

„Programozás” komplex beadandó feladat

*Készítette: Nagy Levente
Neptun-azonosító: uji470
E-mail: levente0517@gmail.com*

*Kurzuskód: IT-18PROGEG
Gyakorlatvezető neve: Dobrovszki Gergő Károly*

2024. december 28.

Tartalom

Felhasználói dokumentáció	3
Feladat	3
Futási környezet	3
Használat	3
A program indítása	3
A program használata billentyűzetről való bevitel esetén	3
A program használata fájlból való bevitel esetén	3
A program kimenete	4
Minta bemenet és kimenet	4
Hibalehetőségek	Hiba! A könyvjelző nem létezik.
Fejlesztői dokumentáció	5
Feladat	5
Tervezés	5
Specifikáció	5
Visszavezetés	5
Algoritmus	6
Fejlesztői környezet	6
Forráskód	6
Megoldás	7
Függvénystruktúra	7
A kód	7
A kód (magas szintű függvényekkel)	9
Tesztelés	10
Automatikus tesztek (Bíró)	10
Automatikus tesztek (Bíró, magas szintű függvényekkel)	12
Érvényes tesztesetek	13
Érvénytelen tesztesetek	13
Fejlesztési lehetőségek	13

Felhasználói dokumentáció

Feladat

Leghűvösebb települések legmelegebb napjai

A meteorológiai intézet az ország N településére adott M napos időjárás előrejelzést, az adott településen az adott napra várta legmagasabb hőmérsékletet.
Készíts programot, amely azokat a napokat, amelyeken a leghűvösebb településen a lehető legnagyobb az előre jelzett hőmérséklet!

Futási környezet

IBM PC, exe futtatására alkalmas, 64-bites operációs rendszer (pl. Windows 11). Nem igényel egeret.

Használat

A program indítása

A program az *Komplex\ConsoleApp1\bin\Debug\net8.0\ConsoleApp1.exe* néven található a tömörített állományban.

A program használata billentyűzetről való bevitel esetén

Az *Program.cs* fájl elindításával a program az adatokat a **billentyűzetről** olvassa be a következő sorrendben:

#	Adat	Magyarázat
1.	Települések száma (n)	Nemnegatív egész szám
2.	Napok száma (m)	Nemnegatív egész szám
3.	1. település, 1. napján mért hőmérséklet	Egész szám innentől
4.	1. település, 2. napján mért hőmérséklet	
...	...	
	1. település, m. napján mért hőmérséklet	
	2. település, 1. napján mért hőmérséklet	
	...	
	n. település, m. napján mért hőmérséklet	

A program használata fájlból való bevitel esetén

Lehetőségünk van az adatokat **fájlból** is megadni. Ekkor a programot *parancssorban* a következőképpen kell indítani, feltételezve, hogy a bemeneti fájlok mellette helyezkednek el:

Program.cs < bel.txt

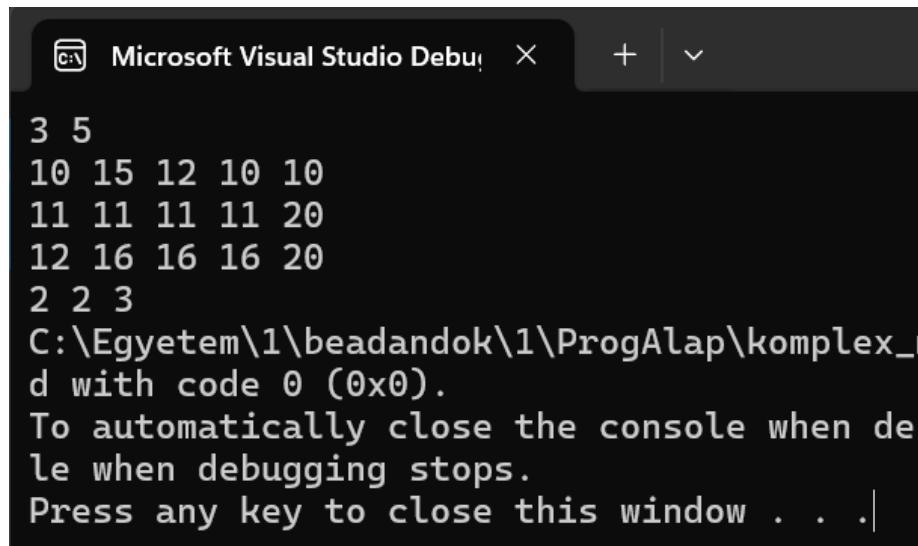
A fájl felépítésének a következő formai követelményei vannak. A fájl első sorában a települések száma (n) és a napok száma (m) van. A következő n sor mindenekében m darabszám szerepel, közülük az i-edik sorban a j-edik szám az i-edik helységben a j-edik sorszámú település megfigyelt hőmérséklet száma. Például:

```
3 5  
10 15 12 10 10  
11 11 11 11 20  
12 16 16 16 20
```

