

„Programozási alapismeretek” beadandó feladat

Készítette: Perczel-Szabó Dániel

Neptun-azonosító: GQF4SF

E-mail: ggf4sf@inf.elte.hu

*Kurzuskód: **IP-18PROGEG***

Gyakorlatvezető neve: Geönczeöl Botond

2023. január 15.

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 kimenete	4
Minta bemenet és kimenet	4
Hibalehetőségek	4
Fejlesztői dokumentáció	5
Feladat.....	5
Tervezés	5
Specifikáció.....	5
Visszavezetés	5
Algoritmus	6
Fejlesztői környezet	6
Forráskód	7
Megoldás.....	7
Programparaméterek	7
Programfelépítés	7
Függvénystruktúra	8
A kód	8
Tesztelés	8
Érvényes tesztesetek	10
Érvénytelen tesztesetek	11
Fejlesztési lehetőségek	12

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

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

Használat

A program indítása

A program a

LegvaltozobbTelepulesek\LegvaltozobbTelepulesek\bin\Debug\LegvaltozobbTelepulesek.exe néven található a tömörített állományban.

A program használata billentyűzetről való bevétel esetén

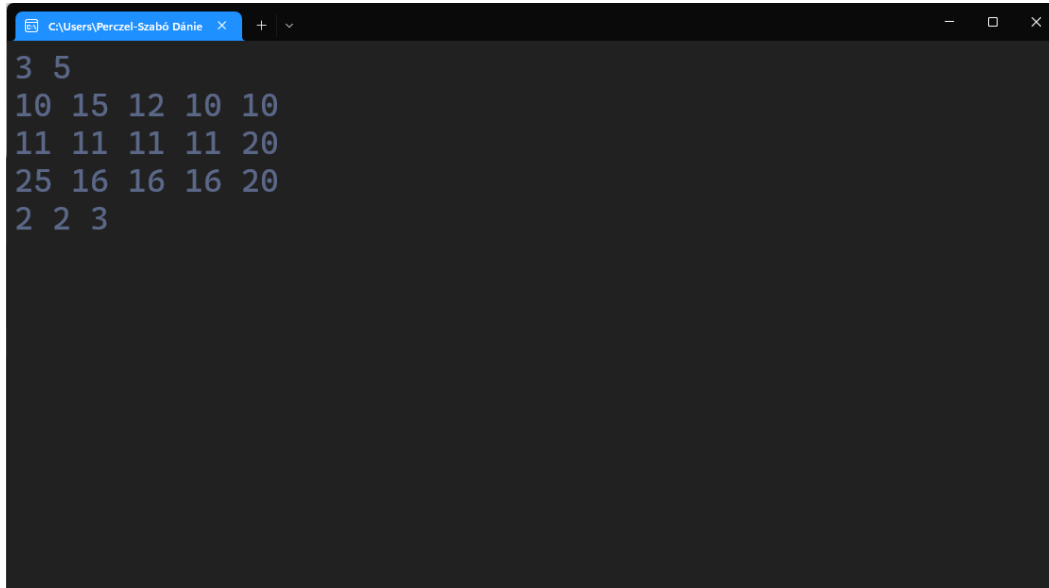
A LegvaltozobbTelepulesek.exe fájl elindításával a program az adatokat a **billentyűzetről** olvassa be a következő sorrendben:

#	Adat	Magyarázat
1.	Helységek száma (n)	Nemnegatív szám, 1-1000 között
2.	Napok száma (m)	Nemnegatív szám, 1-1000 között
3.	1. helységben az 1. napi hőmérséklet	Nemnegatív, -50-50 közötti szám innentől
4.	1. helységben a 2. napi hőmérséklet	
...	...	
	1. helységben az m . napi hőmérséklet	
	2. helységben az 1. napi hőmérséklet	
	...	
	n . helységben az m . napi hőmérséklet	

A program kimenete

A program kiírja azoknak a helységeknek a darabszámát és a sorszámaikat, ahol az előrejelzés szerint egyik napról a másikra a lehető legnagyobb a változás.

