„Programozási alapismeretek”  
beadandó feladat

Készítette: *Seregi Péter***[[1]](#footnote-1)**Neptun-azonosító: *G9SIY0*E-mail: [seregip02@gmail.com](mailto:seregip02@gmail.com)

Kurzuskód: IP-18PROGEGGyakorlatvezető neve: *Wolosz Jakab*

2023. január 10.

Tartalom

[Felhasználói dokumentációó 3](#_Toc124269483)

[Feladat 3](#_Toc124269484)

[Futási környezet 3](#_Toc124269485)

[Használat 3](#_Toc124269486)

[A program indítása 3](#_Toc124269487)

[A program használata billentyűzetről való bevitel esetén 3](#_Toc124269488)

[A program használata fájlból való bevitel esetén 3](#_Toc124269489)

[A program kimenete 4](#_Toc124269490)

[Minta bemenet és kimenet 4](#_Toc124269491)

[Hibalehetőségek 4](#_Toc124269492)

[Fejlesztői dokumentáció 5](#_Toc124269493)

[Feladat 5](#_Toc124269494)

[Tervezés 5](#_Toc124269495)

[Specifikáció 5](#_Toc124269496)

[Algoritmus 5](#_Toc124269497)

[Fejlesztői környezet 5](#_Toc124269498)

[Forráskód 6](#_Toc124269499)

[Megoldás 6](#_Toc124269500)

[Programparaméterek 6](#_Toc124269501)

[Programfelépítés 6](#_Toc124269502)

[A kód 6](#_Toc124269503)

[Tesztelés 9](#_Toc124269504)

[Érvényes tesztesetek 9](#_Toc124269505)

[Érvénytelen tesztesetek 9](#_Toc124269506)

[Fejlesztési lehetőségek 10](#_Toc124269507)

2. Felhasználói dokumentációó
   1. 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!

* 1. Futási környezet

IBM PC, exe futtatására alkalmas, 64-bites operációs rendszer (pl. Windows 7). Nem igényel egeret.

* 1. Használat
     1. A program indítása

A program az komplex.exe néven található a tömörített állományban.

* + 1. A program használata billentyűzetről való bevitel esetén

Az komplex.exe fájl elindításával a program az adatokat a **billentyűzet**ről olvassa be a következő sorrendben:

|  |  |  |
| --- | --- | --- |
| # | Adat | Magyarázat |
| **1.** | *Települések száma (N)* | 1 és 1000 közötti egész szám |
| **2.** | *Napok száma (M)* | 1 és 1000 közötti egész szám |
| **3.** | *1. településen az M db előrejelzett hőmérséklet* | -50 és 50 közötti egész számok innentől, szóközzel elválasztva |
| **4.** | *2. településen az M db előrejelzett hőmérséklet* |  |
| **5.** | *…* |  |
| **6.** | *N. településen az M db előrejelzett hőmérséklet* |  |

* + 1. 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:

komplex.exe < 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 sorszámú napra előrejelzett hőmérséklet. Például:

4 3

1 0 4

2 0 1

0 2 0

1 0 0

* + 1. A program kimenete

A program kiírja azoknak a napoknak a darabszámát, ahány napon minden településen 0 foknál nagyobb volt. Ezután fel is sorolja ezen napok sorszámát növekvő sorrendben.

* + 1. Minta bemenet és kimenet

A képen szöveg látható

Automatikusan generált leírás

* + 1. Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha bármelyik megadandó adat nem természetes szám, vagy ha egy sorban nem a megfelelő számú adatot adja meg a felhasználó.

* + - 1. Mintafutás hibás bemeneti adatok esetén:

