„Programozás”
beadandó feladat

Készítette: Bekovics Dániel
Neptun-azonosító: G9E74R
E-mail: g9e74r@inf.elte.hu

Kurzuskód: IT-18PROGEG
Gyakorlatvezető neve: Menyhárt László Gábor

2024. január 12.

Tartalom

Felhasználói dokumentáció

Feladat

Legváltozóbb települések

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

Készíts programot, amely megadja azokat a településeket, ahol az előrejelzés szerint egyik napról

a másikra a lehető legnagyobb a változás!

Futási környezet

ELF fájl futtatására alkalmas, 64-bites operációs rendszer. Nem igényel egeret.

Használat

A program indítása

A program az `Beadando3/bin/Release/Beadando3` néven található a tömörített állományban.

A program használata billentyűzetről való bevitel esetén

A `Beadando3` fájl elindításával a program az adatokat a **billentyűzet**ről olvassa be a következő sorrendben:

#	Adat	Magyarázat
1.	*Települések száma (N)*	Nemnegatív szám
2.	*Napok száma (M)*	Nemnegatív szám
3.	*1. településen az 1. napon mért legmagasabb hőmérséklet*	-50 és 50 közötti egész szám
4.	*1. településen az 2. napon mért legmagasabb hőmérséklet*	
…	*…*	
	1. településen az M. napon mért legmagasabb hőmérséklet	
	2. településen az 1. napon mért legmagasabb hőmérséklet	
	…	
	N. településen az M. Napon mért legmagasabb hőmérséklet	

A program használata fájlból való bevitel esetén

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

```
Beadando3 < be1.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 mindegyikében M darabszám szerepel, közülük az i-edik sorban a j-edik szám az i-edik településen a j-edik napon mért legmagasabb hőmérséklet. Például:

```
3 4
12 -7 33 -25
-41 18 4 9
37 -14 -3 22
```

A program kimenete

A program kiírja azoknak a településeknek a darabszámát és a sorszámait, ahol a legmagasabb volt két, egymást követő nap maximum hőmérsékletének a különbsége.

Minta bemenet és kimenet

```
btwPad @ ~/.../progalap/Beadando3
>_ bin/Release/net7.0/Beadando3
Települések száma = 3
Napok száma = 5
1. település 1. napján mért legnagyobb hőmérséklet = 10
1. település 2. napján mért legnagyobb hőmérséklet = 15
1. település 3. napján mért legnagyobb hőmérséklet = 12
1. település 4. napján mért legnagyobb hőmérséklet = 10
1. település 5. napján mért legnagyobb hőmérséklet = 10
2. település 1. napján mért legnagyobb hőmérséklet = 11
2. település 2. napján mért legnagyobb hőmérséklet = 11
2. település 3. napján mért legnagyobb hőmérséklet = 11
2. település 4. napján mért legnagyobb hőmérséklet = 11
2. település 5. napján mért legnagyobb hőmérséklet = 20
3. település 1. napján mért legnagyobb hőmérséklet = 25
3. település 2. napján mért legnagyobb hőmérséklet = 16
3. település 3. napján mért legnagyobb hőmérséklet = 16
3. település 4. napján mért legnagyobb hőmérséklet = 16
3. település 5. napján mért legnagyobb hőmérséklet = 20
2 db településen lesz egyik napról a másikra nagy hőmérséklet változás, melyek a következő sorszámúak: 2, 3
```

Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha bármelyik megadandó adat nem természetes szám. Hiba esetén a program azzal jelzi a hibát, hogy újra kérdezi azt.

Mintafutás hibás bemeneti adatok esetén:

```
btwPad @ ~/.../progalap/Beadando3
>_ bin/Release/net7.0/Beadando3
Települések száma = -1
0-nál nagyobb természetes szám kell!
Települések száma = ketto
0-nál nagyobb természetes szám kell!
Települések száma = öt
0-nál nagyobb természetes szám kell!
Települések száma = 2
Napok száma = 5
1. település 1. napján mért legnagyobb hőmérséklet = -75
-50 és 50 közötti egész szám kell! (A negatívak egybeirandók!)
1. település 1. napján mért legnagyobb hőmérséklet = 120
-50 és 50 közötti egész szám kell! (A negatívak egybeirandók!)
1. település 1. napján mért legnagyobb hőmérséklet = kilenc
-50 és 50 közötti egész szám kell! (A negatívak egybeirandók!)
1. település 1. napján mért legnagyobb hőmérséklet =
```

Fejlesztői dokumentáció

Feladat

Legváltozóbb települések

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

Készíts programot, amely megadja azokat a településeket, ahol az előrejelzés szerint egyik napról a másikra a lehető legnagyobb a változás!

Tervezés

Specifikáció

Bemenet:

$N \in \mathbb{N}$
$M \in \mathbb{N}$
$dat \in \mathbb{Z}^{N*M}$

Ki:

$cTemp \in \mathbb{N}$
$kivalasztottak \in \mathbb{Z}^{*}$

Előfeltétel:

$1 \leq N \leq 100$
$1 \leq M \leq 1000$
$\forall\, i \in [1{,}1000] : \forall\, j \in [1{,}1000] : dat_{i,j} \in [-50{,}50]$

Segéd:

$sor = \forall\, i \in [1{,}N] = dat_i$
$diffs = \forall\, i \in [2{,}M] : sor_i - sor_{i-1}$
$diffsMax = \forall\, i \in [1, N] : (, diffsMax_i) = MAX(k = 1..M, diffs_k)$
$(,max) = MAX\,(i = 1..N\,, diffsMax_i)$

Utófeltétel:

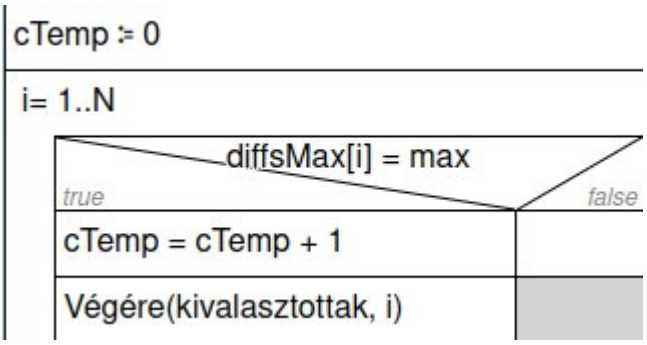
$(cTemp, kivalasztottak) = KIVÁLOGAT(i = 1..N, diffsMax_i = max, i)$

Visszavezetés

KIVÁLOGATÁS
db ~ cTemp
y ~ kivalasztottak
$T\,(i) \sim diffsMax_i = max$
f(*i*) ~ *i*
e..z ~ 1..N

MAX (diffsMax)
maxért ~ $diffsMax_i$
e..u ~ 1..*M*
$f(i) \sim diffs_k$
MAX(max)
maxért ~ *max*
e..u ~ 1..*N*
$f(i) \sim diffsMax_i$

Algoritmus



Fejlesztői környezet

ELF fájl futtatására alkalmas, 64-bites operációs rendszer. .NET Core 7.0.

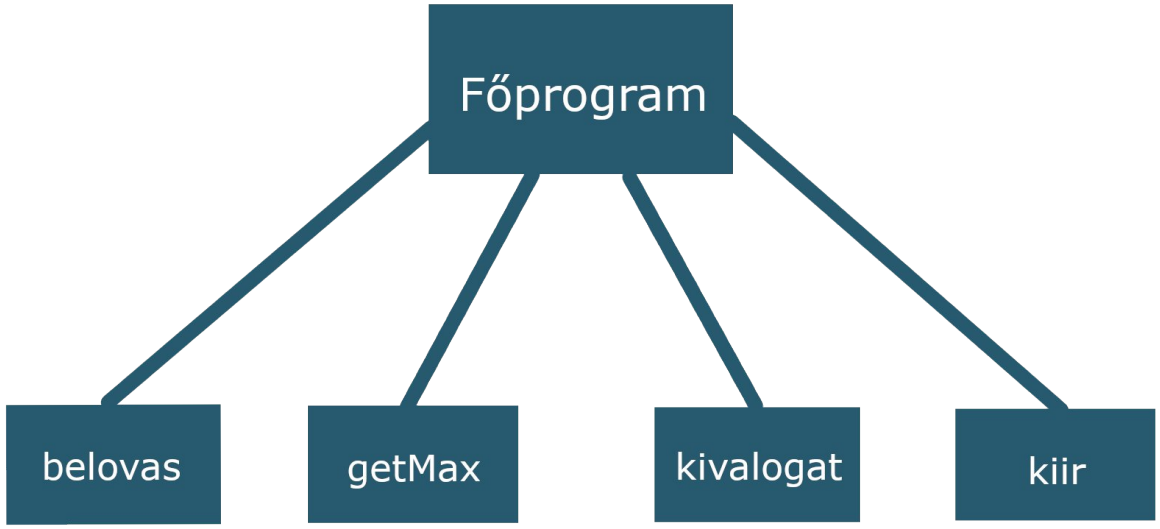
Forráskód

A teljes fejlesztői anyag –kicsomagolás után– az `Beadando3` nevű könyvtárban található meg. A fejlesztés során használt könyvtár-struktúra:

Állomány	Magyarázat
`Beadando3/bin/Release/net7.0/Beadando3`	futtatható kód (a futtatáshoz szükséges fájlokkal)
`Beadando3/obj/`	mappa fordításhoz szükséges kódokkal
`Beadando3/Program.cs`	C# forráskód
`Beadando3/bemenetek/be1.txt`	teszt-bemeneti fájl$_1$
`Beadando3/bemenetek/be2.txt`	teszt-bemeneti fájl$_2$
`Beadando3/bemenetek/be3.txt`	teszt-bemeneti fájl$_3$
`Beadando3/bemenetek/be4.txt`	teszt-bemeneti fájl$_4$
`Beadando3/bemenetek/be5.txt`	teszt-bemeneti fájl$_5$
`Beadando3/doksi/bd_complex_2.pdf`	dokumentációk (ez a fájl)

Megoldás

Függvénystruktúra



A kód

A `Program.cs` fájl tartalma:

```
 using System;
using System.Collections.Generic;
namespace Beadando
{
    class Program
    {
        public static void Main(string[] args)
        {
            int N;
            int[] diffsMax;
            beolvas(out N, out diffsMax);
```

```
        int max = getMax(diffsMax, N);
        int cTemp;
        List<int> kivalasztottak;
        kivalogat(N, diffsMax, max, out cTemp, out kivalasztottak);
        kiir(cTemp, kivalasztottak);

}

static int getMax(int[] arr, int N)
{
        int max = arr[0];
        for (int i = 1; i < N; i++)
        {
                if (arr[i] > max) max = arr[i];
        }
        return max;
}
static void beolvas(out int N, out int[] diffsMax)
{
        if (Console.IsInputRedirected)
        {
                int M;
                string[] fline = Console.ReadLine()!.Split(" ");
                int.TryParse(fline[0], out N);
                int.TryParse(fline[1], out M);
                diffsMax = new int[N];
                for (int i = 0; i < N; i++)
                {
                        string[] line = Console.ReadLine()!.Split(" ");
                        int[] sor = new int[M];
                        for (int j = 0; j < M; j++)
                        {
                                sor[j] = int.Parse(line[j]);
                        }
                        int[] diffs = new int[M];
```

```
            int k = 0;
            for (int j = 1; j < M; j++)
            {
                diffs[k] = Math.Abs(sor[j] - sor[j - 1]);
                k++;
            }
            diffsMax[i] = getMax(diffs, M);