Minta bemenet és kimenet

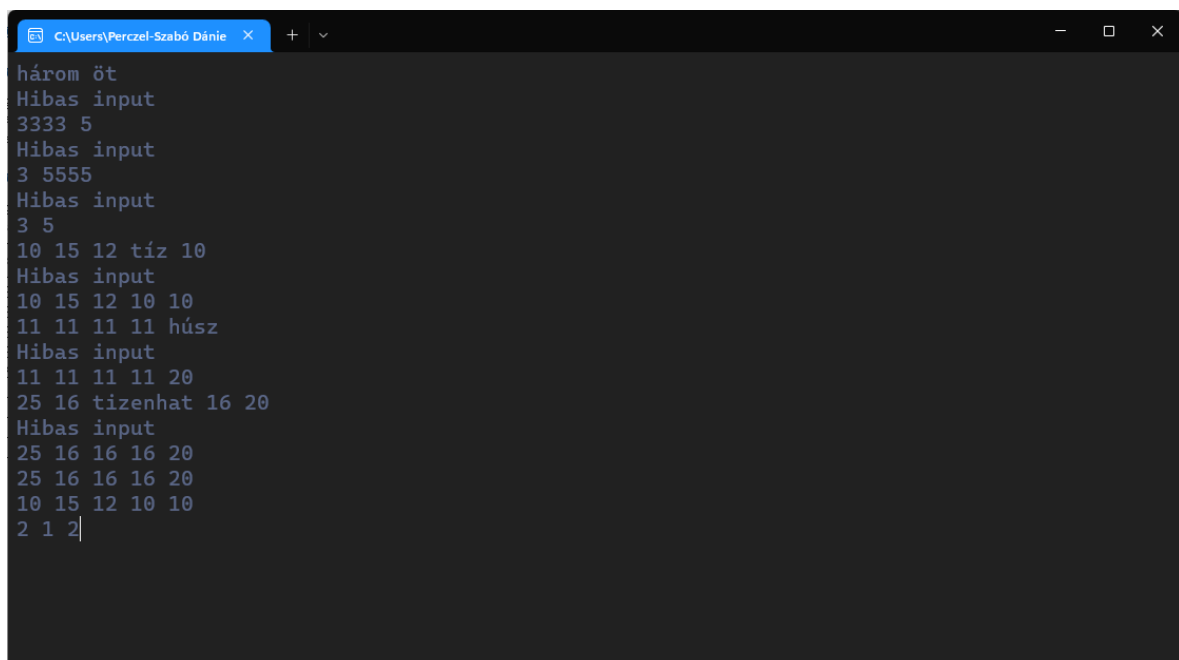


```
C:\Users\Perczel-Szabó Dániel >
3 5
10 15 12 10 10
11 11 11 11 20
25 16 16 16 20
2 2 3
```

Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha bármelyik megadandó adat nem természetes szám vagy az adott határokon kívülre esik. Hiba esetén a program azzal jelzi a hibát, hogy újra kérdezi az azt tartalmazó adathalmazt (az első sort vagy a hőmérsékleteket).

Mintafutás hibás bemeneti adatok esetén:



```
C:\Users\Perczel-Szabó Dániel >
három öt
Hibas input
3333 5
Hibas input
3 5555
Hibas input
3 5
10 15 12 tíz 10
Hibas input
10 15 12 10 10
11 11 11 11 húsz
Hibas input
11 11 11 11 20
25 16 tizenhat 16 20
Hibas input
25 16 16 16 20
25 16 16 16 20
10 15 12 10 10
2 1 2
```

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

Be: $n, m \in \mathbb{N}, H_{1..n, 1..m} \in \mathbb{N}^{n \times m}$

Ki: $n, m \in \mathbb{N}, H_{1..n, 1..m} \in \mathbb{N}^{n \times m}$

Ef: $1 \leq n \leq 1000$ és $1 \leq m \leq 1000$ és $-50 \leq H_{-}(i, j) \leq 50$

Uf: $(t, \text{sorszámok}) = \text{Kiválogat}_{i=1}^n i$ és $jó(i)$

$jó: \mathbb{N} \rightarrow \mathbb{L}$ és

$jó(i) = \forall_{j=2}^m (|H_{i,j} - H_{i,j-1}| = \text{maxDif}(i, j))$ és

$\text{maxDif}: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ és

$\text{maxDif}(i, j) = \text{Max}_{j=2}^m (|H_{i,j} - H_{i,j-1}|)$

Visszavezetés

Kiválogatás

$y \sim \text{sorszámok}$
 $T(x_i) \sim jó(i)$

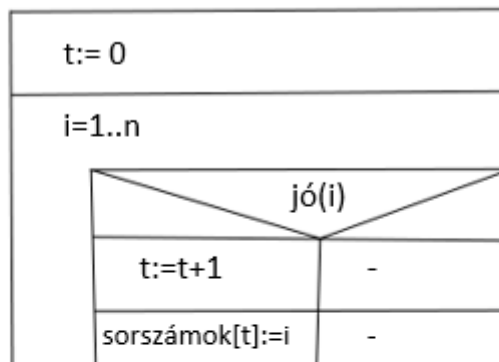
(Optimista) eldöntés

$i \sim j$
 $1..n \sim 1..m$
 $T(x_i) \sim |H_{i,j} - H_{i,j-1}| = \text{maxDif}(i, j)$

