

# Programozás I.

Szekeres György

# Eddig tanultuk

- Kiírás képernyőre
- Ékezetes karakterek kezelése
- Változók használata
- Matematikai műveletek
- Beolvasás konzolról
- Float, double, char
- If-else elágazás
- For ciklus
- While, do-while ciklusok
- Switch-case elágazás
- Tömbök
- Karakterkezelő függvények
- Többdimenziós tömbök
- Struktúrák
- String típus
- String osztály tagfüggvények
- Kiíratás
- Véletlenszám generáltatása
- Fájlkezelés
- Tagfüggvények
- Mutat

# 1. feladat

- Egy 100m-es síkfutás versenyzőinek nevét tároljuk az indulok.txt állományban a nevezés sorrendjében.
- A versenyre összesen 19-en neveztek, de végül volt, aki visszalépett.
- A versenyzők nevét, rajtszámát, legjobb és aktuális eredményét kell tárolnunk.
- Az időeredményeket minden esetben tizedes pontossággal másodpercben kezeljük.
- Végezze el a következő feladatokat:

# 1. feladat

- 1. Hozzunk létre egy megfelelő adatszerkezetet az adatok tárolására!
- 2. Írjuk ki hány fő indult végül a versenyen!
- 3. Töltsük fel az adatszerkezetet adatokkal úgy, hogy a nevet és az eddigi legjobb időt fájlból olvassuk be, a rajtszámot (100-199 lehet!) Véletlenszámgenerátorral adjuk meg, az aktuális időt (10-20 másodperc) pedig ellenőrzötten kérjük be a billentyűzetről adott névhez.
- 4. Írassuk ki az adatokat táblázatosan!
- 5. Számítsuk ki az összesen hány másodpercet futottak!
- 6. Számoljuk meg hányan futottak jobbat az eddigi legjobbjuknál és írjuk is ki a nevüket és az új idejüket!
- 7. Számoljuk meg, hogy akik elmaradtak a legjobbjuktól, átlagosan mennyit futottak!
- 8. Írjuk ki az elsőként nevezett futó nevét, rajtszámát, aktuális idejét és az eltérést az eddigi legjobbjától (jobb negatív érték)!
- 9. Mentsük a feltöltött adatokat eredmény.txt néven

# 1. feladat

```
#include<iostream>
#include<iomanip>
#include<string>
#include<ctime>
#include<fstream>
using namespace std;
struct indulo
{
    string nev;
    float eddigi;
    int rajtszam;
    float ido;
};
```

```
int main()
{
    indulo sik[19];
    int i, jobb = 0, rosszabb=0, letszam=0;
    float elmaradt = 0, ossz = 0;
    srand((unsigned)time(NULL));
    ifstream verseny("indulok.txt");
    for (i = 0; i < 19 && !verseny.eof(); i++)// adatok feltöltése
    {
        verseny>> sik[i].nev;
        verseny >> sik[i].eddigi;
        sik[i].rajtszam = rand() % 100 + 100;
```



# 1. feladat

```
do
{
    cout << "\nAdja meg az " << sik[i].nev << " futo
eredmenyet: ";

    cin >> sik[i].ido;
    cin.ignore();

    if (sik[i].ido < 10 || sik[i].ido > 20)
        cout << "Hibas adat, az idő 10-20mp kozott
lehet!" << endl;

    } while (sik[i].ido < 10 || sik[i].ido > 20);

    letszam++;
}

verseny.close();
system("cls");
```

```
cout << "\nOsszesen " << letszam << " futo indult a versenyen." << endl;
cout.setf(ios::right);
for (i = 0; i < letszam; i++)
{
    cout << setw(20) << sik[i].nev << "\t" << sik[i].rajtszam << "\t\t" <<
sik[i].eddigj << " mp\t\t" << sik[i].ido << " mp" << endl;
    ossz += sik[i].ido;
}

cout << "\nA futok osszesen " << ossz << " masodpercet futottak" << endl;
system("cls");

cout << "\nOsszesen " << letszam << " futo indult a versenyen." << endl;
cout.setf(ios::right);
```

# 1. feladat

```
for (i = 0; i < letszam; i++)
{
    cout << setw(20) << sik[i].nev <<
"\t"<<sik[i].rajtszam << "\t\t" <<
    sik[i].eddigj << " mp\t\t" << sik[i].ido << " mp" <<
endl;
    ossz += sik[i].ido;
}

cout << "\nA futok osszesen " << ossz << "
masodpercet futottak" << endl;

ofstream kesz("eredmeny.txt");
```

```
if (kesz.is_open())
{
    for (i = 0; i < letszam; i++)
    {
        kesz << sik[i].nev << "\t";
        kesz << sik[i].eddigj << "\t";
        kesz << sik[i].rajtszam << "\t";
        kesz << sik[i].ido << endl;
    }
    kesz.close();

    cout << "\nA lista sikeresen elkeszult." << endl;
} else cout << "\nA lista letrehozasa sikertelen." << endl;

return 0;
}
```