa
    printf(" Gombfelulet es terfogat szamitasa\n\n"); //
tajekoztatoszoveg
    printf("A sug r= "); // sugar bekerese
    scanf("%f", &sugar); // & a cime operator, erre a cimre kerjuk be a
sugarat
    felulet = 4 * sugar * sugar * PI;
    terfogat = 4.0 / 3 * sugar * sugar * sugar * PI;
    // kiiratas 10 karakter szeles mezore, 3 tizedes pontossaggal,
lebegopontos (f) alakkal
    printf("\nFelulet= %10.3f, terfogat= %10.3f\n", felulet, terfogat);
}
```

2. Példa

```
/* Kis program kulonfele tipusu valtozok deklaralasara,
ertekuk bekeresere billentyuzetrol es az ertek kiiratasara
FIGYELEM, char tipusu valtozoban tarolhato zamerteket nem
lehet kozvetlenul beadni. A peldaban 2 byte-os int pufferbe
kerjuk be, majd tipuskinverzioval levagva 1 byte-osra,
helyezzuk el a signed, vagy unsigned char valtozokba */
/*
#include <stdio.h>
#include <conio.h>
main()
{
    char          betu;
    int           puffer; // ebbe irodik a negy byte, melybol csak egy
hasznos
    signed char   kisszam;
    unsigned char db;
```

```
int            egesz;
unsigned int   evszam;
long int       nagyszam;
unsigned long int nagypozitiv;
float          valos;
double         nagyvalos;
long double    nagyonnagyszam;
printf("Ertekadas kulonfele tipusu valtozoknak, majd kiiratas\n\n");
printf("Adj meg egy betut:");
betu = getch();
printf("\nAdj meg egy szamot -128 es +127 kozott :");
scanf("%d",&puffer); kisszam = (signed char) puffer;
printf("Adj meg egy szamot 0-255 kozott:");
scanf("%u",&puffer); db = (unsigned char) puffer;
printf("Adj meg egy egesz szamot -32768 es 32767 kozott:");
scanf("%d",&egesz);
printf("Adj meg egy evszamot:");
scanf("%u",&evszam);
printf("Adj meg egy 8 jegyu negativ szamot:");
scanf("%ld",&nagyszam);
printf("Adj meg egy 9 jegyu pozitiv szamot:");
scanf("%lu",&nagypozitiv);
printf("Adj meg egy valos szamot:");
scanf("%f",&valos);
printf("Adj meg egy 16 jegyu valos szamot:");
scanf("%lf",&nagyvalos);
printf("Adj meg egy 24 jegyu negativ szamot:");
scanf("%Lf",&nagyonnagyszam);
printf("\n\nKiiratasok kovetkeznek:\n");
printf("\nA betu: %c, kodja: %d",betu,betu);
printf("\nA szam -128 es 127 kozott= %d",kisszam);
printf("\nA szam 0-255 kozott= %u",db);
printf("\nAz egesz -32768 es 32767 kozott= %d",egesz);
printf("\nAz evszam csak pozitiv lehetett, = %u",evszam);
printf("\nA 8 jegyu negativ szam= %ld",nagyszam);
printf("\nA 9 jegyu pozitiv szam= %lu",nagypozitiv);
printf("\nA valos lebegopontosan= %f, kitevos alakban=
%e",valos,valos);
    printf("\nA duplapontos valos= %lf, kitevosen=
%le",nagyvalos,nagyvalos);
    printf("\nA 24 jegyu negativ= %Lf, kitevosen= %Le",nagyonnagyszam,
nagyonnagyszam);
    // valos szamoknal elofordulhat elteres a szamabrazolas nem kello
pontossaga miatt
}
```

3. példa

Példa a mutató használatára: a sima változoba beírt értéket mutató segítségével elérve módosítjuk, majd sima változoként és mutatóval elért változoként is kiírjuk.

