**„Programozás”  
komplex beadandó feladat**

Készítette: *Soós Csaba*Neptun-azonosító: *AZXX1Z*E-mail: azxx1z@inf.elte.hu

Kurzuskód: IT-18PROGEGGyakorlatvezető neve: *Pluhár Zsuzsa*

2025. január 12.

Tartalom

[Felhasználói dokumentáció 3](#_Toc186797335)

[Feladat 3](#_Toc186797336)

[Futási környezet 3](#_Toc186797337)

[Használat 3](#_Toc186797338)

[A program indítása 3](#_Toc186797339)

[A program használata billentyűzetről való bevitel esetén 3](#_Toc186797340)

[A program használata fájlból való bevitel esetén 3](#_Toc186797341)

[A program kimenete 4](#_Toc186797342)

[Minta bemenet és kimenet 4](#_Toc186797343)

[Hibalehetőségek 4](#_Toc186797344)

[Fejlesztői dokumentáció 5](#_Toc186797345)

[Feladat 5](#_Toc186797346)

[Tervezés 5](#_Toc186797347)

[Specifikáció 5](#_Toc186797348)

[Visszavezetés 5](#_Toc186797349)

[Algoritmus 6](#_Toc186797350)

[Fejlesztői környezet 6](#_Toc186797351)

[Forráskód 6](#_Toc186797352)

[Megoldás 7](#_Toc186797353)

[Függvénystruktúra 7](#_Toc186797354)

[A kód 7](#_Toc186797355)

[A kód (magas szintű függvényekkel) 10](#_Toc186797356)

[Tesztelés 11](#_Toc186797357)

[Automatikus tesztek (Bíró) 11](#_Toc186797358)

[Automatikus tesztek (Bíró, magas szintű függvényekkel) 12](#_Toc186797359)

[Érvényes tesztesetek 13](#_Toc186797360)

[Érvénytelen tesztesetek 13](#_Toc186797361)

[Fejlesztési lehetőségek 14](#_Toc186797362)

2. Felhasználói dokumentáció
   1. Feladat

**Nagy változású települések**

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 egyik napról a következőre legalább 10 fokot változik!

* 1. 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 nagy\_valtozasu\_telepulesek\bin\Debug\net8.0\nagy\_valtozasu\_telepulesek.exe néven található a tömörített állományban.

* + 1. A program használata billentyűzetről való bevitel esetén

Az nagy\_valtozasu\_telepulesek.exe fájl elindításával a program az adatokat a **billentyűzet**ről olvassa be a következő sorrendben:

|  |  |  |
| --- | --- | --- |
| # | Adat | Magyarázat |
| **1.** | *Települések száma (N)* | 1≤N≤1000 |
| **2.** | *Napok száma (M)* | 1≤N≤1000 |
| **3.** | *1. településen az 1. napon jósolt hőmérséklet* | -50≤Hi,j≤50 innentől |
| **4.** | *1. településen az 2. napon jósolt hőmérséklet* |  |
| **…** | *…* |  |
|  | *1. településen az m. napon jósolt hőmérséklet* |  |
|  | *2. településen az 1. napon jósolt hőmérséklet* |  |
|  | *…* |  |
|  | *n. településen az m. napon jósolt 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:

nagy\_valtozasu\_telepulesek.exe < be1.txt

A fájl felépítésének a következő formai követelményei vannak. A standard bemenet első sorában a települések száma (N) és a napok száma (M) van. Az ezt követő N sorban az egyes napokra jósolt M hőmérséklet értéke található. Például:

3 3

10 19 12 10 10

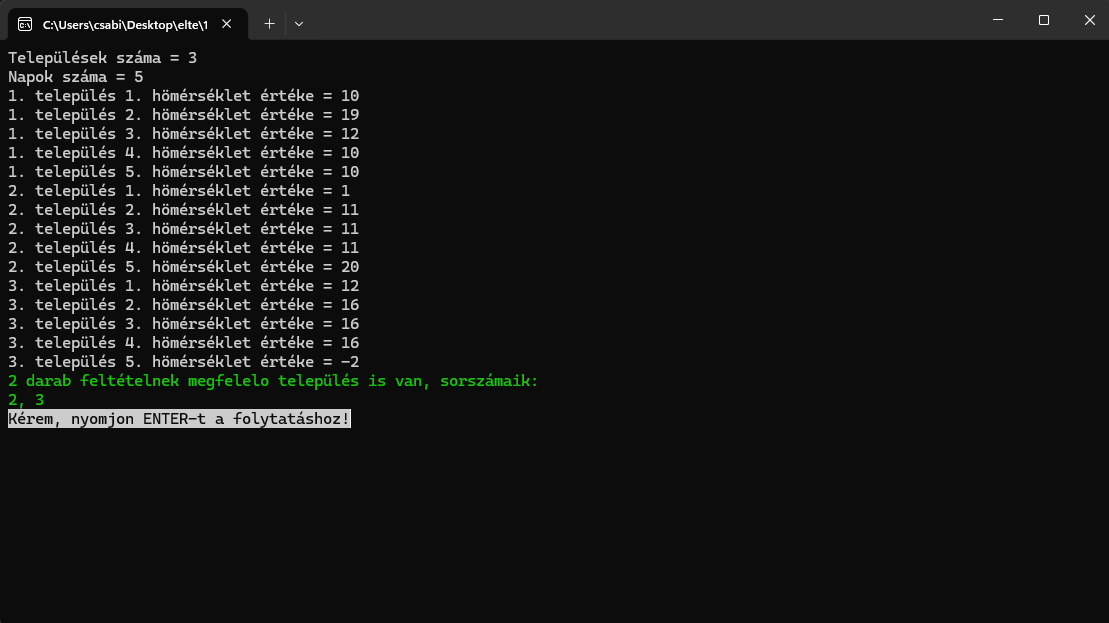
1 11 11 11 20

12 16 16 16 -2

* + 1. A program kimenete

A program kiírja azon települések T számát kell kiírni, ahol a hőmérséklet egyik napról a következőre legalább 10 fokot változik. Ezt követi ezen települések sorszáma.

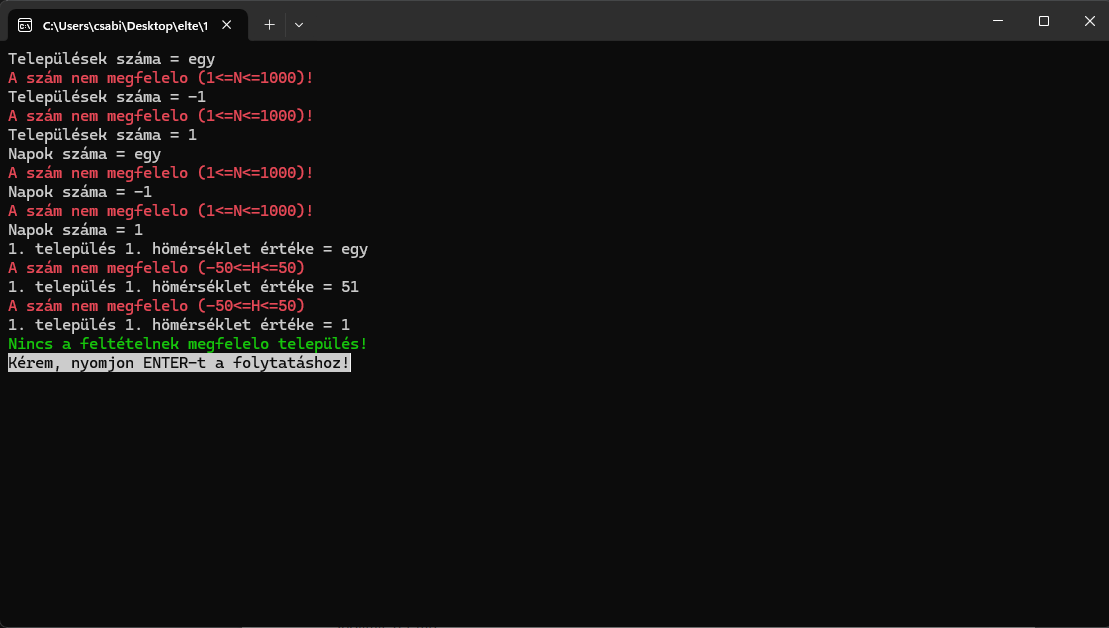
* + 1. Minta bemenet és kimenet



* + 1. 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.