A képen szöveg látható

Automatikusan generált leírás

1. Fejlesztői dokumentáció
   1. 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!

* 1. Tervezés
  2. Specifikáció

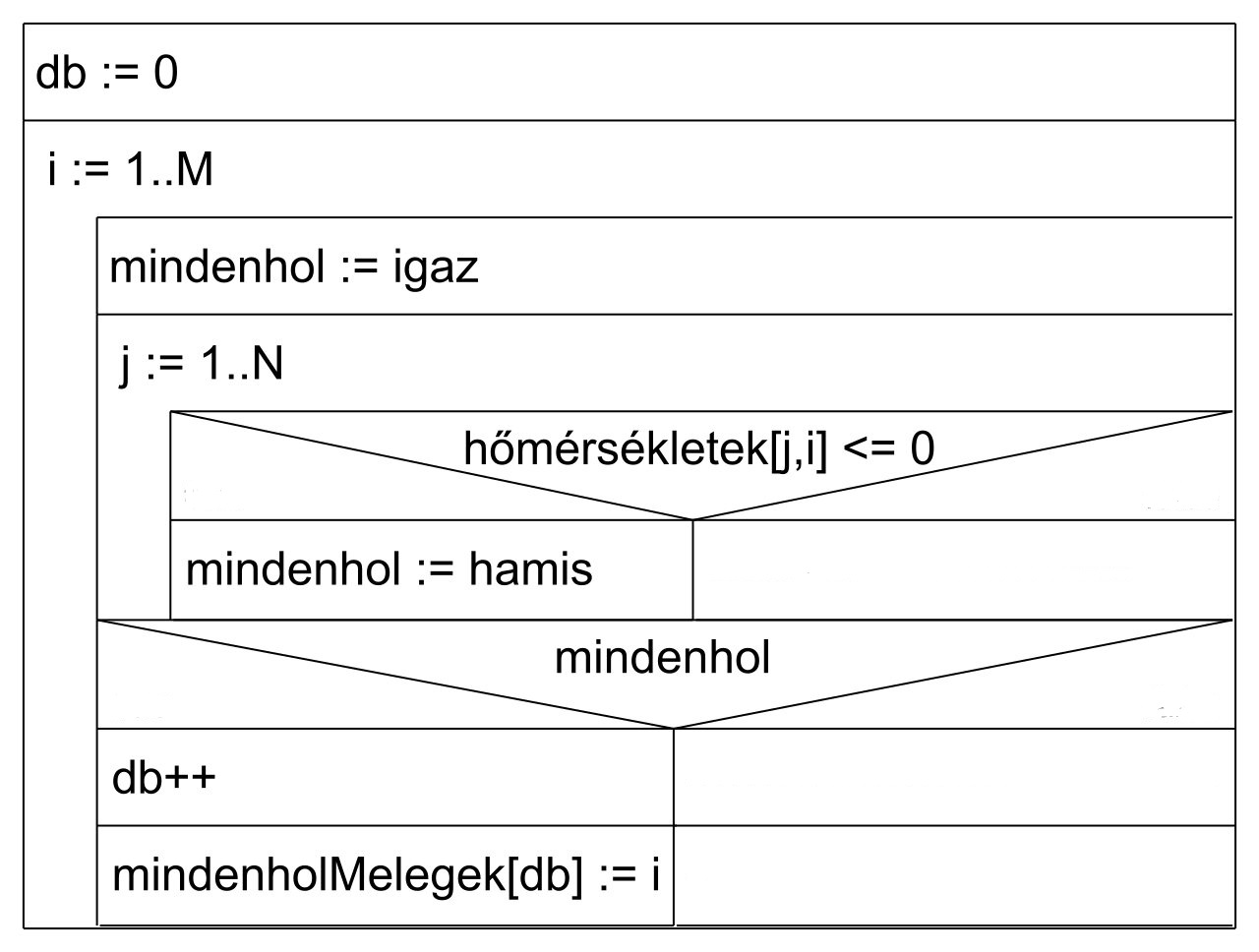
Be:

Ki:

Ef:

Uf: ( 1<= j<=N: hőmérsékletekj,i > 0)

* 1. Algoritmus



* 1. Fejlesztői környezet

IBM PC, exe futtatására alkalmas operációs rendszer (pl. Windows 11). Visual Studio 2022 (Version 17.2.3) fejlesztői környezet.

* 1. Forráskód

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

|  |  |
| --- | --- |
| Állomány | Magyarázat |
| komplex\bin\Debug\net6.0\komplex.exe | futtatható kód (a futtatáshoz szükséges fájlokkal) |
| komplex\obj\ | mappa fordításhoz szükséges kódokkal |
|  |  |
| komplex\Program.cs | C# forráskód |
|  |  |
| komplex\teszt1.txt | teszt-bemeneti fájl1 |
| komplex\teszt2.txt | teszt-bemeneti fájl2 |
| komplex\teszt3.txt | teszt-bemeneti fájl3 |
| komplex\teszt4.txt | teszt-bemeneti fájl4 |
| komplex\teszt5.txt | teszt-bemeneti fájl5 |
|  |  |
| komplex\doksi\dokumentacio.docx | dokumentációk (ez a fájl) |

* 1. Megoldás
     1. Programparaméterek
        1. Konstans

1. N : **Egész** (a települések száma)
2. M : **Egész** (a napok száma)

Változó

homersekletek : **Tömb(**1..N**,**1..M**:Egész)**  
db : **Egész**  
mindenholMelegek : **Tömb(**1..n: **Egész)**

* + 1. Programfelépítés

A program által használt modulok (és helyük):

Program.cs – program, a forráskönyvtárban   
komplex.csproj – program ’projekt fájl’, a forráskönyvtárban

* + 1. A kód

A Program.cs fájl tartalma:

/\*

  Készítette: Seregi Péter

  Neptun: G9SIY0

  E-mail: seregip02@gmail.com

  Feladat: Mindenhol meleg napok

\*/

//#define BIRO

using System;

namespace komplex{

    class Program{

        static void Main(string[] args){

#if BIRO

            string[] elsosor = Console.ReadLine().Split();

            int N = int.Parse(elsosor[0]);

            int M = int.Parse(elsosor[1]);

#else

            bool hiba1 = false;

            bool hiba2 = false;

            int N = 0;

            int M = 0;

            do{

                hiba1 = false;

