

„Programozás” beadandó feladat

*Készítette: Vida Levente
Neptun-azonosító: DLNINP*

*Kurzuskód: IT-18PROGEG
Gyakorlatvezető neve: Lipták Attila*

2024. január 21.

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	4
Fejlesztői dokumentáció	5
Feladat.....	5
Tervezés	5
Specifikáció.....	5
Visszavezetés	5
Algoritmus.....	6
Fejlesztői környezet	6
Forráskód	6
Megoldás	7
Függvénystruktúra	Error! Bookmark not defined.
A kód	7
Tesztelés.....	9
Érvényes tesztesetek	9
Érvénytelen tesztesetek	10
Fejlesztési lehetőségek.....	Error! Bookmark not defined.

Felhasználói dokumentáció

Feladat

Település valamikor minimális hőmérséklettel

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, amelyeken előfordul valamelyik napi előrejelzések minimuma!

Futási környezet

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

Használat

A program indítása

A program az `telepules\bin\Debug\telepules.exe` néven található a tömörített állományban.

A program használata billentyűzetről való bevétel esetén

Az `telepules.exe` 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. településen az 1. nap hőmérséklete	Egész szám innentől
4.	1. településen az 2. nap hőmérséklete	
...	...	
	1. településen az m . nap hőmérséklete	
	2. településen az 1. nap hőmérséklete	
	...	
	n . településen az m . nap hőmérséklete	

A program használata fájlból való bevétel esetén

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

```
telepules.exe < bel.txt
```

A standard bemenet első sorában a települések száma ($1 \leq N \leq 1000$) és a napok száma ($1 \leq M \leq 1000$) van. Az ezt követő N sorban az egyes napokra jósolt M hőmérséklet értéke található ($-50 \leq H_{i,j} \leq 50$). Például:

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

A program kimenete

A standard kimenet első sorába azon települések T száma van, amelyeken előfordul valamelyik napi előrejelzések minimuma. Ezt követi ezen települések sorszáma, növekvő sorrendben. Például:

```
2 1 2
```

Minta bemenet és kimenet

```
C:\Users\User\Desktop\ELTE-I x + v - □ x
Települések száma = 3
Napok száma = 3
1. település 1. napjának hőmérséklete = 10
1. település 2. napjának hőmérséklete = 15
1. település 3. napjának hőmérséklete = 13
2. település 1. napjának hőmérséklete = 17
2. település 2. napjának hőmérséklete = 11
2. település 3. napjának hőmérséklete = 10
3. település 1. napjának hőmérséklete = 20
3. település 2. napjának hőmérséklete = 22
3. település 3. napjának hőmérséklete = 10
3 darab feltételnek megfelelő település van, sorszáma(ik):
1, 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 természetes szám. Hiba esetén a program azzal jelzi a hibát, hogy újra kérdezi azt.

Mintafutás hibás bemeneti adatok esetén:

