**„Programozás”  
beadandó feladat**

Készítette: Pályi Kristóf FerencNeptun-azonosító: *Q9NL7W*E-mail: q9nl7w@inf.elte.hu

Kurzuskód: IT-18PROGEGGyakorlatvezető neve: *Baksza Dávid*

2024. január 21.

Tartalom

[Felhasználói dokumentáció 3](#_Toc155108363)

[Feladat 3](#_Toc155108364)

[Futási környezet 3](#_Toc155108365)

[Használat 3](#_Toc155108366)

[A program indítása 3](#_Toc155108367)

[A program használata billentyűzetről való bevitel esetén 3](#_Toc155108368)

[A program használata fájlból való bevitel esetén 3](#_Toc155108369)

[A program kimenete 4](#_Toc155108370)

[Minta bemenet és kimenet 4](#_Toc155108371)

[Hibalehetőségek 4](#_Toc155108372)

[Fejlesztői dokumentáció 5](#_Toc155108373)

[Feladat 5](#_Toc155108374)

[Tervezés 5](#_Toc155108375)

[Specifikáció 5](#_Toc155108376)

[Visszavezetés 5](#_Toc155108377)

[Algoritmus 6](#_Toc155108378)

[Fejlesztői környezet 6](#_Toc155108379)

[Forráskód 7](#_Toc155108380)

[Megoldás 7](#_Toc155108381)

[Függvénystruktúra 7](#_Toc155108382)

[A kód 7](#_Toc155108383)

[Tesztelés 10](#_Toc155108384)

[Érvényes tesztesetek 10](#_Toc155108385)

[Érvénytelen tesztesetek 11](#_Toc155108386)

[Fejlesztési lehetőségek 11](#_Toc155108387)

2. Felhasználói dokumentáció
   1. Feladat

Sokáig meleg települések

* 1. 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, ahol a hőmérséklet legalább egy héten át 30 fok felett lesz!
  2. Futási környezet

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

* 1. Használat
     1. A program indítása

A program az komplex\bin\Debug\net8.0/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.** | *Helységek száma (n)* | Egész szám 1 és 1000 között |
| **2.** | *Napok száma (m)* | Egész szám 1 és 1000 között |
| **3.** | *1. helységben az 1. nap hőmérséklete* | Egész szám -50 és 50 között innentől |
| **4.** | *1. helységben az 2. nap hőmérséklete* |  |
| **…** | *…* |  |
|  | *1. helységben az m. nap hőmérséklete* |  |
|  | *2. helységben az 1. nap hőmérséklete* |  |
|  | *…* |  |
|  | *n. helységben az m. nap hőmérséklete* |  |

* + 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 helységek 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 helységben a j-edik napon mért időjárás. Például:

4 3

1 0 4

2 0 1

0 2 0

1 0 0

* + 1. A program kimenete

A program kiírja azoknak a helységeknek a darabszámát és a sorszámait, ahol zsinórban 7-szer 30 foknál magasabb hőmérsékletet mértek.

* + 1. Minta bemenet és kimenet

A képen szöveg, képernyőkép, Betűtípus, menü látható

Automatikusan generált leírás

* + 1. Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha bármelyik megadandó adat nem a megadott intervallumban van. Hiba esetén a program azzal jelzi a hibát, hogy újra kérdezi azt.

* + - 1. Mintafutás hibás bemeneti adatok esetén:

2. A képen szöveg, képernyőkép, Betűtípus, tervezés látható

   Automatikusan generált leírás Fejlesztői dokumentáció
   1. Feladat

Sokáig meleg települések

* 1. 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, ahol a hőmérséklet legalább egy héten át 30 fok felett lesz!
  2. Tervezés
     1. Specifikáció

Be: n eleme N, m eleme N, Hom eleme N[1..n,1..m]

Ki: db eleme N, sorsz eleme N[1..db]

