

„Programozás” komplex beadandó feladat

*Készítette: Pócze Máté
Neptun-azonosító: EBYPPB
E-mail: ebyppb@inf.elte.hu*

*Kurzuskód: IT-18PROGEG
Gyakorlatvezető neve: Éles Júlia*

2025. január 11.

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
Specifikáció.....	5
Visszavezetés	5
Algoritmus	5
Fejlesztői környezet	5
Forráskód	6
Megoldás.....	6
Függvénystruktúra	6
A kód.....	7
A kód (magas szintű függvényekkel).....	10
Tesztelés	13
Automatikus tesztek (Bíró)	13
.....	13
Automatikus tesztek (Bíró, magas szintű függvényekkel).....	14
Érvényes tesztesetek	15
Érvénytelen tesztesetek.....	16
Fejlesztési lehetőségek.....	16

Felhasználói dokumentáció

Feladat

Minden településen melegebb van

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

A program megadja azokat a napokat, amikor minden településen melegebb lesz, mint az előző nap!

Futási környezet

Exe futtatására alkalmas PC, 64-bites operációs rendszer (pl. Windows 11). A program más módon nem futtatható (pl. mobilról, vagy böngészőben). A program használatához billentyűzet szükséges.

Használat

A program indítása

A program az `EBYPPB\beadando_alacsony\bin\Debug\net9.0\beadando_alacsony.exe` néven található a tömörített állományban.

A program használata billentyűzetről való bevétel esetén

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

#	Adat	Magyarázat
1.	Települések száma (n)	Nemnegatív szám (1 és 1000 között)
2.	Napok száma (m)	Nemnegatív szám (1 és 1000 között)
3.	1. településen 1. nap hőfoka	Egész szám inntől (-50 és 50 között)
4.	1. településen 2. nap hőfoka	
...	...	
	1. településen m . nap hőfoka	
	2. településen 1. nap hőfoka	
	n . településen m . nap hőfoka	

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:

```
beadando_alacsony.exe < bel.txt
```

A fájl felépítésének a következő formai követelményei vannak. A fájl első sorában a települések száma (n) és a hőfokok 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 napon jósolt hőmérséklet. Például:

```
3 4
10 9 12 14
-20 -15 0 10
20 25 24 30
```

A program kimenete

A program kiírja azt a számot, ahányszor minden településen melegebb lesz, mint az előző nap, és ezt követően felsorolja ezeknek a napoknak a sorszámát növekvő sorrendben.

Minta bemenet és kimenet

```
Települések száma: 3
Napok száma: 4
1. település 1. napi hőmérséklete: 10
1. település 2. napi hőmérséklete: 9
1. település 3. napi hőmérséklete: 12
1. település 4. napi hőmérséklete: 14
2. település 1. napi hőmérséklete: -20
2. település 2. napi hőmérséklete: -15
2. település 3. napi hőmérséklete: 0
2. település 4. napi hőmérséklete: 10
3. település 1. napi hőmérséklete: 20
3. település 2. napi hőmérséklete: 25
3. település 3. napi hőmérséklete: 24
3. település 4. napi hőmérséklete: 30
1 napon volt melegebb, ezen a napon: 4
Kérem, nyomjon ENTER-t a folytatáshoz!
```

Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni.

Hiba, ha a települések száma vagy napok száma nem 1 és 1000 közé eső egész szám, illetve, ha a hőmérsékletek nem -50 és 50 közé eső egész számok.

Hiba esetén a program azzal jelzi a hibát, hogy újra kérdezi azt.