```
C:\Users\User\Desktop\ELTE-I x + v - □ x
Települések száma = egy
Helytelen input! (1 <= n <= 1000)
Települések száma = -1
Helytelen input! (1 <= n <= 1000)
Települések száma = 1
Napok száma = egy
Helytelen input! (1 <= m <= 1000)
Napok száma = -1
Helytelen input! (1 <= m <= 1000)
Napok száma = 1
1. település 1. napjának hőmérséklete = egy
Helytelen input! (-50 <= hő <= 50)
1. település 1. napjának hőmérséklete = -1
Helytelen input! (-50 <= hő <= 50)
1. település 1. napjának hőmérséklete = 0
1 darab feltételnek megfelelő település van, sorszáma(ik):
1
Kérem, nyomjon ENTER-t a folytatáshoz!
```

Fejlesztői dokumentáció

Feladat

Település valamikor minimális hőmérséklettel

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, amelyeken előfordul valamelyik napi előrejelzések minimuma!

Tervezés

Specifikáció

Specifikáció

$B_e: n \in \mathbb{N}, m \in \mathbb{N}, h_0 \in \mathbb{R}[n, m]$

$k_i: db \in \mathbb{N}, város \in \mathbb{R}[db]$

$\{f: 1 \leq n \leq 1000, 1 \leq m \leq 1000, \forall i \in [n]: (\forall j \in [m]: (-50 \leq h_0[i, j] \leq 50))$

$f_v: oszlopMin: \mathbb{N} \rightarrow \mathbb{Z}, oszlopMin(j) = \min(i: 1..n, h_0[i, j])$

$U_f: (db, város) = KIVÁLOGAT(i: 1..n, VAN(j: 1..m, h_0[i, j] = oszlopMin(j))), i$

Visszavezetés

Visszavezetés

1, Tétel: Kiválogatás

$B_e: e \in \mathbb{Z}, u \in \mathbb{Z}$

$k_i: db \in \mathbb{N}, y \in H[db]$

$\{f: -$

$U_f: (db, y) = KIVÁLOGAT(i: e..u, T(i), f(i))$

$e..u \sim 1..n$

$db, y \sim db, város$

$T(i) \sim VAN(j: 1..m, h_0[i, j])$

$= oszlopMin(j)$

$f(i) \sim i$

2, Tétel: Elő döntés

$B_e: e \in \mathbb{Z}, u \in \mathbb{Z}$

$k_i: van \in \mathbb{L}$

$\{f: -$

$U_f: van = VAN(i: e..u, T(i))$

$e..u \sim 1..m$

$van \sim van$

$T(i) \sim h_0[i, j] = oszlopMin(j)$

3, Tétel: Minimumkiválogatás (függvény)

$B_e: e \in \mathbb{Z}, u \in \mathbb{Z}$

$k_i: minind \in \mathbb{Z}, minért \in H$

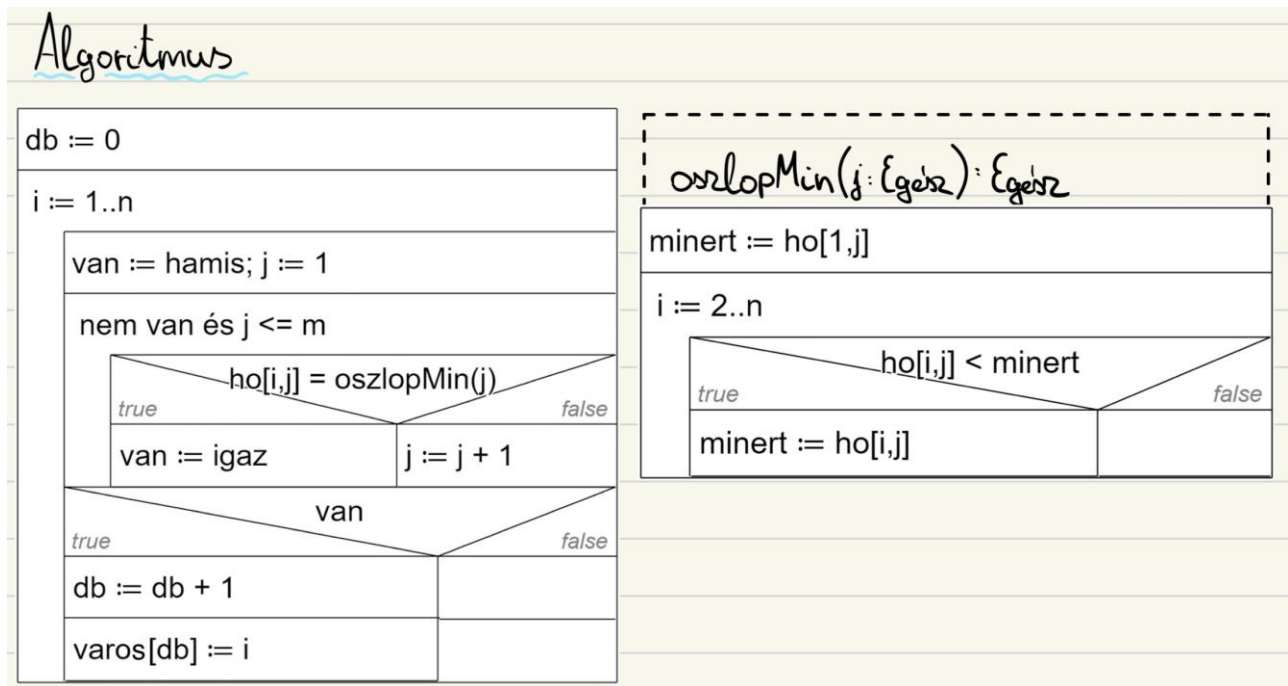
$\{f: e \leq u$

$U_f: (minind, minért) = \min(i: e..u, f(i))$

$e..u \sim 1..n$

$f(i) \sim h_0[i, j]$

Algoritmus



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– a telepules nevű könyvtárban található meg. A fejlesztés során használt könyvtár-struktúra:

Állomány	Magyarázat
telepules\bin\Debug\net8.0\telepules.exe	futtatható kód (a futtatáshoz szükséges fájlokkal)
telepules\obj\	mappa fordításhoz szükséges kódokkal
telepules\Program.cs	C# forráskód
telepules\teszt1.txt	teszt-bemeneti fájl ₁
telepules\teszt2.txt	teszt-bemeneti fájl ₂
telepules\teszt3.txt	teszt-bemeneti fájl ₃
telepules\teszt4.txt	teszt-bemeneti fájl ₄
telepules\teszt5.txt	teszt-bemeneti fájl ₅
telepules\dlinp.docx	dokumentációk (ez a fájl)

Megoldás

A kód

A Program.cs fájl tartalma:

```
using System;
using System.Collections.Generic;

namespace telepules
{
    internal class Program
    {
        static void Main(string[] args)
        {
            //Deklarálás
            int n, m;
            int[,] ho;
            int db = 0;
            List<int> varos = new List<int>();

            //Beolvasás
            (n,m,ho) = beolvas();

