„Programozási alapismeretek”  
beadandó feladat

Készítette: *Trefiman Viktor Ádám*Neptun-azonosító: *DVZCBT*E-mail: dvzcbt@inf.elte.hu

Kurzuskód: IP-18PROGEGGyakorlatvezető neve: *Csepregi-Horváth Zsófia Ágnes*

2023. január 15.

Tartalom

[Felhasználói dokumentáció 3](#_Toc124690213)

[Feladat 3](#_Toc124690214)

[Futási környezet 3](#_Toc124690215)

[Használat 3](#_Toc124690216)

[A program indítása 3](#_Toc124690217)

[A program használata billentyűzetről való bevitel esetén 3](#_Toc124690218)

[A program használata fájlból való bevitel esetén 3](#_Toc124690219)

[A program kimenete 4](#_Toc124690220)

[Minta bemenet és kimenet 4](#_Toc124690221)

[Hibalehetőségek 4](#_Toc124690222)

[Fejlesztői dokumentáció 5](#_Toc124690223)

[Feladat 5](#_Toc124690224)

[Tervezés 5](#_Toc124690225)

[Specifikáció 5](#_Toc124690226)

[Visszavezetés 5](#_Toc124690227)

[Algoritmus 6](#_Toc124690228)

[Fejlesztői környezet 7](#_Toc124690229)

[Forráskód 7](#_Toc124690230)

[Megoldás 7](#_Toc124690231)

[Programparaméterek 7](#_Toc124690232)

[Programfelépítés 7](#_Toc124690233)

[Függvénystruktúra 8](#_Toc124690234)

[A kód 8](#_Toc124690235)

[Tesztelés 11](#_Toc124690236)

[Érvényes tesztesetek 11](#_Toc124690237)

[Érvénytelen tesztesetek 12](#_Toc124690238)

[Fejlesztési lehetőségek 12](#_Toc124690239)

2. Felhasználói dokumentáció
   1. Feladat

**Települések átlag szerinti sorrendje**

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 a településeket átlaghőmérséklet szerint csökkenő sorrendben!

* 1. Futási környezet

IBM PC, exe futtatására alkalmas, 32-bites operációs rendszer (pl. Windows 7). Nem igényel egeret.

* 1. Használat
     1. A program indítása

A program az DVZCBT\bin\Release\DVZCBT.exe néven található.

* + 1. A program használata billentyűzetről való bevitel esetén

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

|  |  |  |
| --- | --- | --- |
| # | Adat | Magyarázat |
| **1.** | *Települések száma (n)* | 1-től 1000-ig terjedő egész szám |
| **2.** | *Napok száma (m)* | 1-től 1000-ig terjedő egész szám |
| **3.** | *1. település 1. nap* | -50-től 50-ig terjedő egész szám innentől |
| **4.** | *1. település 2. nap* |  |
| **…** | *…* |  |
|  | *2. település 1. nap* |  |
|  | *2. település 2. nap* |  |
|  | *…* |  |
|  | *n. település m. nap* |  |

* + 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:

DVZCBT.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 hőmérséklet szerepel, közülük az i-edik sorban a j-edik szám az i-edik település a j-edik sorszámú napon mért hőmérséklete.

Például:

3 5

10 15 12 10 10

11 11 11 11 20

12 16 16 16 20

* + 1. A program kimenete

A program kiírja a települések sorszámát az átlaghőmérsékletek szerint csökkenő sorrendben. Az azonos átlaghőmérsékletű elemek növekvő sorrendben kerülnek kiírásra.

* + 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 egész szám és nincs benne a megadott intervallumban. Hiba esetén a program azzal jelzi a hibát, hogy újra kérdezi azt.

* + - 1. Mintafutás hibás bemeneti adatok esetén:

A képen szöveg látható

Automatikusan generált leírás

Fejlesztői dokumentáció

* 1. Feladat

**Települések átlag szerinti sorrendje**

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 a településeket átlaghőmérséklet szerint csökkenő sorrendben!