Minta futás hibás bemeneti adatok esetén:

```
Települések száma: egy
Helytelen formátum! (1-1000)
Települések száma: -1
Helytelen formátum! (1-1000)
Települések száma: 1
Napok száma: egy
Helytelen formátum! (1-1000)
Napok száma: -1
Helytelen formátum! (1-1000)
Napok száma: 1
1. település 1. napi hőmérséklete: egy
Helytelen formátum! (-50-50)!
1. település 1. napi hőmérséklete: -1
Nem volt egyszer sem, hogy mindenhol melegebb volt, mint az elozo nap!
Kérem, nyomjon ENTER-t a folytatáshoz!
```

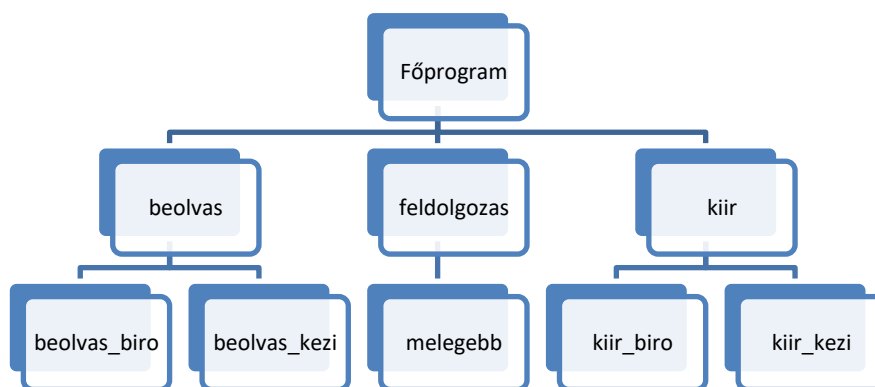

Forráskód

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

Állomány	Magyarázat
EBYPPB\beadando_alacsony\bin\Release\net9.0\beadando_alacsony.exe	futtatható kód (a futtatáshoz szükséges fájlokkal) – alacsony megvalósítás
EBYPPB\beadando_alacsony\obj\	mappa fordításhoz szükséges kódokkal
EBYPPB\beadando_alacsony\EBYPPB_alacsony.cs	C# forráskód – alacsony megvalósítás
EBYPPB\beadando_magas\bin\Release\net9.0\beadando_magas.exe	futtatható kód (a futtatáshoz szükséges fájlokkal) – magas megvalósítás
EBYPPB\beadando_magas\obj\	mappa fordításhoz szükséges kódokkal
EBYPPB\beadando_magas\EBYPPB_magas.cs	C# forráskód – magas megvalósítás
EBYPPB\tesztek\teszt1.txt	teszt-bemeneti fájl ₁
EBYPPB\tesztek\ki1.txt	teszt-kimeneti fájl ₁
...	
EBYPPB\tesztek\teszt8.txt	teszt-bemeneti fájl ₈
EBYPPB\tesztek\ki8.txt	teszt-kimeneti fájl ₈
EBYPPB\doksi\EBYPPB.docx	dokumentációk (ez a fájl)