* + - 1. Mintafutás hibás bemeneti adatok esetén:



1. Fejlesztői dokumentáció
   1. Feladat

**Nagy változású települések**

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 egyik napról a következőre legalább 10 fokot változik!

* 1. Tervezés
     1. [Specifikáció](https://progalap.elte.hu/specifikacio/?data=H4sIAAAAAAAAE42RXU7CQBDHrzLpE02mZLftkrihJpgYYzD4ojxY%2BlBo1QK7GFpNlJDoG1fxHNyEkzgtUEqVxHa785H%2FzP5muzDSl3iUPCajMEtm2pDGRSxBb1arHoLamueZiudpPJnGGSUefN5saqRNBQPdTSREw60unc3Tj1DNJnmYq6IhKS4fJfC2p2H9nYJue5wxVviUVIWjKsnN6jOh8uKMQDYoHO9CRaElWNur4PgJjoOirp5se4KZ5kDf0%2BmNaIglmwkedK%2F766%2Bb26vOXSPximmg3%2Bk1xp5Nx2C9l8UDq97%2BnIjhLXx6%2F3VwXUrVudhESEwDjSxOs9SQ%2FsKIwiyk6wYtwRloUBIEmUq1BJ8S9PicIT9DbmPusAD3aeT7ZVeyJGvtlmXTD4D8i4YSbLLlPVB3G8EJiEmHKiYS8uZx%2BjrNDMmXeATobgFbpwAFcoG2QEegy2gvYSwitgQyFDl7hZJh%2BZY5mxU9cl2uFlV2p8ZOI6P7P3ixhXdOwjN0RRW6ICWUwxTEZNFgZI8HOKAXN3BE7P5B7JDkBHOw%2FAGmP1T9iwMAAA%3D%3D)

Be: n∈N, m∈N, homerseklet∈Z[1..n,1..m]

Ki: db∈N, sorszamok∈N[1..db]

Ef: 1<=n és n<=1000 és 1<=m és m<=1000 és ∀i∈[1..n]:(∀j∈[1..m]:(-50<=homerseklet[i,j] és homerseklet[i,j]<=50))

Uf: (db,sorszamok) = KIVÁLOGAT(i=1..n, VAN(j=2..m, homerseklet[i,j-1]-homerseklet[i,j]>=10 vagy homerseklet[i,j]-homerseklet[i,j-1]>=10), i)

* + 1. Visszavezetés

*Kiválogatás*

db   ~ db

y    ~ sorszamok

e..u ~ 1..n

T(i) ~ VAN(j=2..m, homerseklet[i,j-1]-homerseklet[i,j]>=10 vagy homerseklet[i,j]-homerseklet[i,j-1]>=10)

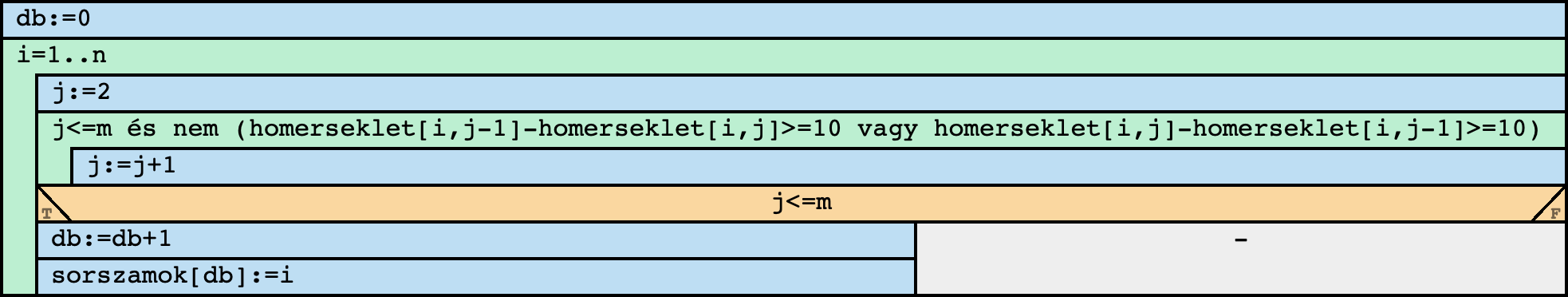
f(i) ~ i

*Eldöntés (van)*

e..u ~ 2..m

T(i) ~ homerseklet[i,j-1]-homerseklet[i,j]>=10 vagy homerseklet[i,j]-homerseklet[i,j-1]>=10

