A kimenő és a be- és kimenő argumentumok kezelése
A függvény(művelet)eknél a paraméterek kezelése érték szerint történik, amely csak bemenő módú paraméterek használatát teszi lehetővé. (Lásd pl. korábbi gyakorlaton a **double hatv(double x, int n)** függvényt.) Előfordulhat, hogy kimenő paraméterre is szükségünk van.

Példa: A csere függvényművelet feladata az argumentumok értékének megcserélése.

```
#include <stdio.h>

csere (int x, int y)
{
        int m;
        m=x;
        x=y;
        y=m;
}
main {
        int a,b;
        a=1;
        b=2;
        csere(a,b);
        scanf("a=%d , b=%d",a,b);
}
```

A fenti egy rossz változat, mert a csere függvény csak az a és b változók értékét kapta meg, és ezeket használta a lokális változókban. Át kell adni az a, b változók címét, hogy értékük megcserélhető legyen: csere(&a,&b). Ekkor a csere függvényművelet deklarációja:

```
csere (int *x, int *y)
{
        int m;
        m=*x;
        *x=*y;
        *y=m;
}
```

Ha nem használunk globális változókat, akkor kimenő módú argumentumok kezelésére kielégítő megoldást ad a következő séma is: a lokális változókban kialakuló értékeket a return utasítás előtt adjuk át az aktuális paramétereknek:

```
csinalvalamit (int *px, int *py)
{
        int x,y;        /*lokális változók*/
        …               /*itt alakul ki x és y értéke*/
        *px=x;
```

```
        *py=y;
        return;
}
```

Ha nem használunk globális változókat, akkor be- és kimenő módú argumentumok kezelésére kielégítő megoldást ad a következő séma is: az aktuális paraméterek értékét lokális változókba tesszük, majd kialakuló értékeket a return utasítás előtt adjuk át az aktuális paramétereknek:

```
csinalvalamit (int *px, int *py)
{
        int x,y;        /*lokális változók*/
        x=*px;
        y=*py;
        …               /*itt módosulhat x és y értéke*/
        *px=x;
        *py=y;
        return;
}
```

FELADAT: Másodfokú egyenlet valós gyökeinek meghatározása. Input: a,b,c valós számok. Output: x,y valós számok, a másodfokú egyenlet gyökei, ha léteznek, különben egy szöveg, hogy nincs valós gyök. A feladat megoldása olyan függvényművelettel adható meg, amelynek három bemenő paramétere van, az egyenlet együtthatói, és két kimenő paramétere van, a két valós gyök, továbbá a függvény logikai értékkel tér vissza, amely akkor és csak akkor lesz igaz, ha van valós gyök. A közismert megoldóképletet használjuk.

```
#include <stdio.h>
#include <math.h>

typedef enum {hamis, igaz} logikai; /*saját típus definiálása*/

logikai megoldo(double a, double b, double c, double *x1, double *x2)
{
double d; /*a diszkrimináns*/
logikai valos; /*van-e valós megoldás*/

valos=igaz;
if (a==0.0) {
        if (b==0.0) {
                valos=hamis;
        } else {
                *x2=*x1=-(c/b);
```

```
                }
        } else {
                d=b*b-4.0*a*c;
                if (d<0.0) {
                        valos=hamis;
                } else {
                        x1=(-b+sqrt(d))/(2.0*a);
                        x2=(-b-sqrt(d))/(2.0*a);
                        }
                }
return valos;
} /*a függvény vége*/

main()
{
double a,b,c,x,y;
printf("Kérem az egyenlet együtthatóit:");
scanf("%lf%lf%lf",&a,&b,&c);
if (megoldo(a,b,c,&x,&y)) {
        printf("Az egyenlet gyökei: %6.3lf és %6.3lf\n",x,y);
        } else {
        printf("Az egyenletnek nincs valós megoldása!\n");
}
}/*main vége*/
```