

# **JEGYZŐKÖNYV**

**Operációs rendszerek BSc**

**2021 tavaszi féléves feladat**

## A feladat leírása:

Írjon egy C programot, ami egy másodfokú egyenlet megoldóképletét reprezentálja osztott memória szegmens segítségével. A műveletvégzéshez szükséges adatokat egy bemeneti fájlból olvassa be, majd az adatokat és az eredményt adja vissza egy kimeneti fájlba. A Bemeneti ill. kimeneti fájl struktúrája kötött!

Példa a bemeneti és kimeneti fájl struktúrájára:

Bemeneti fájl:

i (A megoldani kívánt egyenletek száma)

a b c

Kimeneti fájl:

a b c x y (Az a,b,c jelzi a bemeneti adatokat, az x,y pedig a kimeneti eredményeket)

## A feladat elkészítésének lépései:

1. Létrehoztam egy „Bemenet.txt” fájlt. A feladat leírásának megfelelően.
2. Létrehoztam és megnyitottam a „Kimenet.txt” fájlt.

```
int main()
{
    int szamitasok_szama;
    int eredmeny;
    FILE *fp;
    double kimenet[2];
    fp= fopen("Kimenet.txt", "w");
    double bemenet[3];
    fp= fopen("Bemenet.txt", "r");
    if (fp < 0) {
        perror("Nem sikerult a megnyitas!\n");
        exit(-1);
    }
```

3. Beolvasom és eltárolom a számítások számát.

```
fscanf(fp, "%d", &szamitasok_szama); //Jelenleg az első sorban állunk és így be tudjuk olvasni az i adatot!
printf("%d db számításat kell elvégezni.\n", szamitasok_szama);
```

4. Megírom a másodfokú függvényt kiszámító algoritmust.

```
//Masodfokú egyenletszámító
int szamito(double bemenet[], double kimenet[]){
    int visszaad=0;
    double gyokos = sqrt((bemenet[1]*bemenet[1]) - (4*bemenet[0]*bemenet[2]));
    if (gyokos == 0){
        kimenet[0] = -(bemenet[1]/2*bemenet[0]);
        visszaad=1;
    } else if(gyokos > 0){
        kimenet[0] = -((bemenet[1]+gyokos)/(2*bemenet[0]));
        kimenet[1] = -((bemenet[1]-gyokos)/(2*bemenet[0]));
        visszaad=2;
    }
    return visszaad;
}
```

Készítette: Jordán Kornél

5. Megírom a végeredmény kiíró algoritmust.

```
//Végeredményt kiírja
void vegeredmeny_kiiró(int visszaad, double kimenet[]){
    if(visszaad==2){
        printf("Ket megoldas: %.2lf, %.2lf\n", kimenet[0], kimenet[1]);
    }else if(visszaad==1){
        printf("Egy megoldas: %.2lf\n", kimenet[0]);
    }else if(visszaad==0){
        printf("Nincs megoldas mert a gyokiel alatt nem lehet kisebb szam mint 0!\n");
    }
    return;
}
```

6. Megírom a végeredményt fájlba kiíró algoritmust.

```
//Végeredményt beírja a "Kimenet.txt"-be
void kiiras_txt_be(int visszaad, double kimenet[], double bemenet[]){
    FILE *fp2 = fopen("Kimenet.txt", "a");
    if (fp2 < 0) {
        perror("Nem sikerult megnyitni azt a file-t amibe kiirjuk az eredmenyeket!\n");
        exit(-1);
    }
    if(visszaad==2){
        fprintf(fp2, "a=%.2lf \t b=%.2lf \t c=%.2lf \t Elso eredmény x=%.2lf \t Masodik eredmény y=%.2lf\n", bemenet[0], bemenet[1], bemenet[2], kimenet[0], kimenet[1]);
    }else if(visszaad==1){
        fprintf(fp2, "a=%.2lf \t b=%.2lf \t c=%.2lf \t Egy eredmény x=%.2lf \n", bemenet[0], bemenet[1], bemenet[2], kimenet[0]);
    }else if(visszaad==0){
        fprintf(fp2, "a=%.2lf \t b=%.2lf \t c=%.2lf \t Nem talalhato megoldas, mert a gyokiel alatt nem lehet kisebb szam mint 0!\n", bemenet[0], bemenet[1], bemenet[2]);
    }
    fclose(fp2);
    return;
}
```

7. Meghívom a megírt algoritmusokat: 4.+5.+6. úgy, hogy minden számra meglegyenek hívva.

```
fscanf(fp, "%d", &szamitasok_szama); //Jelenleg az elso sorban allunk es igy be tudjuk olvasni az i adatot!
printf("%d db szamitast kell elvegezni.\n", szamitasok_szama);
printf("Jelenlegi adatok:\n");
for (int i=0; i<szamitasok_szama; i++){//Aktualis sor
    for (int j=0; j<3; j++){//Aktualis 3 adat (oszlop)
        fscanf(fp, "%lf", &bemenet[j]);
    }
    //Mindig az aktualis szamokkal dolgozunk, mert benne van az elso for ciklusban!
    eredmeny = szamito(bemenet, kimenet);
    vegeredmeny_kiiró(eredmeny, kimenet);
    kiiras_txt_be(eredmeny, kimenet, bemenet);
}
fclose(fp);
return 0;
```

**Futtatás eredménye:**

**Készítette: Jordán Kornél**

8. Lefuttatom a programot:

```
6 db szamitast kell elvegezni.  
Jelenlegi adatok:  
Ket megoldas: -4.79, -0.21  
Nincs megoldas mert a gyokjel alatt nem lehet kisebb szam mint 0!  
Nincs megoldas mert a gyokjel alatt nem lehet kisebb szam mint 0!  
Nincs megoldas mert a gyokjel alatt nem lehet kisebb szam mint 0!  
Ket megoldas: -6.00, -5.00  
Ket megoldas: -25.90, -0.60  
  
Process returned 0 (0x0)   execution time : 0.048 s  
Press any key to continue.
```

9. Megnézem a létrehozott „Kimenet.txt”-t

\*Kimenet – Jegyzettömb

Fájl	Szerkesztés	Formátum	Nézet	Súgó
a=2.00	b=10.00	c=2.00	Elso eredmény x=-4.79	Masodik eredmény y=-0.21
a=5.00	b=3.00	c=6.00	Nem található megoldas, mert a gyokjel alatt nem lehet kisebb szam mint 0!	
a=2.00	b=2.00	c=4.00	Nem található megoldas, mert a gyokjel alatt nem lehet kisebb szam mint 0!	
a=7.00	b=3.00	c=5.00	Nem található megoldas, mert a gyokjel alatt nem lehet kisebb szam mint 0!	
a=1.00	b=11.00	c=30.00	Elso eredmény x=-6.00	Masodik eredmény y=-5.00
a=2.00	b=53.00	c=31.00	Elso eredmény x=-25.90	Masodik eredmény y=-0.60

Készítette: Jordán Kornél