* + 1. [Algoritmus](https://progalap.elte.hu/stuki/?data=H4sIAAAAAAAAE61W3W7aMBh9FeRK1abRKgm0ECYuSru2dKOtmv5AEReO7ZBAsFkSoKXigfYce7HZaWycFrRGm29ijj%2Bfc74fAi8gwKABjH3Drh4aZr1yYB1aNdM2D0AZ%2BGxOojY%2Fp7MwLIOYhAQlBK8RyjCJQeMF3NSvTjr2kyP2KaECymAKI0KT9SU%2FwJhQ0PBgGJMySJ6nhF%2BIyc8ZoYjwC8gPQszvtDHn7oMT5%2BT09qF7y0%2BC42%2FXqGt%2FB4NVeY1LTS1wrak72aKcwIRMeDiPSMhTwiHsNpoG4BrO9b29OGqdKQ0F5DSUr%2BLZjY6ureO2c8NPLozb3mNYqfLt3cJP6JIrikQVuzShyRVJdMGVUweM4iAJGBVUTXN%2Fn3LQZfjZyVy2c5lyB8qkdKC51hxoxflwqUeNpiUkKmdzP74PWkpCATkJrUZFK31%2BdRYOu%2B5jWlPFI%2BU04iIZbarpaDdMvjYnpd%2B%2F4hIlk9Inn01IFJNxSJJ%2BUB7tmYO9N9Bgd8ivmEZpDofPpbeHb6M5QRb%2FeVPnVOV4nippmacCcnlqxS7SudEXU4hM79oOnZoPSkQBOZHLcb3TmqHjrSKUsangU4GST7up8akvSfFZ%2BDG6dOqVq3S4rJ5vOd1lOhZRddk6DTtPSlkBOeUevnjs9sbwr5moQMmn3fw%2FmSiDwr7ikXIacZGpDrzNIy2aH81IK4IU%2BbmZ0zqUUm0IUalzo6r%2B0qjekAITs%2BXdjd1sMGVv1WBK4J9lYhbFSzhh4z52B41mAFZcL%2BbVQQkbRnAimvMCKJyI%2Bx0YiHfsHEYBdEPxkwn4x4ixd2%2Fu2GeL%2B3VYZkOg5wRiEmXQaiD6E3IPIrVXS32w43kGX5xmRzw90UWOIYW5xoHaeemuZlQMmO4QgV69BgavGbYgGg8jNqM4I3Yt1xJh4imJCZIYqSEod166w1VkEEvQiT8J7%2BgQX6kqX5LOsyRGPIy20Wk1fscqVpbzh%2Fg9z7ZfTw2jjmzxn2eHEFLFh3pFVqs%2FGfHrHiYJAAA%3D)



* 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 nagy\_valtozasu\_telepulesek nevű könyvtárban található meg. A fej­lesztés során használt könyvtár-struktúra:

|  |  |
| --- | --- |
| Állomány | Magyarázat |
| nagy\_valtozasu\_telepulesek\bin\Debug\net8.0\nagy\_valtozasu\_telepulesek.exe | futtatható kód (a futtatáshoz szükséges fájlokkal) |
| nagy\_valtozasu\_telepulesek\obj\ | mappa fordításhoz szükséges kódokkal |
|  |  |
| nagy\_valtozasu\_telepulesek\Program.cs | C# forráskód |
|  |  |
| nagy\_valtozasu\_telepulesek\teszt1.txt | teszt-bemeneti fájl1 |
| nagy\_valtozasu\_telepulesek\teszt2.txt | teszt-bemeneti fájl2 |
| nagy\_valtozasu\_telepulesek\teszt3.txt | teszt-bemeneti fájl3 |
| nagy\_valtozasu\_telepulesek\teszt4.txt | teszt-bemeneti fájl4 |
| nagy\_valtozasu\_telepulesek\teszt5.txt | teszt-bemeneti fájl5 |
|  |  |
| nagy\_valtozasu\_telepulesek\doksi\Programozás komplex beadandó fázis2.docx | dokumentációk (ez a fájl) |