* 1. Tervezés
  2. Specifikáció

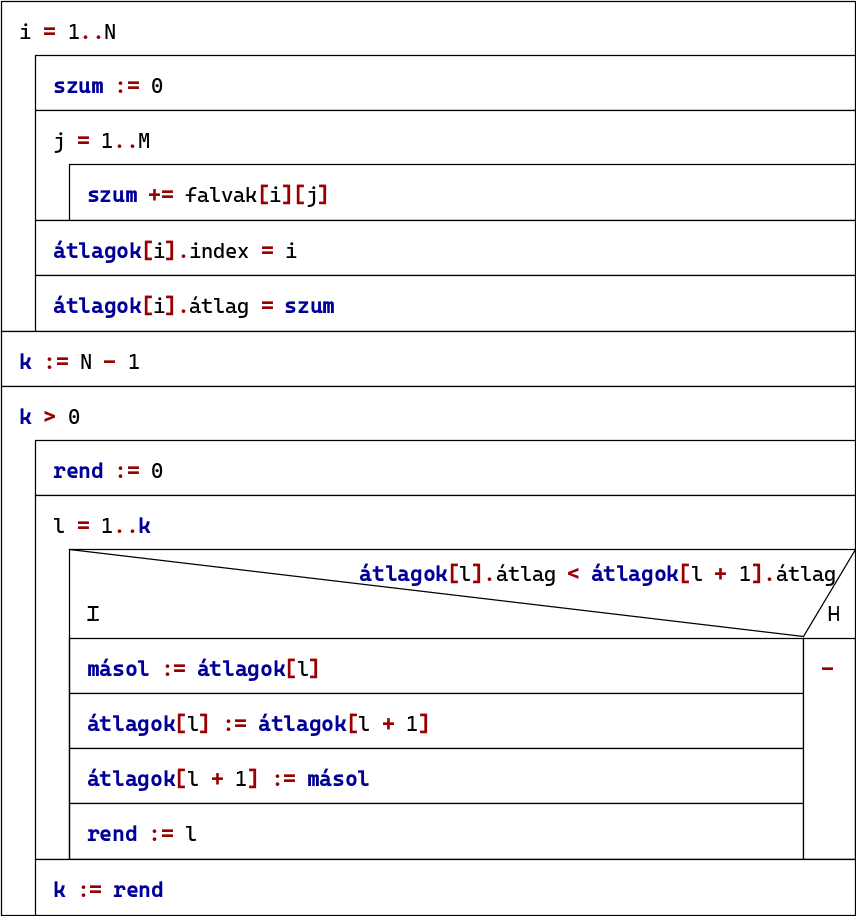
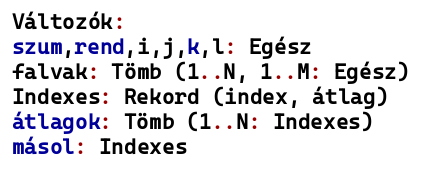
|  |  |
| --- | --- |
| Bemenet: | Nℕ, Mℕ, falvak1..N,1..MℤN×M |
| Kimenet: | átlagok’1..NIndexesN |
| Előfeltétel: | 1≤N≤1000 és 1≤M≤1000 és ∀i[1..N] és  ∀j[1..M]: -50≤falvaki,j≤50  j = 1  M  N  i = 1 |
| Utófeltétel: | i,j)) és RendezettE≤(’.átlag) és  ’∈Permutáció() |
| Definíció: | Indexes = index × átlag, index = ℕ, átlag = ℤ |

* 1. Visszavezetés

*Összegzés*

|  |  |  |
| --- | --- | --- |
| *N* | *~* | *N* |
| *M* | *~* | *M* |
| *X* | *~* | *nap* |
| *S* | *~* | *összeg* |

Algoritmus



* 1. Fejlesztői környezet

IBM PC, exe futtatására alkalmas operációs rendszer (pl. Windows 10 Pro). Visual Studio 2022 (Version 17.3.6) fejlesztői környezet.

