

## Programozás alapjai

### 2. gyakorlat

#### Első számítást végző C programok

1. Készítsünk kalkulátort. Adjon meg két számot, és végezzen el rajtuk valamilyen aritmetikai műveletet. Az eredményt írassa ki a képernyőre. Módosítsa úgy a programot, hogy kérje be a felhasználótól a számokat. Egész és valós értékekkel is próbálják ki!

Variációk:

1. Változók használata nélkül
2. Változók inicializálásával
3. Változók értékének beolvasásával
4. Különböző típusú változókkal

Aritmetikai operátorok: +, -, /, \*, %

*Figyelem! Két int érték osztásának eredménye is int. Ha azt szeretnénk, hogy az eredmény valós szám legyen, az osztás egyik operandusát explicite valós számra kell konvertálnunk.*

I/O függvényekben használható formátumspecifikációk (szigorú típusjegyzetetés!):

*int - %d, float - %f, double - %lf, valós 2 tizedesjeggyel - %.2f*

```
#include <stdio.h>                                /* Változók használata nélkül */
int main()
{
    printf("\nSzámológép\n");
    printf("%d * %d = %d\n", 7, 3, 7*3);
    return 0;
}
```

```
#include <stdio.h>                                /* Változók inicializálásával */
int main()
{
    int a = 5, b = 2;
    double c;
    printf("\nSzámológép\n");
    c = a / b;                                     /* helyesen: c = (double) a / b; */
    printf("%d / %d = %.2f\n", a, b, c);
    return 0;
}
```

```
#include <stdio.h>                                /* Változók értékének beolvasásával */
int main()
{
    int a, b, c;
    printf("\nSzámológép\n");
    scanf("%d, %d", &a, &b); // &: címképző operátor, a változó memóriacíme
    c = a % b;
    printf("%d %% %d = %d\n", a, b, c);
    return 0;
}
```

Figyeljék meg a scanf függvény működését: space vagy újsor karakterig olvas! Az input adatok egy ideiglenes tároló területre kerülnek. Innen csak annyi karaktert dolgoz fel a függvény, amennyit értelmezni képes a megadott formátum szerint.

## 2. Megadott sugarú kör területének és kerületének a kiszámítása.

```
#include <stdio.h>
int main()
{
    int r;
    double k, t, pi = 3.14;          /* tizedespont és nem tizedesvessző */
    printf("\nAdja meg a kör sugarát: ");
    /* Az r változó futási időben kap értéket, nem fordítási időben */
    scanf("%d", &r);                  /* az r változó címe: &r */
    t = r*r*pi;
    k = 2*pi*r;
    printf("\nA kör területe: %f\n", t);
    printf("\nA kör kerülete: %f\n", k);
    return 0;
}
```

### **Konstans megadása:**

const típusminősítővel:      const double pi = 3.14;  
konstans makróval:            #define PI 3.14  
A math.h standard header állományban definiálva van a PI konstans: M\_PI

3. Írjon C programot, amely kiszámítja a háromszög területét oldalainak megadásával. Használja a Hérón képletet:  $t^2 = s(s-a)(s-b)(s-c)$ , ahol  $a$ ,  $b$ ,  $c$  a háromszög oldalainak a hossza és  $s$  a terület fele. Ügyeljen a változók típusára! A terület meghatározásához használja a négyzetgyökvonást végrehajtó függvényt: **double sqrt(double num)**. Ennek deklarációját a *math.h* standard header állomány tartalmazza.

```
#include <stdio.h>
#include<math.h>
int main()
{
    int a, b, c;
    double t, t2, s;
    printf("\n A háromszög oldalai: ");
    scanf("%d, %d, %d", &a, &b, &c);
    s = (double)(a + b + c) / 2; //egész értékek osztásakor az eredmény is egész
    t2 = s*(s-a)*(s-b)*(s-c);
    t = sqrt(t2);
    printf("\nA háromszög területe: %.2f, kerülete: %d\n", t, a+b+c);
    return 0;
}
```

Egy háromszög bármely két oldalának összege nagyobb a harmadiknál. Érvényes adatok például:  $a=6$ ,  $b=8$ ,  $c=10$ . Ekkor a háromszög kerülete 24, területe 24.