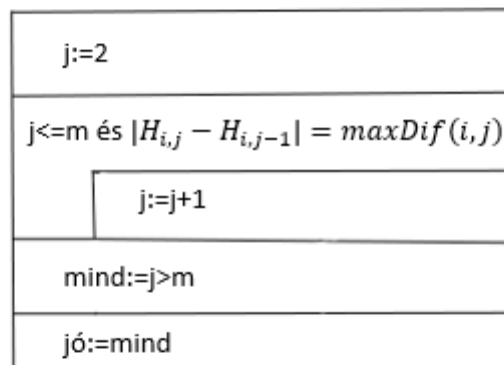
Maximumkiválasztás

$1..n \sim 1..m$
 $\text{maxÉrt} \sim \text{maxDif}$

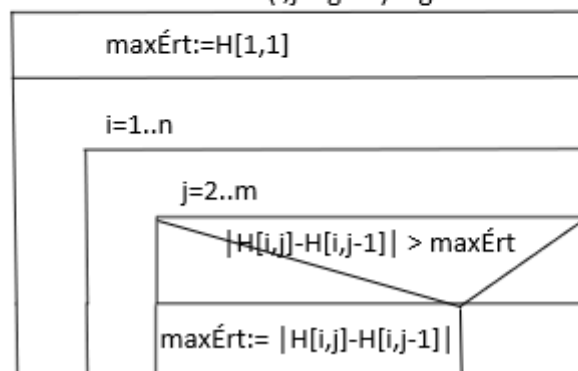
Algoritmus



jó(i: Egész): Logikai



maxDif(i,j: Egész): Egész



Fejlesztői környezet

PC, exe futtatására alkalmas operációs rendszer (Windows 11). Visual Studio 2022 (Version 17.4.2) fejlesztői környezet.

Forráskód

A teljes fejlesztői anyag –kicsomagolás után– az A1B2C3 nevű könyvtárban található meg. A fejlesztés során használt könyvtár-struktúra:

Állomány	Magyarázat
LegvaltozobbTelepulesek\LegvaltozobbTelepulesek\bin\Debug\LegvaltozobbTelepulesek.exe	futtatható kód (a futtatáshoz szükséges fájlokkal)
LegvaltozobbTelepulesek\LegvaltozobbTelepulesek\obj\	mappa fordításhoz szükséges kódokkal
LegvaltozobbTelepulesek\LegvaltozobbTelepulesek\Program.cs	C# forráskód
LegvaltozobbTelepulesek\LegvaltozobbTelepulesek\teszt1.txt	tesztbemenetet tartalmazó fájl
LegvaltozobbTelepulesek\LegvaltozobbTelepulesek\teszt2.txt	tesztbemenetet tartalmazó fájl
LegvaltozobbTelepulesek\LegvaltozobbTelepulesek\teszt3.txt	tesztbemenetet tartalmazó fájl
LegvaltozobbTelepulesek\LegvaltozobbTelepulesek\teszt4.txt	tesztbemenetet tartalmazó fájl
LegvaltozobbTelepulesek\LegvaltozobbTelepulesek\teszt5.txt	tesztbemenetet tartalmazó fájl
LegvaltozobbTelepulesek\LegvaltozobbTelepulesek\Programozás komplex beadandó dokumentáció.docx	dokumentációk (ez a fájl)

Megoldás

Programparaméterek

Változó

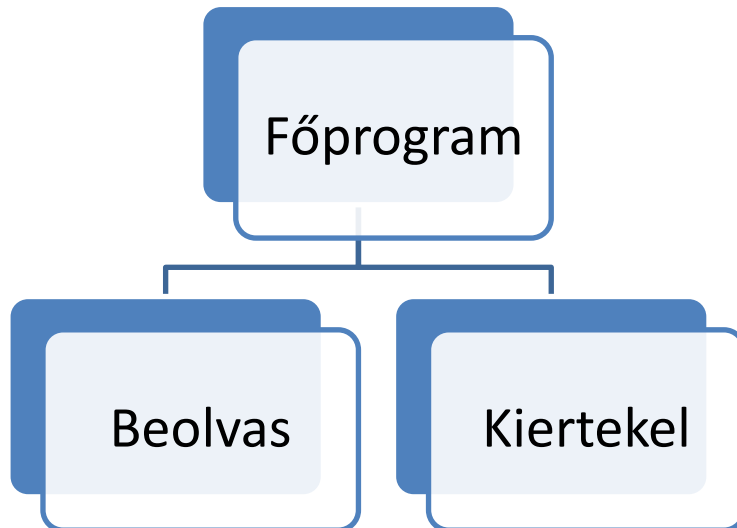
temps : Kétdimenziós Tömb (1..n,1..m:Egész)
places : Egész
days : Egész