* 1. Forráskód

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

|  |  |
| --- | --- |
| Állomány | Magyarázat |
| DVZCBT\bin\Release\DVZCBT.exe | futtatható kód (a futtatáshoz szükséges fájlokkal) |
| DVZCBT\obj\ | mappa fordításhoz szükséges kódokkal |
|  |  |
| DVZCBT\Program.cs | C# forráskód |
|  |  |
| DVZCBT\Documets\be1.txt | teszt-bemeneti fájl1 |
| DVZCBT\Documets\be2.txt | teszt-bemeneti fájl2 |
| DVZCBT\Documets\be3.txt | teszt-bemeneti fájl3 |
| DVZCBT\Documets\be4.txt | teszt-bemeneti fájl4 |
| DVZCBT\Documets\be5.txt | teszt-bemeneti fájl5 |
|  |  |
| DVZCBT\Documets\Dokumentáció.docx | dokumentációk (ez a fájl) |

* 1. Megoldás
     1. Programparaméterek
        1. Típus

1. DataWithIndex = **Rekord**(index,avg:**Egész**)
   * + 1. Változó

villages : **Tömb(**1..n**,**1..m**:Egész)**   
avgs : **Tömb(**1..n: **Egész)**

* + 1. Programfelépítés

A program által használt modulok (és helyük):

Program.cs – program, a forráskönyvtárban   
DVZCBT.sln – program ’megoldás fájl’, a forráskönyvtárban   
DVZCBT.csproj – program ’projekt fájl’, a forráskönyvtárban

* + 1. Függvénystruktúra
    2. A kód

A Program.cs fájl tartalma:

/\*

Készítette: Trefiman Viktor Ádám

Neptun: DVZCBT

E-mail: dvzcbt@inf.elte.hu

Feladat: Települések átlag szerinti sorrendje

\*/

using System;

namespace DVZCBT

{

internal class Program

{

struct DataWithIndex

{

public int index;

public int avg;

}

static void Main()

{

// Mátrix és méretei:

int n = 0, m = 0;

int[,] villages = null;

// Input formájának eldöntése:

if (Console.IsInputRedirected)

Input\_Biro(ref n, ref m, ref villages);

else

Input\_Console(ref n, ref m, ref villages);

// A feladat:

DataWithIndex[] avgs = new DataWithIndex[n];

Calculate(villages, n, m, ref avgs);

// Kiírás:

Output(avgs, n);

}

static void Input\_Biro(ref int n, ref int m, ref int[,] villages)

{

// Méretek bekérése:

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

n = int.Parse(part[0]);

m = int.Parse(part[1]);

// Mátrix feltöltése:

 villages = new int[n, m];

for (int i = 0; i < n; ++i)

{

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

for (int j = 0; j < m; ++j)

villages[i, j] = int.Parse(part[j]);

}

}

static void DataChecker(string ask, ref int data, int min, int max)

{

// Adatot kér be és leellenőrzi hogy megfelel-e az intervallumnak:

do

{

Console.ResetColor();

Console.Write(ask);

if (int.TryParse(Console.ReadLine(), out data) && data >= min && data <= max)

return;

else

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine($"{min}-től {max}-ig terjedő egész szám kell!");

}

} while (true);

}

static void Input\_Console(ref int n, ref int m, ref int[,] villages)

{

// Méretek bekérése:

DataChecker("Települések száma = ", ref n, 1, 1000);

DataChecker("Napok száma = ", ref m, 1, 1000);

// Mátrix feltöltése:

villages = new int[n, m];

for (int i = 0; i < n; ++i)

{

for (int j = 0; j < m; ++j)

DataChecker($"{i + 1}. település {j + 1}. nap = ", ref villages[i, j], -50, 50);

}

}

static void Calculate(int[,] villages, int n, int m, ref DataWithIndex[] avgs)

{

// Hőmérsékletek összegzése településenként:

for (int i = 0; i < n; ++i)