```
#include <stdio.h>
main()
{
    int valt;
    int *valtramutato;
    printf("Adj értéket a változonak:");
    scanf("%d", &valt);
    valtramutato = &valt;
    *valtramutato = *valtramutato + 10; // valt tartalmát növeltük 10-zel
    printf("\nA 10-zel növelt érték változoként elérve: valt= %d", valt);
    printf("\nUgyanaz mutatóval elérve: *valtramutato= %d", *valtramutato);
}
```

4. Példa

Az alábbi kis program C kifejezésekre, operatorokra ad példát

```
#include <stdio.h>
#include <conio.h>
main()
{
    short int a, b, c, d, e;
    float p, r;
    a = 32000; b = 30000; c = a + b;
    printf("a=32000, b=30000, a + b= %d miért nem 62000?? ", c); // nem
    jelez hibát!
    a = 7; b = 2; c = a/b;
    printf("\nA / egészek esetén egészszosztás! Pl. a=7, b=2, c= a/b =
    %d", c);
    a = b = 9;
    printf("\nTöbbszörös értékadás lehetséges: a=b=9; a= %d lesz", a);
    printf("\nGyorsabb programot eredményeznek az értékadó operatorok:");
    printf("\n +=, -=, *=, /=, %=");
    a = 10; a = a + 3; printf("\n a = 10, a = a + 3; a = %d", a);
    a = 10; a += 3; printf("\n a = 10, a += 3; a = %d", a);
    a = 7; a = a * 2; printf("\n a = 7; a = a * 2; a = %d", a);
    a = 7; a *= 2; printf("\n a = 7; a *= 2; a = %d", a);
    printf("\n\nLeptető operatorok mellekhatással bírnak:");
    a = 5; b = 2;
    c = ++a * --b;
    printf("a =5; b = 2; c = ++a * --b");
    printf("\nEredmény: a= %d, b= %d, c=%d", a, b, c);
    a = 5; b = 2;
```

```

++a; --b; c = a * b;
printf("\nMegfelel ++a; --b; c = a * b utasitasoknak,");
printf("\naz eredmeny: a=%d, b=%d, c=%d lesz",a,b,c);
a =3; b= 6; c = 10; c += ++a + b--;
printf("\n\n a=3; b=6; c=10; c += ++a + b--");
printf("\nEredmeny: a=%d; b=%d, c=%d",a,b,c);
a =3; b =6; c = 10; ++a; c += (a+b); b--;
printf("\nMegfelel a ++a; c += (a+b); b-- utasitasoknak:");
printf("\naz eredmeny: a=%d, b=%d, c=%d lesz", a, b, c);
printf("Logikai kifejezesekben a relaciooperatorok magasabb\n");
printf("precedenciajuak, mint az &&, || logikai muveletek!\n");
printf("Ezert sok zarojelezes elmaradhat, lasd alabbi peldat:\n");
c = (25+3 < 45 && 12 > 7);
b = ( ( (25+3)<45 ) && ( 12>7 ) );
printf("\n c = (25+3 < 45 && 12 > 7) es b = ( ( (25+3)<45 ) && ( 12>7 )
)");
printf("\n egyenertekuek, c=%d, b=%d, igazak",c,b);
}

```

5. példa

```

#include <stdio.h>
#include <math.h>
void main(void)
{
int n,i;
float xalso, xfelso,x,y;
float deltax, nevező;
printf("      Függvénytáblázat készítése");
printf("\naz y= Gyök(x)*Sin(2*x)/((x-5)*(x-2)) függvényhez");
printf("\nTesztadatok: xalso= -2; xfelso= 6; lépésközök száma= 16");
printf("\nA számköz alsó határa= ");
scanf("%f", &xalso);
printf("\nA számköz felső határa= ");
scanf("%f", &xfelso);
printf("\nLépésközök száma= ");
scanf("%d", &n);
deltax = (xfelso-xalso) / n;
printf("\n\n      x      y(x)");
for (i=0; i<=n; i++)
{
x=xalso+i*deltax;
printf("\n%10.3lf",x);
if (x < 0)
{
printf("      Gyök alatt negativ szám!");
}
}
}

```

```
else
{
    nevező = (x-5) * (x-2);
    if (nevező == 0)
    {
        printf("    A nevező értéke nulla!");
    }
    else
    {
        y=sqrt(x)*sin(2*x)/nevező;
        printf("%10.3lf", y);
    }
}
}
```

From:

<https://edu.iit.uni-miskolc.hu/> - **Institute of Information Science - University of Miskolc**

Permanent link:

https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szamitastechnika:c_-_bevezetes

Last update: **2022/09/05 21:46**