A program kimenete

A program kiírja azoknak a napoknak a mennyiséget és indexét , ahol a napi leghidegebb hőmérsékletek közül a legmelegebb a legkisebb.

Minta bemenet és kimenet



The screenshot shows a Microsoft Visual Studio Debug console window. The title bar says "Microsoft Visual Studio Debug". The console displays the following text:

```
3 5  
10 15 12 10 10  
11 11 11 11 20  
12 16 16 16 20  
2 2 3  
C:\Egyetem\1\beadandok\1\ProgAlap\komplex_r  
d with code 0 (0x0).  
To automatically close the console when del  
le when debugging stops.  
Press any key to close this window . . .|
```

Fejlesztői dokumentáció

Feladat

Leghűvösebb települések legmelegebb napjai

A meteorológiai intézet az ország N településére adott M napos időjárás előrejelzést, az adott településen az adott napra várt legmagasabb hőmérsékletet.

Készíts programot, amely azokat a napokat, amelyeken a leghűvösebb településen a lehető legnagyobb az előre jelzett hőmérséklet!

Tervezés

Specifikáció

Be: telepules ∈ N, napok ∈ N, matrix ∈ N[1..telepules,1..napok]

Sa: legnagyobbminimum ∈ N

Ki: db ∈ N, indexek ∈ N[1..db]

Ef: -

Uf: (legnagyobbminimum)=**MAX**(i=1..napok, **MIN**(j=1..telepules, matrix[j][i]).2) és

(db, indexek) = **KIVÁLOGAT**(j=1..napok, legnagyobbminimum = **MIN**(k=1..telepules, matrix[k][j]).2, j)

Visszavezetés

Maximumkiválasztás

i	i
e..u	1..napok
f(i)	MIN(i=e..u,f(i))

Minimumkiválasztás

i	j
e..u	1..települes
f(i)	matrix[j][i]

Kiválogatás

i	j
e..u	1..napok
T(i)	legnagyobbminimum = MIN(i=e..u,f(i))
f(i)	j

Minimumkiválasztás

i	k
---	---

e..u	1..települes
f(i)	matrix[k][j]

Algoritmus

```

telepules, napok, matrix[1..telepules, 1..napok], legnagyobbminimum, db = 1, indexek[1..db], tempmin, minimumnapok[1..napok]
i=1..napok
tempmin = matrix[i,i]
j=1..telepulesek
    if matrix[j,i] < tempmin then
        tempmin = matrix[j,i]
        minimumnapok[i] = tempmin
legnagyobbminimum = minimumnapok[1]
i=1..napok
    if minimumnapok[i] > legnagyobbminimum then
        legnagyobbminimum = minimumnapok[i]
i=1..napok
    if minimumnapok[i] = legnagyobbminimum then
        indexek[db] = i
        db++

```

Fejlesztői környezet

IBM PC, exe futtatására alkalmas operációs rendszer (pl. Windows 11 Home). Visual Studio 2022 (Version 17.2.3) fejlesztői környezet.

