

„Progalap” komplex beadandó feladat

*Készítette: Golyha Gergő
Neptun-azonosító: A7MMZ1
E-mail: golyhagergo@gmail.com*

*Kurzuskód: **K81DNC-PROG**
Gyakorlatvezető neve: Horváth Győző*

2024. január 14.

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

Felhasználói dokumentáció

Feladat

Mindenhol meleg napok

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 napokat, amikor mindenhol 0 foknál melegebb várható!

Futási környezet

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

Használat

A program indítása

A program a `mindenholMelegNapok\bin\Debug\net8.0\mindenholMelegNapok.exe` néven található a tömörített állományban.

A program használata billentyűzetről való bevitel esetén

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

| # | Adat | Magyarázat |
|-----|---|--------------------------------------|
| 1. | Települések száma (n) | Egész szám 1 és 1000 között |
| 2. | Napok száma (m) | Egész szám 1 és 1000 között |
| 3. | 1. településen az 1. napra jósolt legmagasabb hőmérséklet | Egész szám -50 és 50 között innentől |
| 4. | 1. településen a 2. napra jósolt legmagasabb hőmérséklet | |
| ... | ... | |
| | 2. településen az 1. napra jósolt legmagasabb hőmérséklet | |
| | 2. településen a 2. napra jósolt legmagasabb hőmérséklet | |
| | ... | |
| | n . településen az m . napra jósolt legmagasabb hőmérséklet | |

A program használata fájlból való bevitel esetén

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

```
mindenholMelegNapok.exe < bel.txt
```

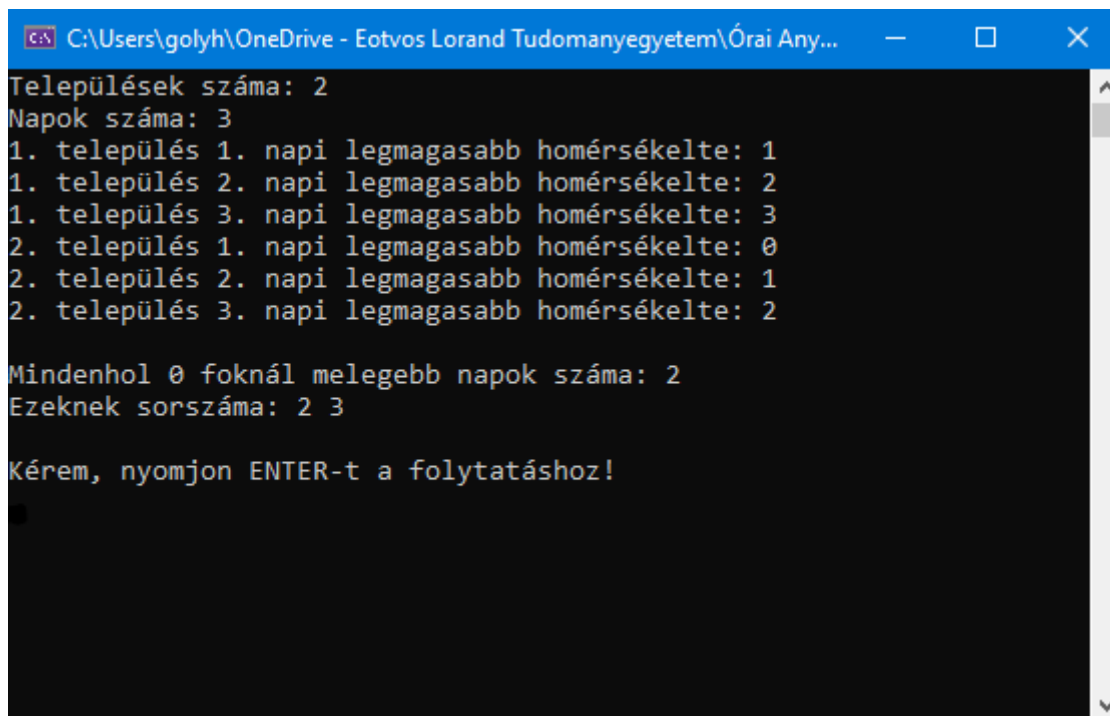
A fájl felépítésének a következő formai követelményei vannak. A fájl első sorában a települések száma (n) és a napok száma (m) van. A következő n sor mindegyikében m darab szám szerepel, közülük az i-edik sorban a j-edik szám az i-edik településen a j-edik napra jósolt legmagasabb hőmérséklet. Például:

```
2 4
10 15 12 -10
-11 11 11 11
```

A program kimenete

A program kiírja azoknak a napoknak a darabszámát és a sorszámaikat, amikor mindenhol 0 foknál melegebbet jósoltak, sorszám szerint növekvő sorrendben.

