„Programozási alapismeretek”  
beadandó feladat

Készítette: *Kovács Gergely Zsolt*Neptun-azonosító: *X8B97C*E-mail: [x8b97c@inf.elte.hu](mailto:x8b97c@inf.elte.hu)

Kurzuskód: IP-18PROGEGGyakorlatvezető neve: Bakonyi Viktória Judit

2020. január 7.

Tartalom

[Felhasználói dokumentáció 3](#_Toc410323328)

[Feladat 3](#_Toc410323329)

[Futási környezet 3](#_Toc410323330)

[Használat 3](#_Toc410323331)

[A program indítása 3](#_Toc410323332)

[A program bemenete 3](#_Toc410323333)

[A program kimenete 3](#_Toc410323334)

[Minta bemenet és kimenet 4](#_Toc410323335)

[Hibalehetőségek 4](#_Toc410323336)

[Fejlesztői dokumentáció 5](#_Toc410323337)

[Feladat 5](#_Toc410323338)

[Specifikáció 5](#_Toc410323339)

[Fejlesztői környezet 5](#_Toc410323340)

[Forráskód 5](#_Toc410323341)

[Megoldás 6](#_Toc410323342)

[Programparaméterek 6](#_Toc410323343)

[Programfelépítés 6](#_Toc410323344)

[Függvénystruktúra 6](#_Toc410323345)

[Algoritmus 6](#_Toc410323346)

[A kód 7](#_Toc410323347)

[Tesztelés 10](#_Toc410323348)

[Érvényes tesztesetek 10](#_Toc410323349)

[Érvénytelen tesztesetek 11](#_Toc410323350)

[Fejlesztési lehetőségek 11](#_Toc410323351)

2. Felhasználói dokumentáció
   1. Feladat

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 fog csökkenni!

* 1. Futási környezet

exe futtatására alkalmas, 64-bites operációs rendszer (pl. Windows 10). Nem igényel egeret.

* 1. Használat
     1. A program indítása

A program az "X8B97C.zip" néven található a tömörített állományban. A main.exe fájl kiválasztásával indítható.

* + 1. A program bemenete

A program az adatokat a **billentyűzet**ről olvassa be a következő sorrendben:

|  |  |  |
| --- | --- | --- |
| # | Adat | Magyarázat |
| **1.** | *N* | A települések száma (1≤N≤1000). |
| **2.** | *M* | A napok száma (1≤M≤1000). |
| **3.** | *H1,1* | Az első település első napjának maximális hőmérsékletértéke (‑50≤H1,1≤50). |
| **4.** | *H1,2* | Az első település második napjának maximális hőmérsékletértéke (-50≤H1,2≤50). |
| **...** | *…* |  |
| **N×M+2.** | *HN,M* | Az N-edik település M-edik napjának maximális hőmérsékletértéke (-50≤HN,M≤50). |

* + 1. A program kimenete

A program kiírja az azon települések T számát, ahol a hőmérséklet legalább egy héten át (7 alkalommal közvetlenül egymás után) csökken. Ezt ezen települések sorszáma követi növekvő sorrendben.

* + 1. Minta bemenet és kimenet

Telepulesek szama: 1

Napok szama: 8

Homerseklet: 3

Homerseklet: 2

Homerseklet: 1

Homerseklet: 0

Homerseklet: -1

Homerseklet: -2

Homerseklet: -3

Homerseklet: -4

A telepulesek szama: 1

A telepulesek: 1

Press any key to continue . . .

* + 1. Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha a települések vagy napok száma nem egész szám, vagy nem esik az **1..1000** intervallumba; vagy valamely hőmérsékletérték nem egész szám, vagy nem esik a **-50..50** intervallumba. Hiba esetén a program azzal jelzi a hibát, hogy újra kérdezi azt.

* + - 1. Mintafutás hibás bemeneti adatok esetén:

Telepulesek szama: sok

Telepulesek szama: 1

Napok szama: 1.1

Napok szama: 1

Homerseklet: -100

Hibas homerseklet. [-50..50] intervallumban kell lennie.

Homerseklet: -10

A telepulesek szama: 0

A telepulesek:

Press any key to continue . . .

1. Fejlesztői dokumentáció
   1. Feladat

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 fog csökkenni!