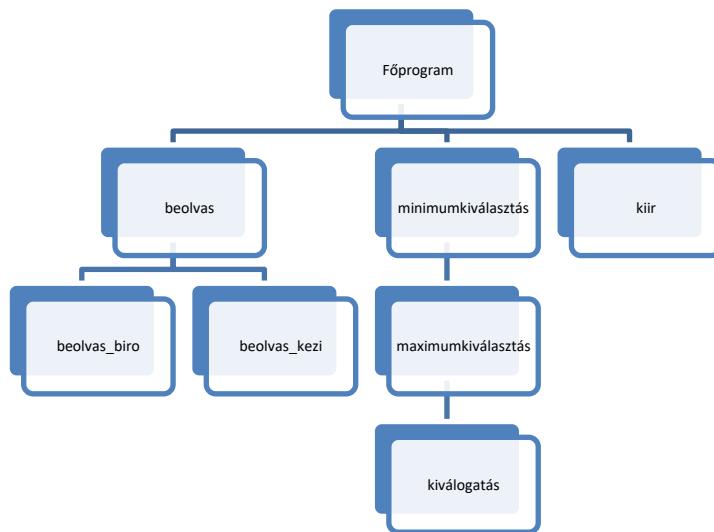
Forráskód

A teljes fejlesztői anyag –kicsomagolás után– az Komplex nevű könyvtárban található meg. A fejlesztés során használt könyvtár-struktúra:

Állomány	Magyarázat
Komplex\ConsoleApp1\bin\Debug\net8.0\A1B2C3.exe	futtatható kód (a futtatáshoz szükséges fájlokkal)
Komplex\ConsoleApp1\obj\	mappa fordításhoz szükséges kódokkal
Komplex\ConsoleApp1\Program.cs	C# forráskód
Komplex\ConsoleApp1\be1.txt	teszt-bemeneti fájl ₁
Komplex\ConsoleApp1\be2.txt	teszt-bemeneti fájl ₂
Komplex\ConsoleApp1\ki1.txt	tesztfájl-kimenet ₁
Komplex\ConsoleApp1\ki2.txt	tesztfájl-kimenet ₂
Komplex\doksi\uji470.docx	dokumentációk (ez a fájl)

Megoldás

Függvénystruktúra



A kód

A Program.cs fájl tartalma:

```
/*
Készítette: Nagy Levente
Neptun: Uji470
E-mail: levente0517@gmail.com
Feladat: Leghűvösebb település legmelegebb napjai
*/
```

```
namespace ConsoleApp1
{
    using System;

    internal class Program
    {
        static void Main(string[] args)
        {

            int telepules, napok;

            string[] temp = new string[2];

            temp = Console.ReadLine().Split(' ');

            telepules = int.Parse(temp[0]);
            napok = int.Parse(temp[1]);

            int[,] matrix = new int[telepules, napok];

            string[] temp2 = new string[napok];

            for (int i = 0; i < telepules; i++)
            {
                temp2 = Console.ReadLine().Split(' ');
                for (int j = 0; j < napok; j++)
                {
```

```

        matrix[i,j] = int.Parse(temp2[j]);
    }
}

//napi minimumok megállapítása

int[] napok_min = new int[napok];

for (int i = 0; i < napok; i++)
{
    int temp_min = matrix[0,i];
    for (int j = 1; j < telepules; j++)
    {
        if (matrix[j,i] < temp_min)
        {
            temp_min = matrix[j,i];
        }
    }
    napok_min[i] = temp_min;
}

//legnagyobb minimum

int legnagyobb_minimum = napok_min[0];

for (int i = 1; i < napok; i++)
{
    if (napok_min[i] > legnagyobb_minimum)
    {
        legnagyobb_minimum = napok_min[i];
    }
}

//kiválogatás mikor igaz

int[] vanilyen = new int[napok];
int t = 0;

for (int i = 0; i < napok; i++)
{
    if (napok_min[i] == legnagyobb_minimum)
    {
        vanilyen[t] = i + 1;
        t++;
    }
}

Console.WriteLine(t + " ");
for (int i = 0; i < t; i++)
{
    Console.Write(vanilyen[i] + " ");
}

//Console.ReadLine();
}
}
}
```

A kód (magas szintű függvényekkel)

A Program.cs fájl tartalma:

```
namespace ConsoleApp1
{
    using magas_szintű_mintamegvalósítások;
    internal class Program
    {
        static void Main(string[] args)
        {
            int telepules, napok;

            string[] temp = new string[2];

            temp = Console.ReadLine().Split(' ');

            telepules = int.Parse(temp[0]);
            napok = int.Parse(temp[1]);

            int[,] matrix = new int[telepules, napok];

            string[] temp2 = new string[napok];

            for (int i = 0; i < telepules; i++)
            {
                temp2 = Console.ReadLine().Split(' ');
                for (int j = 0; j < napok; j++)
                {
                    matrix[i, j] = int.Parse(temp2[j]);
                }
            }

            int[] napok_min = new int[napok];
            int bin = 0;

            //napi minimumok megállapítása
            for (int i = 0; i < napok; i++)
            {
                (bin, napok_min[i])= Mintak.Min(0, telepules - 1 , j => matrix[j, i]);
            }
        }
}
```

```

//legnagyobb minimum

(bin, int legnagyobb_minimum) = Mintak.Max(0, napok - 1, i => napok_min[i]);

//kiválogatás mikor igaz

    int[] vanilyen = Mintak.Kivalogat(napok_min, min => min == legnagyobb_minimum, (min, index) => index + 1);

    Console.WriteLine(vanilyen.Length);

    for (int i = 0; i < vanilyen.Length; i++)
    {
        Console.Write(" " + vanilyen[i]);
    }

    Console.ReadLine();
}

}
}