Minta bemenet és kimenet



```
C:\Users\golyh\OneDrive - Eotvos Lorand Tudomanyegyetem\Órai Any...
Települések száma: 2
Napok száma: 3
1. település 1. napi legmagasabb homérsékelte: 1
1. település 2. napi legmagasabb homérsékelte: 2
1. település 3. napi legmagasabb homérsékelte: 3
2. település 1. napi legmagasabb homérsékelte: 0
2. település 2. napi legmagasabb homérsékelte: 1
2. település 3. napi legmagasabb homérsékelte: 2

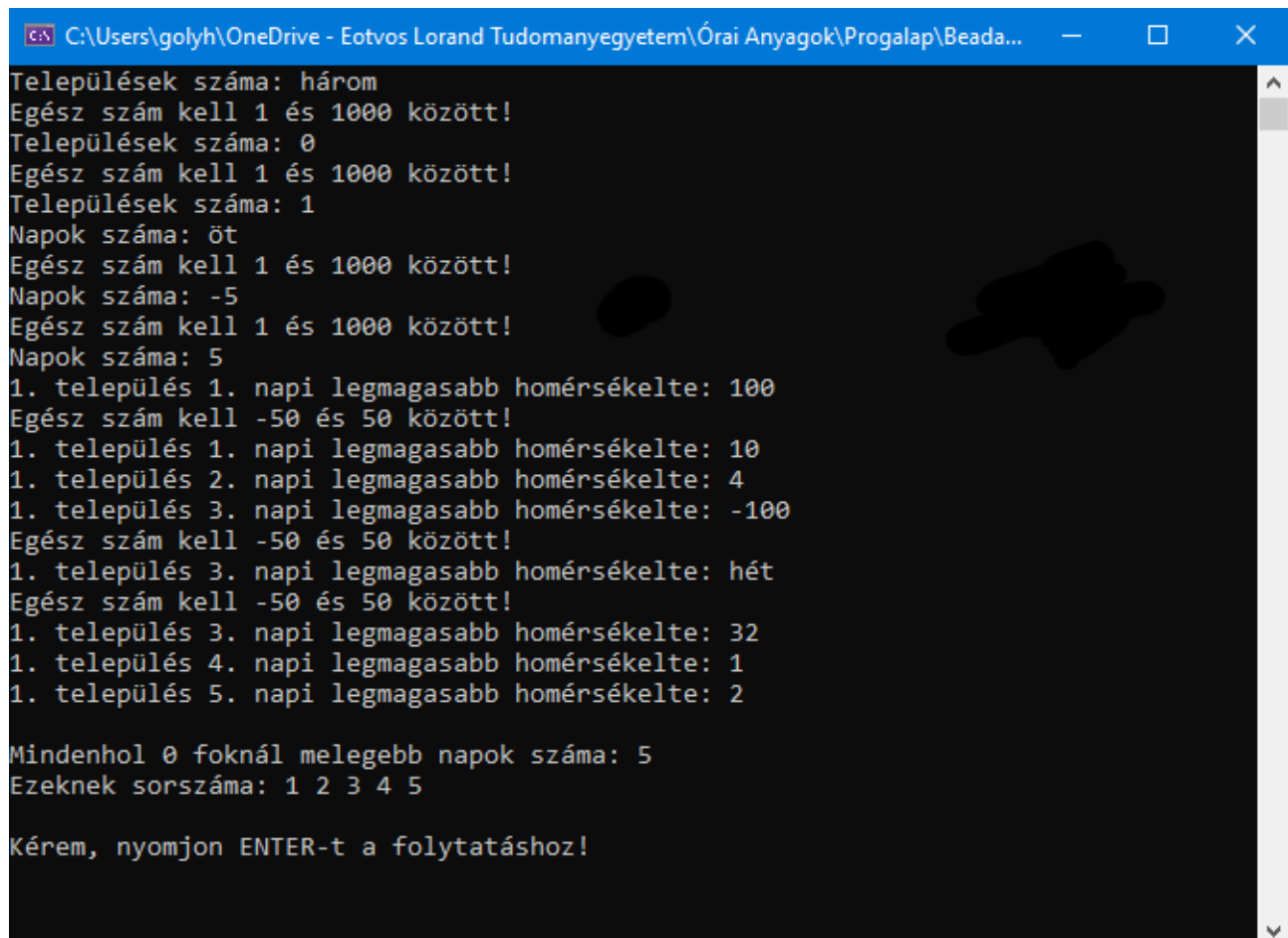
Mindenhol 0 foknál melegebb napok száma: 2
Ezeknek sorszáma: 2 3

Kérem, nyomjon ENTER-t a folytatáshoz!
```

Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha bármelyik megadandó adat nem egész szám vagy az adott intervallumon kívül esik. Hiba esetén a program azzal jelzi a hibát, hogy újra kérdezi azt.

Minta futás hibás bemeneti adatok esetén:



```
C:\Users\golyh\OneDrive - Eotvos Lorand Tudomanyegyetem\Órai Anyagok\Progalap\Bead...
Települések száma: három
Egész szám kell 1 és 1000 között!
Települések száma: 0
Egész szám kell 1 és 1000 között!
Települések száma: 1
Napok száma: öt
Egész szám kell 1 és 1000 között!
Napok száma: -5
Egész szám kell 1 és 1000 között!
Napok száma: 5
1. település 1. napi legmagasabb homérsékelte: 100
Egész szám kell -50 és 50 között!
1. település 1. napi legmagasabb homérsékelte: 10
1. település 2. napi legmagasabb homérsékelte: 4
1. település 3. napi legmagasabb homérsékelte: -100
Egész szám kell -50 és 50 között!
1. település 3. napi legmagasabb homérsékelte: hét
Egész szám kell -50 és 50 között!
1. település 3. napi legmagasabb homérsékelte: 32
1. település 4. napi legmagasabb homérsékelte: 1
1. település 5. napi legmagasabb homérsékelte: 2

Mindenhol 0 foknál melegebb napok száma: 5
Ezeknek sorszáma: 1 2 3 4 5

Kérem, nyomjon ENTER-t a folytatáshoz!
```

Fejlesztői dokumentáció

Feladat

Mindenhol meleg napok

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 napokat, amikor mindenhol 0 foknál melegebb várható!

Tervezés

Specifikáció

Be: $n \in \mathbb{N}$, $m \in \mathbb{N}$, $idojaras \in \mathbb{Z}[1..n, 1..m]$

Ki: $db \in \mathbb{N}$, $y \in [1..db]$

Fv: $folott: N \rightarrow L$, $folott(x) = \text{MIND}(i=1..n, idojaras[i,x] > 0)$

Ef: -

Uf: $(db, y) = \text{KIVÁLOGAT}(i=1..m, folott(i), i)$

Visszavezetés

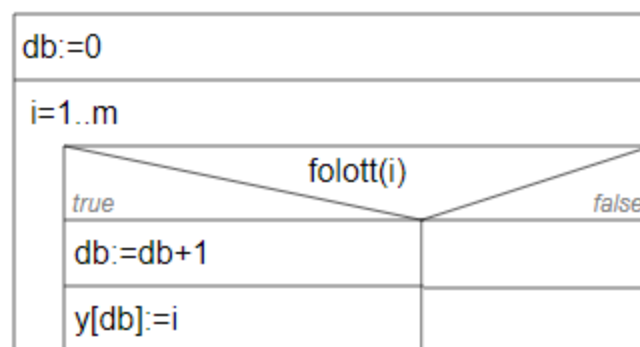
Kiválogatás

| | | |
|--------|--------|-------------|
| $e..u$ | \sim | $1..m$ |
| $T(i)$ | \sim | $folott(i)$ |
| $f(i)$ | \sim | i |

Mind eldöntés (folott)

| | | |
|--------|--------|---------------------|
| $e..u$ | \sim | $1..n$ |
| $T(i)$ | \sim | $idojaras[i,x] > 0$ |

Algoritmus



folott(x: Egész): Logikai
Változó i:Egész, mind: Logikai

| |
|-------------------------|
| i:=1 |
| i<=n és idojaras[i,x]>0 |
| i:=i+1 |
| mind:=i>n |
| folott:=mind |

Fejlesztői környezet

IBM PC, exe futtatására alkalmas operációs rendszer (pl. Windows 11 Home). Visual Studio 2022 (Version 17.8.1) fejlesztői környezet.