Fv: HarmincFolottHetszer: N -> L,

HarmincFolottHetszer(i) = (létezik [j eleme [1..m-6] : Hom[i,j] > 30 és Hom[i,j+1] > 30 és Hom[i,j+2] > 30 és Hom[i,j+3] > 30 és Hom[i,j+4] > 30 és Hom[i,j+5] > 30 és Hom[i,j+6] > 30](https://en.wikipedia.org/wiki/%C6%8E)

)

Ef: 1<=n<=1000 és 1<=m<=1000 és minden i eleme [1..n] : minden j eleme [1..m] : (-50<=Hom[i,j] és 50>=Hom[i,j])

* + 1. Uf: (db, sorsz) = KIVÁLOGAT(i=1..n, HarmincFolottHetszer(i), i)
    2. Visszavezetés

1. KIVÁLOGATÁS
2. e..u ~ 1..n
3. db ~ db
4. y ~ sorsz
5. T(i) ~ HarmincFolottHetszer(Hom[i])
6. f(i) ~ i
   * 1. Algoritmus

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

* 1. Fejlesztői környezet

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

* 1. Forráskód

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

|  |  |
| --- | --- |
| Állomány | Magyarázat |
| komplex\bin\Debug\net8.0 | 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\Programozás komplex beadandó fázis2.docx | dokumentációk (ez a fájl) |

* 1. Megoldás
     1. Függvénystruktúra
     2. A kód

A Program.cs fájl tartalma:

1. using System;
2. using System.Collections.Generic;
3. using System.Data.Common;
4. *class* Program
5. {
6. #region Adataim
7. // név: Pályi Kristóf Ferenc
8. // neptun-kód: Q9NL7W
9. // email: q9nl7w@inf.elte.hu
10. #endregion
11. static void Main(string[] args)
12. {
13. #region Beolvasás
14. int[,] adatok = Beolvas();
15. int db;
16. int[] helysegek;
17. #endregion
18. #region Feldolgozás
19. (db, helysegek) = Kivalogat(adatok);
20. #endregion
21. #region Kiírás
22. Kiir(db, helysegek);
23. #endregion
24. }
25. static int[,] Beolvas()
26. {
27. if (Console.IsInputRedirected)
28. {
29. return Beolvas\_biro();
30. }
31. else
32. {
33. return Beolvas\_kezi();
34. }
35. }
36. static int[,] Beolvas\_biro()
37. {
38. string[] elsosor = Console.ReadLine().Split(" ");
39. int faluk = int.Parse(elsosor[0]);
40. int napok = int.Parse(elsosor[1]);
41. int[,] adatok = new int[faluk, napok];
42. for (int i = 0; i < faluk; i++)
43. {
44. string[] sor = Console.ReadLine().Split(" ");
45. for (int j = 0; j < napok; j++)
46. {
47. adatok[i, j] = int.Parse(sor[j]);
48. }
49. }
50. return adatok;
51. }
52. static int[,] Beolvas\_kezi()
53. {
54. int n, m;
55. bool jo;
56. do
57. {
58. Console.ResetColor();
59. Console.Write("Helységek száma = ");
60. jo = int.TryParse(Console.ReadLine(), out n) && n >= 1 && n <= 1000;
61. if (!jo)
62. {
63. Console.ForegroundColor = ConsoleColor.Red;
64. Console.WriteLine("Érvényes érték kell!");
65. }
66. } while (!jo);
67. do
68. {
69. Console.ResetColor();
70. Console.Write("Napok száma = ");
71. jo = int.TryParse(Console.ReadLine(), out m) && m >= 1 && m <= 1000;
72. if (!jo)
73. {
74. Console.ForegroundColor = ConsoleColor.Red;
75. Console.WriteLine("Érvényes érték kell!");
76. }
77. } while (!jo);
78. int[,] adatok = new int[n, m];
79. for (int i = 0; i < n; i++)
80. {
81. for (int j = 0; j < m; j++)
82. {
83. do
84. {
85. Console.ResetColor();
86. Console.Write("{0}. helyseg {1}. nap homerseklete: = ", i + 1, j + 1);
87. jo = int.TryParse(Console.ReadLine(), out adatok[i, j]) && adatok[i, j] >= -50 && adatok[i, j] <= 50;
88. if (!jo)
89. {
90. Console.ForegroundColor = ConsoleColor.Red;
91. Console.WriteLine("Érvényes érték kell!");
92. }
93. } while (!jo);
94. }
95. }
96. return adatok;
97. }
98. static (int db, int[] helysegek) Kivalogat(int[,] adatok)
99. {
100. int db = 0;
101. int[] helysegek = new int[adatok.GetLength(0)];
102. for (int i = 0; i < adatok.GetLength(0); i++)
103. {
104. int counter = 0;
105. for (int j = 0; j < adatok.GetLength(1); j++)
106. {
107. if (adatok[i, j] <= 30)
108. {
109. counter = 0;
110. }
111. else
112. {
113. counter++;
114. }
115. if (counter >= 7)
116. {
117. j = adatok.GetLength(1);
118. }
119. }
120. if (counter >= 7)
121. {
122. db++;
123. helysegek[i] = i + 1;
124. }
125. }
126. return (db, helysegek);
127. }
128. static void Kiir(int db, int[] helysegek)
129. {
130. if (Console.IsInputRedirected)
131. {
132. Kiir\_biro(db, helysegek);
133. }
134. else
135. {
136. Kiir\_kezi(db, helysegek);
137. }
138. }
139. static void Kiir\_biro(int db, int[] helysegek)
140. {
141. Console.Write(db);
142. for (int i = 0; i < helysegek.Length; i++)
143. {
144. if (helysegek[i] != 0)
145. {
146. Console.Write(" " + helysegek[i]);
147. }
148. }
149. }
150. static void Kiir\_kezi(int db, int[] helysegek)
151. {
152. Console.WriteLine("A kiválogatás eredménye:");
153. Console.WriteLine("Helységek száma: {0}", db);
154. Console.Write("Helységek: ");
155. for (int i = 0; i < helysegek.Length; i++)
156. {
157. if (helysegek[i] != 0)
158. {
159. Console.Write("{0} ", helysegek[i]);
160. }
161. }
162. }
163. }
     1. Tesztelés
        1. Érvényes tesztesetek
           1. teszteset: be1.txt