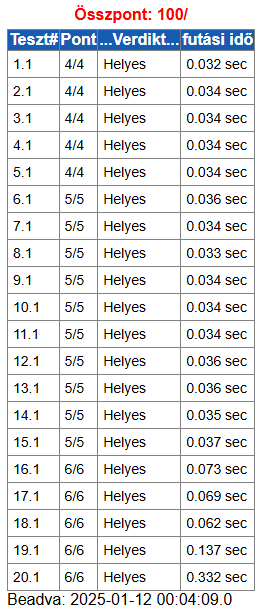
* 1. Megoldás
     1. Függvénystruktúra
     2. A kód

A Program.cs fájl tartalma:

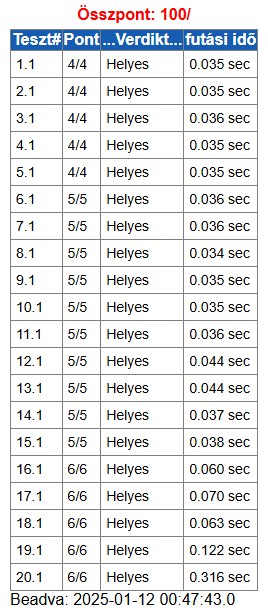
1. using System;
2. using System.Collections.Generic;
3. namespace nagy\_valtozasu\_telepulesek
4. {
5. class Program
6. {
7. static int[,] beolvas()
8. {
9. if (Console.IsInputRedirected)
10. {
11. return beolvas\_biro();
12. }
13. else
14. {
15. return beolvas\_kezi();
16. }
17. }
18. static int[,] beolvas\_biro()
19. {
20. string[] sortomb = Console.ReadLine().Split(' ');
21. int n = int.Parse(sortomb[0]);
22. int m = int.Parse(sortomb[1]);
23. int[,] homerseklet = new int[n, m];
24. for (int i = 0; i < n; i++)
25. {
26. sortomb = Console.ReadLine().Split(' ');
27. for (int j = 0; j < m; j++)
28. {
29. homerseklet[i, j] = int.Parse(sortomb[j]);
30. }
31. }
32. return homerseklet;
33. }
34. static int[,] beolvas\_kezi()
35. {
36. int n, m;
37. bool jo;
38. do
39. {
40. Console.ResetColor();
41. Console.Write("Települések száma = ");
42. jo = int.TryParse(Console.ReadLine(), out n) && 1<=n && n<=1000;
43. if (!jo)
44. {
45. Console.ForegroundColor = ConsoleColor.Red;
46. Console.WriteLine("A szám nem megfelelő (1<=N<=1000)!");
47. }
48. } while (!jo);
49. do
50. {
51. Console.ResetColor();
52. Console.Write("Napok száma = ");
53. jo = int.TryParse(Console.ReadLine(), out m) && 1 <= m && m <= 1000;
54. if (!jo)
55. {
56. Console.ForegroundColor = ConsoleColor.Red;
57. Console.WriteLine("A szám nem megfelelő (1<=N<=1000)!");
58. }
59. } while (!jo);
60. int[,] homerseklet = new int[n,m];
61. for (int i = 0; i < n; i++)
62. {
63. for (int j = 0; j < m; j++)
64. {
65. do
66. {
67. Console.ResetColor();
68. Console.Write($"{i + 1}. település {j + 1}. hömérséklet értéke = ");
69. jo = int.TryParse(Console.ReadLine(), out homerseklet[i, j]) && -50 <= homerseklet[i, j] && homerseklet[i, j] <= 50;
70. if (!jo)
71. {
72. Console.ForegroundColor = ConsoleColor.Red;
73. Console.WriteLine("A szám nem megfelelő (-50<=H<=50)");
74. }
75. } while (!jo);
76. }
77. }
78. return homerseklet;
79. }
80. static void kiir(int db, List<int> sorszamok)
81. {
82. if (Console.IsOutputRedirected)
83. {
84. Console.Write($"{db} ");
85. Console.Write(String.Join(' ', sorszamok));
86. }
87. else
88. {
89. Console.ForegroundColor = ConsoleColor.Green;
90. if (db == 0)
91. {
92. Console.WriteLine("Nincs a feltételnek megfelelő település!");
93. }
94. else
95. {
96. Console.WriteLine("{0} darab feltételnek megfelelő település is van, sorszámaik: ", db);
97. Console.WriteLine(String.Join(", ", sorszamok));
98. }
99. Console.ForegroundColor = ConsoleColor.Black;
100. Console.BackgroundColor = ConsoleColor.Gray;
101. Console.Write("Kérem, nyomjon ENTER-t a folytatáshoz!");
102. Console.ResetColor();
103. Console.ReadLine();
104. }
105. }
106. static (int db, List<int> sorszamok) megold(int[,] homerseklet)
107. {
108. int db = 0;
109. List<int> sorszamok = new List<int>();
110. for (int i = 0; i < homerseklet.GetLength(0); i++)
111. {
112. int j = 1;
113. while (j < homerseklet.GetLength(1) && Math.Abs(homerseklet[i, j - 1] - homerseklet[i,j])<10)
114. {
115. j += 1;
116. }
117. if (j < homerseklet.GetLength(1))
118. {
119. db += 1;
120. sorszamok.Add(i + 1);
121. }
122. }
123. return (db, sorszamok);
124. }
125. static void Main(string[] args)
126. {
127. int[,] homerseklet;
128. int db = 0;
129. List<int> sorszamok = new List<int>();
130. homerseklet = beolvas();
131. (db, sorszamok) = megold(homerseklet);
132. kiir(db, sorszamok);
133. }
134. }
135. }
     * 1. A kód (magas szintű függvényekkel)