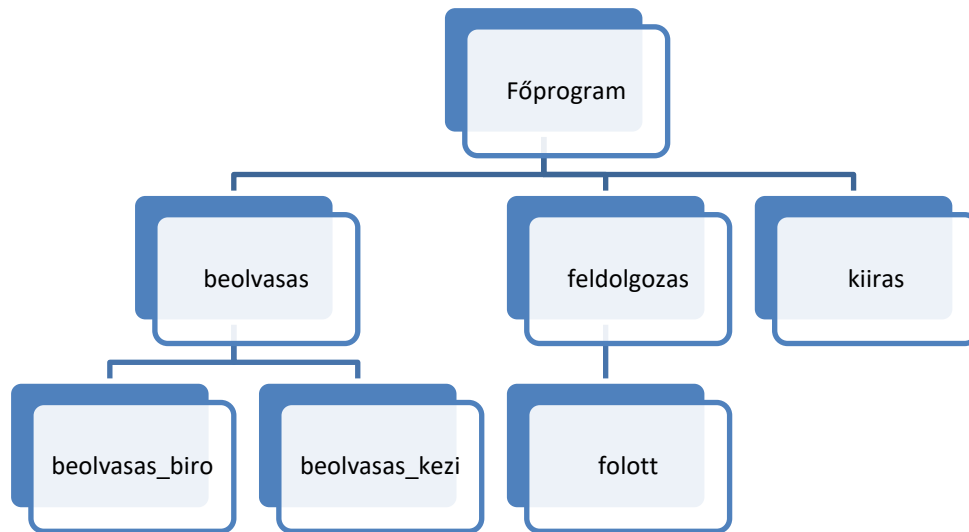
Forráskód

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

| Állomány | Magyarázat |
|--|--|
| mindenholMelegNapok\bin\Debug\net8.0\mindenholMelegNapok.exe | futtatható kód (a futtatáshoz szükséges fájlokkal) |
| mindenholMelegNapok\obj\ | mappa fordításhoz szükséges kódokkal |
| mindenholMelegNapok\Program.cs | C# forráskód |
| mindenholMelegNapok\be1.txt | teszt-bemeneti fájl ₁ |
| mindenholMelegNapok\be2.txt | teszt-bemeneti fájl ₂ |
| mindenholMelegNapok\be3.txt | teszt-bemeneti fájl ₃ |
| mindenholMelegNapok\be4.txt | teszt-bemeneti fájl ₄ |
| mindenholMelegNapok\be5.txt | teszt-bemeneti fájl ₅ |
| mindenholMelegNapok\KBeadandó 2 fázis.pdf | dokumentációk (ez a fájl) |

Megoldás

Függvénystruktúra



A kód

A Program.cs fájl tartalma:

```
/*  
    Készítette: Golyha Gergő  
    Neptun: A7MMZ1  
    E-mail: golhyagergo@gmail.com  
    Feladat: Időjárás előrejelzés: Mindenhol meleg napok  
*/  
  
using System;  
using System.Collections.Generic;  
  
namespace mindenholMelegNapok  
{  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
            //deklarálás bemenet  
            int[,] idojaras;  
  
            //deklarálás kimenet  
            List<int> y;  
  
            idojaras = beolvasas();  
            y = feldolgozas(idojaras);  
            kiiras(y);  
        }  
    }  
}
```



```

static int[,] beolvasas()
{
    int[,] idojaras;
    if (Console.IsInputRedirected) idojaras = beolvasas_biro();
    else idojaras = beolvasas_kezi();
    return idojaras;
}

static int[,] beolvasas_kezi()
{
    int n, m;
    int[,] idojaras;
    bool jo;
    do
    {
        Console.WriteLine("Települések száma: ");
        jo = int.TryParse(Console.ReadLine(), out n) && n >= 1 && n <= 1000;
        if (!jo) Console.WriteLine("Egész szám kell 1 és 1000 között!");
    } while (!jo);
    do
    {
        Console.WriteLine("Napok száma: ");
        jo = int.TryParse(Console.ReadLine(), out m) && m >= 1 && m <= 1000;
        if (!jo) Console.WriteLine("Egész szám kell 1 és 1000 között!");
    } while (!jo);
    idojaras = new int[n, m];
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            do
            {
                Console.WriteLine("{0}. település {1}. napi legmagasabb
hőmérséklete: ", i + 1, j + 1);
                jo = int.TryParse(Console.ReadLine(), out idojaras[i, j]) &&
idojaras[i, j] >= -50 && idojaras[i, j] <= 50;
                if (!jo) Console.WriteLine("Egész szám kell -50 és 50
között!");
            } while (!jo);
        }
    }
    return idojaras;
}

static int[,] beolvasas_biro()
{
    int n, m;
    int[,] idojaras;
    string[] be1 = Console.ReadLine().Split(" ");
    n = int.Parse(be1[0]);
    m = int.Parse(be1[1]);
    idojaras = new int[n, m];
    for (int i = 0; i < n; i++)
    {
        string[] be2 = Console.ReadLine().Split(" ");
        for (int j = 0; j < m; j++) idojaras[i, j] = int.Parse(be2[j]);
    }
    return idojaras;
}