|  |
| --- |
| Bemenet – *egy helység, egy nap* |
| 1 1  1 |
| Kimenet |
| 0 |

* + - 1. teszteset: be2.txt

|  |
| --- |
| Bemenet – *2 helység, 2 nap* |
| 2 2  20 20  21 21 |
| Kimenet |
| 0 |

* + - 1. teszteset: be3.txt

|  |
| --- |
| Bemenet – *1 helység, 7 nap* |
| 1 7  1 2 3 4 5 6 7 |
|  |
| Kimenet |
| 0 |

* + - 1. teszteset: be4.txt

|  |
| --- |
| Bemenet – *3 helység, 8 nap* |
| 3 8  31 33 34 45 46 33 33 32  31 33 34 45 46 33 33 32  31 33 34 45 46 33 33 32 |
|  |
| Kimenet |
| 3 1 2 3 |

* + - 1. teszteset: be5.txt

|  |
| --- |
| Bemenet – *…* |
| 4 7 |
| 31 33 34 45 46 33 33  31 33 34 45 46 33 30  31 33 3 45 46 33 4  31 33 34 45 46 33 31 |
| Kimenet |
| 2 1 4 |

* + 1. Érvénytelen tesztesetek

Billentyűzetes bevitel esetén

* + - 1. teszteset

|  |
| --- |
| Bemenet – *szöveges adat* |
| Helysegek szama = 11tizenegy |
| Kimenet |
| Újrakérdezés: N = |

* + - 1. teszteset

|  |
| --- |
| Bemenet – *Negatív szám* |
| Napok szama = -1 |
| Kimenet |
| Újrakérdezés:  Napok száma= |

…

* 1. Fejlesztési lehetőségek

1. Többszöri futtatás megszervezése
2. Helységek és napok nevének megadása
3. Grafikus visszajelzés a számolás lépéseiről