

Azonosító: MAX02a ¹

Újtémakörök

- 📁 fájlkezelést *minimális*hibaelhárítással
- 📁 „nyíl-nyíl”(<<,>>)operátorok alkalmazása adatok beolvasására
- 📁 parancssorban megadott argumentumok kezelése

Feladat

Olvassunk be egy fájlból egész számok kategóriájú *rba*, majd keressük meg a vektorban valamely maximális elemét, és az eredményt írjuk ki szabványos outputra.

Megkötések:

- 📁 A vektort **fájlból** töltjük fel .
A fájl neve: **adat.dat**
- 📁 A fájlnevét **parancssorban** adjuk meg.
A parancsformája : **max02a adat.dat**
- 📁 A fájl **pontosan8** elemet tartalmaz.
- 📁 A vektor hossza **pontosan8**.

Megoldás

A feladat megoldása három fő részből áll:

- 📁 Adatok előkészítése
- 📁 Számítások elvégzése (*absztrakt megoldóprogram*)
- 📁 Eredmény megjelenítése

Adatok előkészítése

Itt biztosítjuk, hogy a feladat elvégzése *he* szükséges adatok megfelelő *ő* formában rendelkezésünkre álljanak.

Számítások elvégzése

Ez a rész teljességgel megegyezik a **MAX01 feladat** "Számítások elvégzése" című fejezetével.


Eredmény megjelenítése

Ez a rész teljességgel megegyezik a **MAX01 feladat** "Eredmény megjelenítése" című fejezetével.

¹ A feladat támaszkodik a MAX01 feladatra.

Parancssorbanmegadottargumentumok

A programszerkezete	Aparancs
<pre>int main(int argc, char *argv[]) { ... argv[1] ... }</pre>	max02aadat.dat


:

.....


.....

Fájlkezelés

Adatokbeolvasásafájlból

<pre>#include <fstream> ----- ifstream inp; inp.open(fájl neve); inp.close(); inp.fail(); (logikai érték) inp >> ahova_beolvas;</pre>	 :

Adatokkiírásafájlba

<pre>#include <fstream> ----- ofstream out; out.open(fájl neve); out.close(); out.fail(); (logikai érték) out << amit_kiír;</pre>	 :

MegoldóprogramC++ -ban

```

/*****
/*Maximumkeresés vektorban.
/*Program neve: Max02a.cpp
/*****

```


:

```

#include <iostream>
#include <fstream>
using namespace std;

int main(int argc, char *argv[])
{

```

:

.....

```
//Adatok előkészítése és megjelenítése
```

```

char barmi;
if (argc <=1){
    cout << "Hiányzó paraméter!" << endl;
    cin >> barmi;
    return 1;
}

```

:

.....

```
ifstream inp;
```

:

.....

```

inp.open(argv[1]);
if (inp.fail()){
    cerr << "A megadott fájlt nem találom!" << endl;
    cin >> barmi;
    return 2;
}

```

:

.....

²Igyjelöltük meg az új eszközt bevezető sorokat.

```
int v[8];
int n=8;
```



.....

.....

```
for (int j=0; j!=n; j++){
    inp >> v[j];
}
```



.....

.....

.....

```
inp.close();
```



.....

.....

.....

3

```
cout << "A vektor elemei: ";
for (int j=0; j!=n; j++){
    cout << v[j];
    if (j != (n-1))
        cout << ", ";
    else
        cout << "." << endl;
}
```



.....

.....

.....

.....

```
//Maximumkeresés
```

Ezarészteljesegésébenmegegyezika **MAX01feladat** " **Maximumkeresés**"részével

```
//Eredmény megjelenítése
```

Ezarészteljesegésébenmegegyezika **MAX01feladat** " **Eredményekmegjelenítése** " részével

³Ezarészmeggyezika **MAX01**programrésszel

Ateljesmegoldóprogram

```
/* **** */
/* Maximumkeresés vektorban. */
/* Vektor feltöltése fájlból. */
/* A fájl nevét parancssorból kérjük be. */
/* A fájl pontosan 8 elemet tartalmaz. */
/* A vektor hossza pontosan 8. */
/* Program neve: Max02a.cpp */
/* **** */
#include <iostream>
#include <fstream>
using namespace std;

int main(int argc, char *argv[])
{
    //Adatok előkészítése és megjelenítése
    char barmi;
    if (argc <=1){
        cout << "Hiányzó paraméter!" << endl;
        cin >> barmi;
        return 1;
    }
    ifstream inp;
    inp.open(argv[1]);
    if (inp.fail()){
        cerr << "A megadott fájlt nem találom!" << endl;
        cin >> barmi;
        return 2;
    }
    int v[8];
    int n=8;

    for (int j=0; j!=n; j++){
        inp >> v[j];
    }
    inp.close();
}
```

4

```
cout << "A vektor elemei: ";
for (int j=0; j!=n; j++){
    cout << v[j];
    if (j != (n-1))
        cout << ", ";
    else
        cout << "." << endl;
}

//Maximumkeresés
int k,i,max;
k=0;
i=0;
max=v[0];

while(k!=(n-1)){
    if (v[k+1] >= max){
        i=k+1;
        max=v[k+1];
    }
    k=k+1;
}

//Eredmény megjelenítése
cout << "A vektor egyik legnagyobb eleme: " << v[i] << ".";
cout << endl << "Ez a vektor " << (i+1) << ". eleme." << endl;
cin >> barmi;
return 0;
}
```



Saját jegyzeteim



⁴Ezarészmegegyezika **MAX01**program résszel.