{

int sum = 0;

for (int j = 0; j < m; ++j)

sum += villages[i, j];

avgs[i] = new DataWithIndex { index = i + 1, avg = sum };

}

// Javított buborék-rendezés az összegekre:

DataWithIndex temp;

int k = n - 1, cut;

while (k > 0)

{

cut = 0;

for (int l = 0; l < k; ++l)

{

if (avgs[l].avg < avgs[l + 1].avg)

{

temp = avgs[l];

avgs[l] = avgs[l + 1];

avgs[l + 1] = temp;

cut = l;

}

}

k = cut;

}

}

static void Output(DataWithIndex[] avgs, int n)

{

// Az adatok kiírása csőből, illetve konzolról való beolvasás esetén:

if (Console.IsInputRedirected)

{

for (int i = 0; i < n; ++i)

Console.Write($"{avgs[i].index} ");

 }

else

{

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("Az átlagok szerint rendezett települések sorszámai:");

Console.ResetColor();

for (int i = 0; i < n; ++i)

Console.Write($"{avgs[i].index} ");

Console.ForegroundColor = ConsoleColor.Black;

Console.BackgroundColor = ConsoleColor.Gray;

Console.WriteLine("\nKérem, nyomjon ENTER-t a folytatáshoz!");

Console.ResetColor();

Console.ReadLine();

}

}

}

}

Tesztelés

* + 1. Érvényes tesztesetek
       1. teszteset: be1.txt

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

* + - 1. teszteset: be2.txt

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

* + - 1. teszteset: be3.txt

|  |
| --- |
| Bemenet – *3 település, 5 nap, 15 mérés* |
| 3 5  10 15 12 10 10  11 11 11 11 20  12 16 16 16 20 |
| Kimenet |
| 3 2 1 |

* + - 1. teszteset: be4.txt

|  |
| --- |
| Bemenet – *10 település, 10 nap, 100 mérés* |
| 10 10  -2 3 -4 6 8 5 2 -4 0 7  0 0 0 0 2 3 4 5 1 -1  -2 3 -4 6 8 5 2 -4 0 7  1 -4 3 4 -5 3 -1 7 4 6  0 2 1 3 13 8 4 19 12 22  -2 3 -4 6 8 5 2 -4 0 7  0 0 0 0 2 -3 4 5 1 -1  -2 3 -4 6 8 5 2 -4 0 7  1 4 3 4 5 3 1 7 4 6  0 -2 -1 3 13 8 4 19 12 22 |
| Kimenet |
| 5 10 9 1 3 6 8 4 2 7 |

* + - 1. teszteset: be5.txt

|  |
| --- |
| Bemenet – *1000 település, 1000 nap, 1000000 mérés* |
| Megtalálható: DVZCBT\Documets\be5.txt |
| Kimenet |
| Megtalálható: DVZCBT\Documets\ki5.txt |

Érvénytelen tesztesetek

Billentyűzetes bevitel esetén

* + - 1. teszteset

|  |
| --- |
| Bemenet – *Szöveges adat* |
| Települések száma = tizenegy |
| Kimenet |
| Hibaüzenet és újrakérdezés:  1-től 1000-ig terjedő egész szám kell! Települések száma = |

* + - 1. teszteset

|  |
| --- |
| Bemenet – *Értelmezési tartományon kívül eső szám* |
| Települések száma = 1  Napok száma = -1 |
| Kimenet |
| Hibaüzenet és újrakérdezés:  1-től 1000-ig terjedő egész szám kell!  Napok száma = |

8. teszteset

|  |
| --- |
| Bemenet – *Értelmezési tartományon kívül eső szám* |
| Települések száma = 1  Napok száma = 1  1. település 1. nap = -51 |
| Kimenet |
| Hibaüzenet és újrakérdezés:  -50-től 50-ig terjedő egész szám kell!  1. település 1. nap = |

* 1. Fejlesztési lehetőségek

1. Többszöri futtatás megszervezése
2. Települések nevének megadása
3. A napok pontos dátumának megadása
4. Grafikus visszajelzés a számolás lépéseiről