* 1. Specifikáció

**Bemenet**: N∈N, M∈N, H1,1..N,M∈NN×M

**Kimenet**: T∈N, Index∈NT

**Előfeltétel**: N∈[1..1000] ∧ M∈[1..1000] ∧ ∀i∈[1..N], ∀j∈[1..M]: Hi,j∈[-50..50]

**Utófeltétel**: T= ∧  
i∈IndexT 🡒 (i∈[1..N] ∧ ∃j∈[1..M-7]: ∀k∈[j..j+6]: Hi,k<Hi,k+1)

* 1. Fejlesztői környezet

exe futtatására alkalmas operációs rendszer (pl. Windows 10). mingw32-g++.exe c++ fordítóprogram (v8.2), Visual Stidio Code (v1.40) fejlesztői környezet.

* 1. Forráskód

A teljes fejlesztői anyag – kicsomagolás után – az X8B97C nevű könyvtárban található meg. A fej­lesztés során használt könyvtár-struktúra:

|  |  |
| --- | --- |
| Állomány | Magyarázat |
| X8B97C\main.exe | futtatható kód |
|  |  |
| X8B97C\main.cpp | C++ forráskód |
|  |  |
| X8B97C\be1.txt | teszt-bemeneti fájl1 |
| X8B97C\be2.txt | teszt-bemeneti fájl2 |
|  |  |
| X8B97C\X8B97C.docx | dokumentációk (ez a fájl) |

Megoldás

* + 1. Programparaméterek
       1. Konstans

1. MinN : **Egész**(1) [a települések minimális száma]  
   MaxN : **Egész**(1000) [a települések maximális száma]  
   MinM : **Egész**(1) [a napok minimális száma]  
   MaxM : **Egész**(1000) [a napok maximális száma]  
   MinH : **Egész**(-50) [a minimális hőmérséklet]  
   MaxH : **Egész**(50) [a maximális hőmérséklet]
   * + 1. Változó

N : **Egész**  
M : **Egész**  
T : **Egész**  
Kezdet : **Egész**  
Elozo : **Egész**  
Uj : **Egész**  
Hul : **Tömb**(1..MaxN:**Egész**)

* + 1. Programfelépítés

A program által használt modulok (és helyük):

main.cpp – program, a forráskönyvtárban  
iostream – képernyő-, és billentyűkezelés, a C++ rendszer része  
stdlib.h – általános rutinok, a C++ rendszer része  
limits – határértékek, a C++ rendszer része

* + 1. Függvénystruktúra



* + 1. A teljes program algoritmusa

Főprogram:

Változók:  
i, j: Egész

Kezdet: Egész

Elozo: Egész

Hul: Egész

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N = Beolvas() | | | | | | | |
| M = Beolvas() | | | | | | | |
| T := 0 | | | | | | | |
|  | i = 1..N | | | | | | |
|  | Elozo := -50 | | | | | | |
|  | Kezdet := 1 | | | | | | |
|  | j := 1 | | | | | | |
|  |  | | j <= M és j – Kezdet < 8 | | | | |
|  |  | | Uj = Beolvas() | | | | |
|  |  | | I | Uj >= Elozo | | | H |
|  |  | | Kezdet := j | | | - | |
|  |  | | Elozo := Uj | | | | |
|  |  | | j := j + 1 | | | | |
|  | Eldob() | | | | | | |
|  | I | j – Kezdet = 8 | | | | | H |
|  | T := T + 1 | | | | - | | |
|  | Hul[T] = i | | | |
| Kiir(T, Hul) | | | | | | | |

Alprogramok:

|  |
| --- |
| Beolvas(…) |
| | |
| Be: x |
| Beolvas = x |

|  |  |
| --- | --- |
| Eldob(db:Egész) | |
| | | |
|  | i = 1..db |
|  | Be: x |

|  |
| --- |
| Kiir(T:Egész, Hul:Tömb(1..MaxN:Egész)) |
| | |
| Ki: T |
| Ki: Hul |

* + 1. A kód

A main.cpp fájl tartalma:

/\*

  Készítette: Kovács Gergely Zsolt

  Neptun: X8B97C

  E-mail: x8b97c@inf.elte.hu

  Feladat: „ProgAlap beadandó feladatok” téma „Sokáig hidegebbé váló települések” feladat

\*/

#include <iostream>

#include <stdlib.h>

#include <limits>

using namespace std;

const int MinN = 1;

const int MaxN = 1000;

const int MinM = 1;

const int MaxM = 1000;

const int MinH = -50;

const int MaxH = 50;

bool HelyesN(int x)

{

    bool helyes = x >= MinN && x <= MaxN;

    if(!helyes) cerr << "Hibas telepulesszam. [" << MinN << ".." << MaxN << "] intervallumban kell lennie." << endl;

    return helyes;