            // Kiválogatás
            for (int i = 0; i < n; i++)
            {
                // Eldöntés
                bool van = false;
                int j = 0;
                while (!van && j < m)
                {
                    if (ho[i, j] == oszlopMin(ho, j, n))
                        van = true;
                    else
                        j++;
                }

                if (van)
                {
                    db++;
                    varos.Add(i+1);
                }
            }

            // Kírás
            kiir(db, varos);
        }

        public static int oszlopMin(int[,] ho, int j, int n)
        {
            int minert = ho[0, j];
            for (int i = 1; i < n; i++)
                if (ho[i, j] < minert)
                    minert = ho[i, j];
            return minert;
        }

        public static (int n, int m, int[,] ho) beolvas()
        {
            if (Console.IsInputRedirected)
                return beolvas_biro();
        }
    }
}
```

```

        else
            return beolvas_kezi();
    }

static (int n, int m, int[,] ho) beolvas_biro()
{
    string[] row = Console.ReadLine().Split(' ');
    int n = int.Parse(row[0]);
    int m = int.Parse(row[1]);

    int[,] ho = new int[n, m];
    for (int i = 0; i < n; i++)
    {
        row = Console.ReadLine().Split(" ");
        for (int j = 0; j < m; j++)
            ho[i, j] = int.Parse(row[j]);
    }
    return (n, m, ho);
}

static (int n, int m, int[,] ho) beolvas_kezi()
{
    int n, m;
    bool jo;
    do
    {
        Console.ResetColor();
        Console.Write("Települések száma = ");
        jo = int.TryParse(Console.ReadLine(), out n) && n >= 1 && n <= 1000;
        if (!jo)
        {
            Console.ForegroundColor = ConsoleColor.Red;
            Console.WriteLine("Helytelen input! (1 <= n <= 1000)");
        }
    } while (!jo);
    do
    {
        Console.ResetColor();
        Console.Write("Napok száma = ");
        jo = int.TryParse(Console.ReadLine(), out m) && m >= 1 && m <= 1000;
        if (!jo)
        {
            Console.ForegroundColor = ConsoleColor.Red;
            Console.WriteLine("Helytelen input! (1 <= m <= 1000)");
        }
    } while (!jo);

    int[,] ho = new int[n, m];
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            do
            {
                Console.ResetColor();
                Console.Write("{0}. település {1}. napjának hőmérséklete = ", i + 1, j + 1);
                jo = int.TryParse(Console.ReadLine(), out ho[i, j]) && ho[i, j] >= 0;
                if (!jo)
                {
                    Console.ForegroundColor = ConsoleColor.Red;
                    Console.WriteLine("Helytelen input! (-50 <= hő <= 50)");
                }
            } while (!jo);
        }
    }
}

```



```

    }
    return (n, m, ho);
}

static void kiir(int db, List<int> varos)
{
    if (Console.IsOutputRedirected)
    {
        Console.Write(db + " ");
        for (int i = 0; i < db; i++)
            Console.Write(varos[i] + " ");
    }
    else
    {
        Console.ForegroundColor = ConsoleColor.Green;
        if (db == 0)
            Console.WriteLine("Nincs a feltételnek megfelelő település!");
        else
        {
            Console.WriteLine("{0} darab feltételnek megfelelő település van, sorszáma(ik):", db);
            for (int i = 0; i < db - 1; i++)
                Console.Write("{0}, ", varos[i]);
            Console.WriteLine(varos[db - 1]);
        }
        Console.ForegroundColor = ConsoleColor.Black;
        Console.BackgroundColor = ConsoleColor.Gray;
        Console.WriteLine("Kérem, nyomjon ENTER-t a folytatáshoz!");
        Console.ResetColor();
        Console.ReadLine();
    }
}
}
}

```

Tesztelés

Érvényes tesztesetek

1. *teszteset: teszt1.txt*

Bemenet – 1 település, 1 nap, 50 fok
1 1 50
Kimenet
1 1

2. *teszteset: teszt2.txt*

Bemenet – 2 település, 1 nap, 20 fok, 20 fok
2 1 20 20
Kimenet
2 1 2

3. *teszteset: teszt3.txt*

Bemenet – saját kitalált bemenet
4 3

15 16 17 18 11 14 16 19 12 10 13 20
Kimenet
3 2 3 4

4. *teszteset: teszt4.txt*

Bemenet – saját kitalált bemenet
2 5 10 15 12 13 10 11 10 13 12 14
Kimenet
2 1 2

5. *teszteset: teszt5.txt*

Bemenet – saját kitalált bemenet
4 3 30 31 32 29 30 30 31 33 27 26 25 28
Kimenet
2 3 4

Érvénytelen tesztesetek

Billentyűzetes bevitel esetén

6. *teszteset*

Bemenet – szöveges adat
Települések száma = egy
Kimenet
Helytelen input! ($1 \leq n \leq 1000$) Települések száma =

7. *teszteset*

Bemenet – Túl kicsi hőmérséklet
Települések száma = -51
Kimenet
Helytelen input! ($1 \leq n \leq 1000$) Települések száma =