Megoldás

Függvénystruktúra



A kód

A EBYPPB_alacsony.cs fájl tartalma:

```
// #define BIRO
using System;
using System.Collections.Generic;

namespace beadando_komplex
{
    internal class EBYPPB_alacsony
    {
        static int n, m;
        static int[,] h;

        static void beolvas()
        {
            string bemenet;
            string[] darabok;

        #if BIRO
            bemenet = Console.ReadLine();
            darabok = bemenet.Split(' ', StringSplitOptions.RemoveEmptyEntries);

            n = int.Parse(darabok[0]);
            m = int.Parse(darabok[1]);

            h = new int[n + 1, m + 1];

            for (int i = 1; i <= n; i++)
            {
                bemenet = Console.ReadLine();
                darabok = bemenet.Split(' ', StringSplitOptions.RemoveEmptyEntries);

                for (int j = 1; j <= m; j++)
                {
                    h[i, j] = int.Parse(darabok[j - 1]);
                }
            }

        #else

            bool hiba;

            do
            {
                Console.ResetColor();
                Console.Write("Települések száma: ");

                bemenet = Console.ReadLine();

                hiba = !int.TryParse(bemenet, out n) || n < 1 || n > 1000;

                if (hiba)
                {
                    Console.ForegroundColor = ConsoleColor.DarkRed;
                    Console.WriteLine("Helytelen formátum! (1-1000)");
                }
            } while(hiba);

            do
            {
                Console.ResetColor();
                Console.Write("Napok száma: ");

                bemenet = Console.ReadLine();
```

```

        hiba = !int.TryParse(bemenet, out m) || m < 1 || m > 1000;

        if (hiba)
        {
            Console.ForegroundColor = ConsoleColor.DarkRed;
            Console.WriteLine("Helytelen formátum! (1-1000)");
        }
    } while(hiba);

    h = new int[n + 1, m + 1];

    for (int i = 1; i <= n; i++)
    {
        for (int j = 1; j <= m; j++)
        {
            do
            {
                Console.ResetColor();
                Console.Write($"{i}. település {j}. napi hőmérséklete: ");

                bemenet = Console.ReadLine();

                hiba = !int.TryParse(bemenet, out h[i, j]) || h[i, j] < -50 || h[i,
j] > 50;

                if (hiba)
                {
                    Console.ForegroundColor = ConsoleColor.DarkRed;
                    Console.WriteLine("Helytelen formátum! (-50-50)");
                }
            } while (hiba);
        }
    }
#endif
}

static bool melegebb(int j)
{
    int i = 1;
    while (i <= n && h[i, j] > h[i, j - 1])
    {
        i++;
    }

    return i > n;
}

static (int Darab, List<int> Napok) feldolgozas()
{
    int k = 0;
    List<int> napok = new List<int>();

    for (int i = 2; i <= m; i++)
    {
        if (melegebb(i))
        {
            k++;
            napok.Add(i);
        }
    }

    return (k, napok);
}

static void kiir((int Darab, List<int> Napok) t)

```



```

    {
    #if BIRO
        Console.WriteLine(t.Darab + " " + string.Join(" ", t.Napok));
    #else
        if (t.Darab > 1)
        {
            Console.ForegroundColor = ConsoleColor.DarkGreen;
            Console.WriteLine($"{t.Darab} napon volt melegebb, ezeken a napokon:
{string.Join(", ", t.Napok)}");
        }
        else if (t.Darab == 1)
        {
            Console.ForegroundColor = ConsoleColor.DarkGreen;
            Console.WriteLine($"{t.Darab} napon volt melegebb, ezen a napon:
{string.Join(", ", t.Napok)}");
        }
        else
        {
            Console.ForegroundColor = ConsoleColor.DarkGreen;
            Console.WriteLine("Nem volt egyszer sem, hogy mindenhol melegebb volt, mint
az előző nap!");
        }
        Console.BackgroundColor = ConsoleColor.White;
        Console.ForegroundColor = ConsoleColor.Black;
        Console.Write("Kérem, nyomjon ENTER-t a folytatáshoz!");
        Console.ResetColor();
        Console.ReadLine(); // hogy ne zárja be rögtön a program magát
    #endif
    }

    static void Main(string[] args)
    {
        beolvas();
        kiir(feldolgozas());
    }
}

```

A kód (magas szintű függvényekkel)