A Program.cs fájl tartalma:

1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Numerics;
5. using System.Text;
6. using System.Threading.Tasks;
7. using magas\_szintű\_mintamegvalósítások;
8. namespace nagy\_valtozasu\_telepulesek\_magas
9. {
10. internal class Program
11. {
12. static void Main(string[] args)
13. {
14. int[,] homerseklet;
15. int db;
16. int[] sorszamok;
17. homerseklet = beolvas();
18. int n = homerseklet.GetLength(0);
19. int m = homerseklet.GetLength(1);
20. sorszamok = Mintak.Kivalogat(0, n - 1, i => Mintak.Van(1, m - 1, j => (Math.Abs(homerseklet[i, j - 1] - homerseklet[i, j]) >= 10)), i => i + 1);
21. db = sorszamok.Length;
22. kiir(db, sorszamok);
23. }
24. static void kiir(int db, int[] sorszamok)
25. {
26. if (Console.IsOutputRedirected)
27. {
28. Console.Write($"{db} ");
29. Console.Write(String.Join(' ', sorszamok));
30. }
31. else
32. {
33. Console.ForegroundColor = ConsoleColor.Green;
34. if (db == 0)
35. {
36. Console.WriteLine("Nincs a feltételnek megfelelő település!");
37. }
38. else
39. {
40. Console.WriteLine("{0} darab feltételnek megfelelő település is van, sorszámaik: ", db);
41. Console.WriteLine(String.Join(", ", sorszamok));
42. }
43. Console.ForegroundColor = ConsoleColor.Black;
44. Console.BackgroundColor = ConsoleColor.Gray;
45. Console.Write("Kérem, nyomjon ENTER-t a folytatáshoz!");
46. Console.ResetColor();
47. Console.ReadLine();
48. }
49. }
50. static int[,] beolvas()
51. {
52. if (Console.IsInputRedirected)
53. {
54. return beolvas\_biro();
55. }
56. else
57. {
58. return beolvas\_kezi();
59. }
60. }
61. static int[,] beolvas\_biro()
62. {
63. string[] sortomb = Console.ReadLine().Split(' ');
64. int n = int.Parse(sortomb[0]);
65. int m = int.Parse(sortomb[1]);
66. int[,] homerseklet = new int[n, m];
67. for (int i = 0; i < n; i++)
68. {
69. sortomb = Console.ReadLine().Split(' ');
70. for (int j = 0; j < m; j++)
71. {
72. homerseklet[i, j] = int.Parse(sortomb[j]);
73. }
74. }
75. return homerseklet;
76. }
77. static int[,] beolvas\_kezi()
78. {
79. int n, m;
80. bool jo;
81. do
82. {
83. Console.ResetColor();
84. Console.Write("Települések száma = ");
85. jo = int.TryParse(Console.ReadLine(), out n) && 1 <= n && n <= 1000;
86. if (!jo)
87. {
88. Console.ForegroundColor = ConsoleColor.Red;
89. Console.WriteLine("A szám nem megfelelő (1<=N<=1000)!");