```

```

static List<int> feldolgozas(int[, ] idojaras)
{
    bool folott(int x)
    {
        bool mind;
        int i = 0;
        while (i < idojaras.GetLength(0) && idojaras[i, x] > 0) i++;
        mind = i == idojaras.GetLength(0);
        return mind;
    }

    List<int> y = new List<int>();
    for (int i = 0; i < idojaras.GetLength(1); i++) if (folott(i)) y.Add(i +
1);
    return y;
}

static void kiiras(List<int> y)
{
    if (Console.IsOutputRedirected)
    {
        Console.Write(y.Count);
        foreach (var item in y) Console.Write($" {item}");
    }
    else
    {
        if (y.Count == 0)
        {
            Console.WriteLine("\nNincs olyan nap, amikor mindenhol 0 foknál
melegebb van");
        }
        else
        {
            Console.Write($"{y.Count}\nMindenhól 0 foknál melegebb napok száma:
{y.Count}\nEzeknek sorszáma:");
            foreach (var item in y) Console.Write($" {item}");
        }
        Console.WriteLine("\n\nKérem, nyomjon ENTER-t a folytatáshoz!");
        Console.ReadLine();
    }
}
}
}

```

Tesztelés

Érvényes tesztesetek

1. *teszteset: be1.txt*

| Bemenet – 1 település, 1 nap |
|------------------------------|
| 1 1 1 |
| Kimenet |
| 1 1 |

2. *teszteset: be2.txt*

| Bemenet – 2 település, 3 nap |
|------------------------------|
| 2 3 1 2 3 0 1 0 |
| Kimenet |
| 1 2 |

3. *teszteset: be3.txt*

| Bemenet – 3 település, 5 nap |
|--|
| 3 5 10 15 12 -10 10 -11 11 11 11 20 -12 16 16 16 20 |
| Kimenet |
| 3 2 3 5 |

4. *teszteset: be4.txt*

| Bemenet – 2 település, 10 nap |
|--|
| 2 10 15 -10 -30 12 -7 -36 -39 -7 -8 -22 46 16 14 48 -29 -39 44 -42 9 -17 |
| Kimenet |
| 2 1 4 |

5. *teszteset: be5.txt*

| Bemenet – 3 település, 25 nap |
|---|
| 3 25 -11 22 -18 -48 -50 -3 -21 8 -41 18 -35 -23 -17 -10 48 -2 -37 10 -34 49 14 -8 36 46 -14 -12 1 18 31 11 -46 0 2 -41 -27 8 28 20 -27 -8 -23 27 13 4 -21 6 39 43 44 -29 -26 -8 -31 23 -31 -16 6 32 32 39 -2 -16 -5 33 -34 15 8 -44 -48 -26 33 -35 37 -33 -5 |
| Kimenet |
| 3 8 21 23 |

Érvénytelen tesztesetek

Billentyűzetes bevétel esetén

6. *teszteset*

| Bemenet – szöveges adat |
|-------------------------|
| N = tizenegy |
| Kimenet |
| Újrakérdezés: N = |

7. *teszteset*

| Bemenet – túl nagy szám |
|-------------------------|
| N = 10000 |
| Kimenet |
| Újrakérdezés: N = |

8. *teszteset*

| Bemenet – negatív szám |
|------------------------|
| N = -1 |
| Kimenet |
| Újrakérdezés: N = |

9. *teszteset*

| Bemenet – nem egész szám |
|--------------------------|
| N = 1.30 |
| Kimenet |
| Újrakérdezés: N = |

Fejlesztési lehetőségek

1. Grafikus visszajelzés a számolás lépéseiről
2. Többszöri futtatás megszervezése
3. Települések nevének megadása
4. Tetszőleges minimumhőmérséklet megadása
5. Mindenhol hideg változat