A EBYPPB_magas.cs fájl tartalma:

```
//#define BIRO
using System;
using System.Collections.Generic;
using System.Linq;
using System.Numerics;
using System.Text;
using System.Threading.Tasks;
using magas_szintű_mintamegvalósítások;

namespace beadando_k_h
{
    internal class EBYPPB_magas
    {
        static int n, m;
        static int[,] h;

        static void beolvas()
        {
            string bemenet;
            string[] darabok;

#if BIRO
            bemenet = Console.ReadLine();
            darabok = bemenet.Split(' ', StringSplitOptions.RemoveEmptyEntries);

            n = int.Parse(darabok[0]);
            m = int.Parse(darabok[1]);

            h = new int[n + 1, m + 1];

            for (int i = 1; i <= n; i++)
            {
                bemenet = Console.ReadLine();
                darabok = bemenet.Split(' ', StringSplitOptions.RemoveEmptyEntries);

                for (int j = 1; j <= m; j++)
                {
                    h[i, j] = int.Parse(darabok[j - 1]);
                }
            }
#else
            bool hiba;

            do
            {
                Console.ResetColor();
                Console.Write("Települések száma: ");

                bemenet = Console.ReadLine();

                hiba = !int.TryParse(bemenet, out n) || n < 1 || n > 1000;
            }
            while (hiba);
#endif
        }
    }
}
```

```

        if (hiba)
        {
            Console.ForegroundColor = ConsoleColor.DarkRed;
            Console.WriteLine("Helytelen formátum! (1-1000)");
        }
    } while(hiba);

do
{
    Console.ResetColor();
    Console.Write("Napok száma: ");

    bemenet = Console.ReadLine();

    hiba = !int.TryParse(bemenet, out m) || m < 1 || m > 1000;

    if (hiba)
    {
        Console.ForegroundColor = ConsoleColor.DarkRed;
        Console.WriteLine("Helytelen formátum! (1-1000)");
    }
} while(hiba);

h = new int[n + 1, m + 1];

for (int i = 1; i <= n; i++)
{
    for (int j = 1; j <= m; j++)
    {
        do
        {
            Console.ResetColor();
            Console.Write($"{i}. település {j}. napi hőmérséklete: ");

            bemenet = Console.ReadLine();

            hiba = !int.TryParse(bemenet, out h[i, j]) || h[i, j] < -50 ||
h[i, j] > 50;

            if (hiba)
            {
                Console.ForegroundColor = ConsoleColor.DarkRed;
                Console.WriteLine("Helytelen formátum! (-50-50)");
            }
        } while (hiba);
    }
}

#endif
}

static int[] feldolgozas()
{
    int[] napok = Mintak.Kivalogat(2, m, j => Mintak.Mind(1, n, i => h[i, j] >
h[i, j - 1]), j => j);

```

```

        return napok;
    }

    static void kiir(int[] napok)
    {
#if BIRO
        Console.WriteLine(napok.Length + " " + string.Join(" ", napok));
#else
        if (napok.Length > 1)
        {
            Console.ForegroundColor = ConsoleColor.DarkGreen;
            Console.WriteLine($"{napok.Length} napon volt melegebb, ezeken a
napokon: {string.Join(", ", napok)}");
        }
        else if (napok.Length == 1)
        {
            Console.ForegroundColor = ConsoleColor.DarkGreen;
            Console.WriteLine($"{napok.Length} napon volt melegebb, ezen a napon:
{string.Join(", ", napok)}");
        }
        else
        {
            Console.ForegroundColor = ConsoleColor.DarkGreen;
            Console.WriteLine("Nem volt egyszer sem, hogy mindenhol melegebb volt,
mint az előző nap!");
        }
        Console.BackgroundColor = ConsoleColor.White;
        Console.ForegroundColor = ConsoleColor.Black;
        Console.Write("Kérem, nyomjon ENTER-t a folytatáshoz!");
        Console.ResetColor();
        Console.ReadLine(); // hogy ne zárja be rögtön a program magát
#endif
    }

    static void Main(string[] args)
    {
        beolvas();
        kiir(feldolgozas());
    }
}

```

Tesztelés

Automatikus tesztek (Bíró)