```

Tesztelés

Automatikus tesztek (Bíró)¹

¹ Példaképpen egy Mesteres képernyőkép van itt megadva. A végső beadandóban a Bíróból kell a képet kivágni, méghozzá az Eredmény menüpontból kiválasztva a megfelelő „Próbát”.

Total points: 100/

Test#	Point	Verdict	CPU time
1.1	4/4	Helyes	0.027 sec
2.1	4/4	Helyes	0.031 sec
3.1	4/4	Helyes	0.032 sec
4.1	4/4	Helyes	0.032 sec
5.1	4/4	Helyes	0.032 sec
6.1	5/5	Helyes	0.034 sec
7.1	5/5	Helyes	0.032 sec
8.1	5/5	Helyes	0.030 sec
9.1	5/5	Helyes	0.034 sec
10.1	5/5	Helyes	0.030 sec
11.1	5/5	Helyes	0.032 sec
12.1	5/5	Helyes	0.031 sec
13.1	5/5	Helyes	0.033 sec
14.1	5/5	Helyes	0.033 sec
15.1	5/5	Helyes	0.034 sec
16.1	6/6	Helyes	0.063 sec
17.1	6/6	Helyes	0.064 sec
18.1	6/6	Helyes	0.058 sec
19.1	6/6	Helyes	0.117 sec
20.1	6/6	Helyes	0.308 sec

Date of submission: 2024-12-28 14:15:13.0

Automatikus tesztek (Bíró, magas szintű függvényekkel)

Total points: 100/

Test#	Point	Verdict	CPU time
1.1	4/4	Helyes	0.027 sec
2.1	4/4	Helyes	0.031 sec
3.1	4/4	Helyes	0.032 sec
4.1	4/4	Helyes	0.032 sec
5.1	4/4	Helyes	0.032 sec
6.1	5/5	Helyes	0.034 sec
7.1	5/5	Helyes	0.032 sec
8.1	5/5	Helyes	0.030 sec
9.1	5/5	Helyes	0.034 sec
10.1	5/5	Helyes	0.030 sec
11.1	5/5	Helyes	0.032 sec
12.1	5/5	Helyes	0.031 sec
13.1	5/5	Helyes	0.033 sec
14.1	5/5	Helyes	0.033 sec
15.1	5/5	Helyes	0.034 sec
16.1	6/6	Helyes	0.063 sec
17.1	6/6	Helyes	0.064 sec
18.1	6/6	Helyes	0.058 sec
19.1	6/6	Helyes	0.117 sec
20.1	6/6	Helyes	0.308 sec

Date of submission: 2024-12-28 14:15:13.0

Érvényes tesztesetek

1. teszteset: bel.txt

Bemenet – nincs helyseg, nincs madarfaj	
3 5	
10 15 12 10 10	
11 11 11 11 20	
12 16 16 16 20	
Kimenet	
2 2 3	

2. teszteset: SAJÁT

Bemenet – 1 helység, 1 madárfaj, 1 darab	
4 4	
10 1 13 4	
13 34 15 0	
14 22 10 21	
10 5 50 45	
Kimenet	
2 1 3	

Érvénytelen tesztesetek

Billentyűzetes bevitel esetén

1. teszteset

Bemenet – szöveges adat	
N = 11tizenegy	
Kimenet	
Újrakérdezés:	
N =	

2. teszteset

Bemenet – Negatív szám	
N = -1	
Kimenet	
Újrakérdezés:	
N =	

...

Fejlesztési lehetőségek

1. Többszöri futtatás megszervezése
2. Települések és napok nevének megadása
3. Grafikus visszajelzés a számolás lépéseiről