Plusz feladat: Módosítsa úgy a programot, hogy az adatok beolvasása után ellenőrizze a háromszög egyenlőtlenség teljesülését.

**Figyelem!** Függvényhíváskor a fordító ellenőrzi a hívási argumentumok számát és típusát. Ezeknek meg kell egyezniük a függvény deklarációban (és definícióban) megadott paraméterlistával. A függvény argumentumának típusát megpróbálja automatikusan konvertálni a függvény definíció szerinti típusra és a függvény visszatérési értékét is automatikusan átkonvertálja az eredményt tároló változó típusára. Az *sqrt()* függvény definíciója szerint egy *double* típusú paramétert vár és *double* típusú eredményt szolgáltat.

4. Írjon C programot, amely a derékszögű háromszög két befogójának ismeretében kiszámítja az átfogóját. A két befogó értékét a felhasználó adja meg. A négyzetreemeléshez használja a *math.h* -ban deklarált **double pow(double, double)** függvényt.

```
#include <stdio.h>
#include<math.h>
int main()
{
    int a, b;
    double c;
    printf("\n A háromszög 2 befogója: ");
    scanf("%d, %d", &a, &b);
    c = sqrt(pow(a,2)+pow(b,2));
    printf("\n A háromszög átfogója: %lf", c);
    return 0;
}
```

5. A **double fabs(double)** függvény segítségével állapítsa meg egy beolvasott szám abszolút értékét.

```
#include <stdio.h>
#include<math.h>
int main()
{
    int x;
    printf("\nAdj meg egy számot: ");
    scanf("%d", &x);
    printf("\n%d abszolút értéke: %d", x, fabs(x));
    return 0;
}
```

6. Írjon C programot, amely felcserélni két egész szám értékét. A két számot a felhasználó adja meg.

```
#include <stdio.h>          /* Megoldás: segédváltozó használatával */
int main()
{
    int a, b, tmp;
    printf("\n Adj meg két számot: \n");
    printf("a : ");
```

```

scanf("%d", &a);
printf("b : ");
scanf("%d", &b);
/* Változók értékének felcserélése */
tmp = a; a = b; b = tmp;
printf("\nCseré után a = %d, b = %d\n", a, b);
return 0;
}

```

Megoldás változatok:

- A két változó értékének beolvasását oldjuk meg egyetlen utasításban.
- Oldja meg a feladatot segédváltozó felhasználása nélkül.

### Adatbeolvasás egyszerű ellenőrzéssel

Hibás eredményt kapunk, ha a két számot rossz formátumban adjuk meg (nem vesszővel elválasztva), vagy ha nem egész értékeket adunk meg. Ezekben az esetekben a *scanf* függvény vizsgálatával kiszűrhető a hibás működés. Ugyanis a *scanf* függvény által visszaadott érték azt tartalmazza, hogy hány értéket dolgozott fel a függvény az input sorból.

```

if (scanf("%d,%d", &a, &b) == 2) {
    // helyes működés kódja
}
else {
    printf("Hibás adatok!\n");
}

```

Ellenőrzött adatbeolvasás elve (megvalósítás később):

- Adatbeolvasás után input buffer ürítése.
- Adatbeolvasás ismétlése amíg nem helyes (nem megfelelő típusú / formátumú vagy nem az elvárt tartományba esik) az input adat.

### Házi feladat:

1. Készítsen C programot, amely bekéri egy téglatest három oldalának hosszát és kiszámítja annak felszínét ( $A = 2 * (a*b + a*c + b*c)$ ) és térfogatát ( $V = a*b*c$ ). A felszín számításánál ügyeljen a helyes zárójelezésre!
2. Írjon C programot, amely beolvas egy valós (double típusú) számot és visszaadja a hozzá legközelebb eső kisebb ill. nagyobb egész számot. Felhasználandó függvények: *double ceil(double)* ; *double floor(double)*
3. Írjon C programot, amely beolvas két egész számot, és kiírja az osztás eredményét az alábbi formában: *osztandó / osztó = hányados, maradék x*. Ugyanezt oldja meg valós (double típusú) számokkal is. Az osztási maradék kiszámításához használja a *math.h* *double fmod(double, double)* függvényét.