Eredmények listázása

Feladat: 1. Minden településen melegebb van ▾ Proba: 7. próbálkozás ▾

Összpont: 100/

Teszt#	Pont	Verdikt...	futási idő
1.1	4/4	Helyes	0.030 sec
2.1	4/4	Helyes	0.035 sec
3.1	4/4	Helyes	0.034 sec
4.1	4/4	Helyes	0.033 sec
5.1	4/4	Helyes	0.035 sec
6.1	5/5	Helyes	0.033 sec
7.1	5/5	Helyes	0.033 sec
8.1	5/5	Helyes	0.033 sec
9.1	5/5	Helyes	0.033 sec
10.1	5/5	Helyes	0.033 sec
11.1	5/5	Helyes	0.034 sec
12.1	5/5	Helyes	0.034 sec
13.1	5/5	Helyes	0.034 sec
14.1	5/5	Helyes	0.034 sec
15.1	5/5	Helyes	0.038 sec
16.1	6/6	Helyes	0.056 sec
17.1	6/6	Helyes	0.066 sec
18.1	6/6	Helyes	0.060 sec
19.1	6/6	Helyes	0.117 sec
20.1	6/6	Helyes	0.308 sec

Beadva: 2025-01-04 17:19:26.0

Automatikus tesztek (Bíró, magas szintű függvényekkel)

Eredmények listázása

Feladat: 1. Minden településen melegebb van ▾ Proba: 8. próbálkozás ▾

Összpont: 100/

Teszt#	Pont	...Verdikt...	futási idő
1.1	4/4	Helyes	0.031 sec
2.1	4/4	Helyes	0.036 sec
3.1	4/4	Helyes	0.035 sec
4.1	4/4	Helyes	0.034 sec
5.1	4/4	Helyes	0.034 sec
6.1	5/5	Helyes	0.035 sec
7.1	5/5	Helyes	0.035 sec
8.1	5/5	Helyes	0.035 sec
9.1	5/5	Helyes	0.035 sec
10.1	5/5	Helyes	0.035 sec
11.1	5/5	Helyes	0.035 sec
12.1	5/5	Helyes	0.036 sec
13.1	5/5	Helyes	0.036 sec
14.1	5/5	Helyes	0.036 sec
15.1	5/5	Helyes	0.038 sec
16.1	6/6	Helyes	0.059 sec
17.1	6/6	Helyes	0.074 sec
18.1	6/6	Helyes	0.062 sec
19.1	6/6	Helyes	0.119 sec
20.1	6/6	Helyes	0.303 sec

Beadva: 2025-01-04 17:19:48.0

Érvényes tesztesetek

1. *teszteset: teszt1.txt*

Bemenet – 3 település, 3 nap – 1 megoldás
3 3 10 11 12 9 10 3 10 15 30
Kimenet
1 2

2. *teszteset: teszt2.txt*

Bemenet – 4 település, 4 nap – több megoldás
4 4 -10 -8 -5 -11 10 16 17 20 -20 -19 -14 -10 30 34 36 40
Kimenet
2 2 3

3. *teszteset: teszt3.txt*

Bemenet – 3 település, 4 nap – 0 megoldás
3 4 10 9 8 7 20 17 14 11 30 31 32 36
Kimenet
0

4. *teszteset: teszt4.txt*

Bemenet – 4 település, 5 nap – 1 megoldás
4 5 10 15 14 16 17 20 21 22 18 17 -10 -5 -2 -7 -10 30 35 32 31 29
Kimenet
1 2

5. *teszteset: teszt5.txt*

Bemenet – 4 település, 3 nap – több megoldás
4 3 10 11 13 20 21 23 30 31 33 40 41 43
Kimenet
2 2 3

Érvénytelen tesztesetek

Billentyűzetes bevétel esetén

6. *teszteset: teszt6.txt*

Bemenet – szöveges adat
Települések száma: három
Kimenet
Helytelen formátum! (1-1000) Települések száma:

7. *teszteset: teszt7.txt*

Bemenet – előfeltételnek nem megfelelő bemenet
Települések száma: 1001
Kimenet
Helytelen formátum! (1-1000) Települések száma:

8. *teszteset: teszt8.txt*

Bemenet – előfeltételnek nem megfelelő bemenet
1. település 1. napi hőmérséklete: -55
Kimenet
Helytelen formátum! (-50-50) 1. település 1. napi hőmérséklete:

Fejlesztési lehetőségek

1. A már beírt adatok módosítása utólag
2. Települések neveinek megadása