}

bool HelyesM(int x)

{

    bool helyes = x >= MinM && x <= MaxM;

    if(!helyes) cerr << "Hibas napszam. [" << MinM << ".." << MaxM << "] intervallumban kell lennie." << endl;

    return helyes;

}

bool HelyesH(int x)

{

    bool helyes = x >= MinH && x <= MaxH;

    if(!helyes) cerr << "Hibas homerseklet. [" << MinH << ".." << MaxH << "] intervallumban kell lennie." << endl;

    return helyes;

}

int Beolvas(string Uzenet, bool Ellenorzo(int))

{

    int x;

    bool helyes;

    do

    {

        cerr << Uzenet; //ez a biro-s változatban kommentelve szerepel

        cin >> x;

        helyes = true;

        if (cin.fail() || !isspace(cin.peek()))

        {

            cin.clear();

            cin.ignore(numeric\_limits<streamsize>::max(), '\n');

            helyes = false;

        }

        if (helyes)

        {

            helyes = Ellenorzo(x);

        }

    } while (!helyes);

    return x;

}

void Kiir(int T, int Hul[MaxN])

{

    cerr << "A telepulesek szama: ";

    cout << T;

    cerr << endl << "A telepulesek: ";

    for (int i = 1; i <= T; i++)

    {

        cout << " " << Hul[i-1];

    }

    cout << endl;

}

void Eldob(int db){

    int x;

    for (int i = 1; i <= db; i++)

    {

        cin >> x;

    }

    // cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

void BillreVar(){

    system("pause");

}

int main()

{

    int Hul[1000], T, N, M, Kezdet, Elozo, Uj;

    N = Beolvas("Telepulesek szama: ", HelyesN);

    M = Beolvas("Napok szama: ", HelyesM);

    T = 0;

    for (int i = 1; i <= N; i++)

    {

        Elozo = -50;

        Kezdet = 1;

        int j = 1;

        while (j <= M && j - Kezdet < 8)

        {

            Uj = Beolvas("Homerseklet: ", HelyesH);

            if (Uj >= Elozo)

            {

                Kezdet = j;

            }

            Elozo = Uj;

            j++;

        }

        Eldob(M-j+1);

        if (j - Kezdet == 8)

        {

            T++;

            Hul[T-1] = i;

        }

    }

    Kiir(T, Hul);

    BillreVar();

    return 0;

}

* 1. Tesztelés
     1. Érvényes tesztesetek
        1. teszteset

|  |
| --- |
| Bemenet – *egy település, egy nap* |
| N = 1  M = 1  H1,1 = 0 |
| Kimenet |
| 0 |

* + - 1. teszteset

|  |
| --- |
| Bemenet – *egy település, hét nap folyamatos csökkenéssel* |
| N = 1 M = 8 H1,1 = 3  H1,2 = 2  H1,3 = 1  H1,4 = 0  H1,5 = -1  H1,6 = -2  H1,7 = -3  H1,8 = -4 |
| Kimenet |
| T = 1  Hul1 = 1 |

* + - 1. teszteset: be1.txt

|  |
| --- |
| Bemenet – *három település, az elsőn folyamatos csökkenés* |
| N = … |
| H1,1 = …  … |
| Kimenet |
| … |

* + - 1. teszteset

|  |
| --- |
| Bemenet – *nincs megfelelő település* |
| N = … |
| M = …  … |
| Kimenet |
| … |

* + - 1. teszteset

|  |
| --- |
| Bemenet – *csak megfelelő település van* |
| N = … |
| M = …  … |
| Kimenet |
| … |

* + 1. Érvénytelen tesztesetek
       1. teszteset

|  |
| --- |
| Bemenet – *Rossz településszám* |
| N = 11tizenegy |
| Kimenet |
| Újrakérdezés: N = |

* + - 1. teszteset

|  |
| --- |
| Bemenet – *Rossz napszám* |
| N = 11  M = -1 |
| Kimenet |
| Újrakérdezés:  M = |

…

* + - 1. 8. teszteset

…

* 1. Fejlesztési lehetőségek

1. Adatok – a felhasználó igénye szerint – akár fájlból is fogadása.
2. Hibás fájl-bemenetek felismerése, és a hiba helyének (sor sorszámának) kiírása.
3. Többszöri futtatás megszervezése
4. A bemeneti sorozat grafikus megjelenítése, s az eredmény-szigetek elütő színű kijelzése.