                Console.WriteLine("Hány település és hány nap van? (1 és 1000 közötti egész számok)");

                string[] elsosor = Console.ReadLine().Split();

                if(elsosor.Length < 2){

                    hiba1 = true;

                    Console.WriteLine("Karaktert, szöveget, valós számot, nem kettő számot vagy nem megfelelő egész számot adtál meg.");

                }

                    if(!hiba1){

                    hiba1 = !int.TryParse(elsosor[0], out N) || N < 1 || N > 1000 || elsosor.Length != 2;

                    hiba2 = !int.TryParse(elsosor[1], out M) || M < 1 || M > 1000;

                    if(hiba1 || hiba2){

                        Console.WriteLine("Karaktert, szöveget, valós számot, nem kettő számot vagy nem megfelelő egész számot adtál meg.");

                    }

                }

            } while(hiba1 || hiba2);

#endif

            int[,] homersekletek = new int[N, M];

            string[] sor = new string[M];

#if BIRO

            for(int i = 0; i < N; i++){

                sor = Console.ReadLine().Split();

                for(int j = 0; j < M; j++){

                    homersekletek[i,j] = int.Parse(sor[j]);

                }

            }

#else

            bool hiba = false;

            for(int i = 0; i < N; i++){

                do{

                    Console.WriteLine("Adja meg a(z) " + (i+1) + ". település hőmérsékleteit");

                    sor = Console.ReadLine().Split();

                    for(int j = 0; j < M; j++){

                        hiba = !int.TryParse(sor[j], out homersekletek[i,j]) || homersekletek[i,j] < -50 || homersekletek[i,j] > 50 || sor.Length != M;

                        if(hiba){

                            Console.WriteLine("Karaktert, szöveget, valós számot, nem M db számot vagy nem megfelelő egész számot adtál meg.");

                            break;

                        }

                    }

                } while(hiba);

            }

#endif

            int[] mindenholMelegek = new int[M];

            int db = 0;

            for(int i = 0; i < M; i++){

                bool mindenhol = true;

                for(int j = 0; j < N; j++){

                    if(homersekletek[j,i] <= 0){

                        mindenhol = false;

                        break;

                    }

                }

                if(mindenhol){

                    mindenholMelegek[db] = i;

                    db++;

                }

            }

#if BIRO

            Console.Write(db + " ");

            for(int i = 0; i < db; i++){

                Console.Write((mindenholMelegek[i] + 1) + " ");

            }

#else

            if(db == 0){

                Console.WriteLine(db + " ilyen nap van.");

                Console.WriteLine("A bezáráshoz nyomjon egy billentyűt!");

                Console.ReadKey();

            } else{

                Console.Write(db + " ilyen nap van, ezek sorrendben a következők: ");

                for(int i = 0; i < db; i++){

                    Console.Write((mindenholMelegek[i] + 1) + " ");

                }

                Console.WriteLine("\nA bezáráshoz nyomjon egy billentyűt!");

                Console.ReadKey();

            }

#endif

        }

    }

}

* 1. Tesztelés
     1. Érvényes tesztesetek
        1. teszteset: teszt1.txt

|  |
| --- |
| Bemenet – *nincs település, nincsenek napok* |
| 0 0 |
| Kimenet |
| 0 |

* + - 1. teszteset: teszt2.txt

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

* + - 1. teszteset: teszt3.txt

|  |
| --- |
| Bemenet – *2 település, 3 nap* |
| 2 3  10 9 8  8 9 10 |
| Kimenet |
| 3 1 2 3 |

* + - 1. teszteset: teszt4.txt

|  |
| --- |
| Bemenet – *5 település 5 nap* |
| 5 5  1 9 3 -1 5  1 2 3 4 5  3 -1 3 3 1  5 4 3 2 1  2 4 3 1 0 |
| Kimenet |
| 2 1 3 |

* + - 1. teszteset: teszt5.txt

|  |
| --- |
| Bemenet – *7 település 5 nap* |
| 7 5  1 -3 5 8 7  -5 3 2 5 2  -1 8 7 3 5  8 9 -5 2 1  8 4 3 -1 1  2 4 3 1 -1  1 2 3 4 5 |
| Kimenet |
| 0 |

* + 1. Érvénytelen tesztesetek

Billentyűzetes bevitel esetén

* + - 1. teszteset

|  |
| --- |
| Bemenet – *szöveges adat* |
| asd |
| Kimenet |
| Karaktert, szöveget, valós számot, nem kettő számot vagy nem megfelelő egész számot adtál meg.  Hány település és hány nap van? (1 és 1000 közötti egész számok) |

* + - 1. teszteset

|  |
| --- |
| Bemenet – *Negatív szám* |
| -1 -2 |
| Kimenet |
| Karaktert, szöveget, valós számot, nem kettő számot vagy nem megfelelő egész számot adtál meg.  Hány település és hány nap van? (1 és 1000 közötti egész számok) |

* 1. Fejlesztési lehetőségek

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

1. [↑](#footnote-ref-1)