90. }
91. } while (!jo);
92. do
93. {
94. Console.ResetColor();
95. Console.Write("Napok száma = ");
96. jo = int.TryParse(Console.ReadLine(), out m) && 1 <= m && m <= 1000;
97. if (!jo)
98. {
99. Console.ForegroundColor = ConsoleColor.Red;
100. Console.WriteLine("A szám nem megfelelő (1<=N<=1000)!");
101. }
102. } while (!jo);
103. int[,] homerseklet = new int[n, m];
104. for (int i = 0; i < n; i++)
105. {
106. for (int j = 0; j < m; j++)
107. {
108. do
109. {
110. Console.ResetColor();
111. Console.Write($"{i + 1}. település {j + 1}. hömérséklet értéke = ");
112. jo = int.TryParse(Console.ReadLine(), out homerseklet[i, j]) && -50 <= homerseklet[i, j] && homerseklet[i, j] <= 50;
113. if (!jo)
114. {
115. Console.ForegroundColor = ConsoleColor.Red;
116. Console.WriteLine("A szám nem megfelelő (-50<=H<=50)");
117. }
118. } while (!jo);
119. }
120. }
121. return homerseklet;
122. }
123. }
124. }
     1. Tesztelés
        1. Automatikus tesztek (Bíró)



* + 1. Automatikus tesztek (Bíró, magas szintű függvényekkel)



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

|  |
| --- |
| Bemenet |
| 3 4  10 20 30 40  5 15 25 35  -10 0 10 20 |
| Kimenet |
| 3 1 2 3 |

* + - 1. teszteset: teszt2.txt

|  |
| --- |
| Bemenet |
| 2 5  -10 -20 -30 -40 -50  50 40 30 20 10 |
| Kimenet |
| 2 1 2 |

* + - 1. teszteset: be3.txt

|  |
| --- |
| Bemenet |
| 4 6  0 0 0 0 0 0  10 10 10 10 10 10  -50 -40 -30 -20 -10 0  5 15 5 15 5 15 |
| Kimenet |
| 2 3 4 |

* + - 1. teszteset: be4.txt

|  |
| --- |
| Bemenet |
| 1 3  -5 10 -5 |
| Kimenet |
| 1 1 |

* + - 1. teszteset: be5.txt

|  |
| --- |
| Bemenet |
| 5 7  10 10 10 10 10 10 10  -10 -10 -10 -10 -10 -10 -10  50 40 30 20 10 0 -10  15 15 25 15 25 15 25  0 10 20 30 40 50 50 |
| Kimenet |
| 3 3 4 5 |

* + 1. Érvénytelen tesztesetek

Billentyűzetes bevitel esetén

* + - 1. teszteset

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

* + - 1. teszteset

|  |
| --- |
| Bemenet – *Nem megfelelő szám* |
| N = -1 |
| Kimenet |
| Újrakérdezés:  N= |

…

* + - 1. 8. teszteset

…

* 1. Fejlesztési lehetőségek

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

([Github repo](https://github.com/csabisoos/elte/blob/main/1.%20felev/programozas%20gyak/komplex_beadando/komplex_dokumentacio.md" \l "fejleszt%C3%A9si-lehet%C5%91s%C3%A9gek))