Programfelépítés

A program által használt modulok (és helyük):

Program.cs – program, a forráskönyvtárban
LegvaltozobbTelepulesek.sln – program 'megoldás fájl', a forráskönyvtárban
LegvaltozobbTelepulesek.csproj – program 'projekt fájl', a forráskönyvtárban

Függvénystruktúra



A kód

A Program.cs fájl tartalma:

```
/*
 * Készítette: Perczel-Szabó Dániel
 * Neptun: GQF4SF
 * E-mail: gqf4sf@inf.elte.hu
 * Feladat: Legváltozóbb települések
 */

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace LegvaltozobbTelepulesek
{
    class Program
    {
        public static int[,] temps;
        public static int places;
        public static int days;

        static void Main(string[] args)
        {
            Beolvas();
            Kiertekel();

            Console.ReadKey();
        }

        static void Beolvas()
        {
            string[] firstLine;
            bool placesNum = false;
            bool daysNum = false;

            do
            {
                firstLine = Console.ReadLine().Split(' ');
```



```

        placesNum = int.TryParse(firstLine[0], out places);
        daysNum = int.TryParse(firstLine[1], out days);

        if(!placesNum || !daysNum || places < 1 || places > 1000 || days < 1 || days
> 1000)
            Console.WriteLine("Hibas input");
    }
    while (!placesNum || !daysNum || places < 1 || places > 1000 || days < 1 || days
> 1000);

    temps = new int[places, days];

    bool allGood=false;

    while (!allGood)
    {
        allGood = true;

        for (int i = 0; i < places; i++)
        {
            string[] mfl = Console.ReadLine().Split(' ');

            bool tempNum = false;

            for (int j = 0; j < days; j++)
            {
                tempNum = int.TryParse(mfl[j], out temps[i, j]);

                if (!tempNum || temps[i,j]<-50 || temps[i, j] > 50)
                {
                    allGood = false;
                    break;
                }
            }

            if (!allGood)
                break;
        }

        if(!allGood)
            Console.WriteLine("Hibas input");
    }
}

static int MaxDif()
{
    int maxdif = 0;

    for(int i=0;i<places;i++)
        for(int j = 1; j < days; j++)
        {
            if (Math.Abs(temps[i, j] - temps[i, j - 1]) > maxdif)
                maxdif = Math.Abs(temps[i, j] - temps[i, j - 1]);
        }

    return maxdif;
}

static void Kiertekel()
{
    int[] res = new int[days];
    int db = 0;

```

```

int maxDif = MaxDif();

for(int i=0;i<places;i++)
    for(int j = 1; j < days; j++)
    {
        if(Math.Abs(temps[i, j] - temps[i, j - 1]) == maxDif)
        {
            res[db] = i + 1;
            db++;
            break;
        }
    }

Console.Write(db);

for (int i = 0; i < db; i++)
    Console.Write(" " + res[i]);
}
}
}

```

Tesztelés

Érvényes tesztesetek

1. *teszteset: teszt1.txt*

Bemenet – egy helység, egy nap, egy adat	
1 1	
1	
Kimenet	
0	

2. *teszteset: teszt2.txt*

Bemenet – 2 helység, 1 nap, 2 adat	
2 1	
12	
14	
Kimenet	
0	

3. *teszteset: teszt3.txt*

Bemenet – 2 helység, 2 nap, 4 adat	
2 2	
12 14	
14 16	
Kimenet	
2 1 2	

4. *teszteset: teszt4.txt*

Bemenet – 3 helység, 3 nap, 9 adat
3 3
12 14 43
14 16 43
-29 0 29
Kimenet
2 1 3

5. *teszteset: teszt5.txt*

Bemenet – 3 helység, 5 nap, 15 adat
3 5
10 15 12 10 10
11 11 11 11 20
25 16 16 16 20
Kimenet
2 2 3

Érvénytelen tesztesetek

Billentyűzetes bevitel esetén

6. *teszteset*

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

7. *teszteset*

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

...

8. *teszteset*

Bemenet – Határon kívüli szám
N = 1001
Kimenet
Újrakérdezés: N =

Fejlesztési lehetőségek

1. Többszöri futtatás megszervezése
2. Helységek nevének megadása
3. Grafikus visszajelzés a számolás lépéseiről