        }
    }
    else
    {
        bool jo;
        do
        {
            System.Console.Write("Települések száma = ");
            jo = int.TryParse(Console.ReadLine()!, out N);
            jo = jo && N >= 1;
            if (!jo)
            {
                Console.ForegroundColor = ConsoleColor.Red;
                System.Console.WriteLine("0-Természetes szám kell!");
                Console.ResetColor();
            }
        }
        while (!jo);

        int M;
        do
        {
            System.Console.Write("Napok száma = ");
            jo = int.TryParse(Console.ReadLine()!, out M);
            jo = jo && M >= 1;
            if (!jo)
            {
```

```
                Console.ForegroundColor = ConsoleColor.Red;
                System.Console.WriteLine("0-nál nagyobb természetes szám kell!");
                Console.ResetColor();
            }
        } while (!jo);
        diffsMax = new int[N];
        for (int i = 0; i < N; i++)
        {
            int[] sor = new int[M];
            for (int j = 0; j < M; j++)
            {
                do
                {

                    System.Console.Write((i + 1) + ". település " + (j + 1) + ". napján mért legnagyobb hőmérséklet = ");
                    jo = int.TryParse(Console.ReadLine()!, out sor[j]);
                    jo = jo && sor[j] >= -50 && sor[j] <= 50;
                    if (!jo)
                    {
                        Console.ForegroundColor = ConsoleColor.Red;
                        System.Console.WriteLine("-50 és 50 közötti egész szám kell! (A negatívak egybeirandók!)");
                        Console.ResetColor();


                    }
                } while (!jo);
            }
            int[] diffs = new int[M];
            int k = 0;
            for (int j = 1; j < M; j++)
            {
                diffs[k] = Math.Abs(sor[j] - sor[j - 1]);
                k++;
            }
            diffsMax[i] = getMax(diffs, M);
```

```
                }
            }
        }
        static void kivalogat(int N, int[] diffsMax, int max, out int cTemp, out List<int> kivalasztottak)
        {
            cTemp = 0;
            kivalasztottak = new List<int>();

            for (int i = 0; i < N; i++)
            {
                if (diffsMax[i] == max)
                {
                    cTemp++;
                    kivalasztottak.Add(i + 1);
                }
            }
        }

        static void kiir(int cTemp, List<int> kivalasztottak)
        {
            if (Console.IsOutputRedirected)
            {
                System.Console.WriteLine(cTemp + " " + String.Join(" ", kivalasztottak));
            }
            else
            {
                System.Console.WriteLine(cTemp + " db településen lesz egyik napról a másikra nagy hőmérséklet változás, melyek a következő sorszámúak: " + String.Join(", ", kivalasztottak));
            }
        }

    }

}
```

Tesztelés

Érvényes tesztesetek

1. teszteset: be1.txt

Bemenet – *nincs helyseg, nincs madarfaj*
3 5 10 15 12 10 10 11 11 11 11 20 25 16 16 16 20
Kimenet
2 2 3

2. teszteset: be2.txt

Bemenet – *1 helység, 1 madárfaj, 1 darab*
1000 1000 33 37 ... 19 ... 7 9 12 ... -10
Kimenet
976 1 2 3 ... 1000

3. teszteset: be3.txt

Bemenet – *1 helység, 1 madárfaj, nincs madár*
3 4 12 -7 33 -25 -41 18 4 9 37 -14 -3 22
Kimenet
1 2

4. teszteset: be4.txt

Bemenet – ...
4 3 10 25 -45 6 17 38 -22 11 -9 33 -30 4
Kimenet
1 1

5. teszteset: be5.txt

Bemenet – ...

2 5 -7 19 -36 28 15 -42 9 -20 41 -3
Kimenet
1 1

Érvénytelen tesztesetek

Billentyűzetes bevitel esetén

6. teszteset

Bemenet – *szöveges adat*
Települések száma = 11tizenegy
Kimenet
Újrakérdezés: Települések száma =

7. teszteset

Bemenet – *Negatív szám*
Települések száma = -1
Kimenet
Újrakérdezés: Települések száma =

Fejlesztési lehetőségek

1. Többszöri futtatás megvalósítása
2. Települések nevének megadása
3. Átirányítás helyett parancssori argumentumként olvasott fájlnév