„Programozási alapismeretek”  
beadandó feladat

Készítette: *Magyar Tamás*Neptun-azonosító: *rnyr2f*E-mail: [rnyr2f@inf.elte.hu](mailto:rnyr2f@inf.elte.hu)

Kurzuskód: IP-18PROGEGGyakorlatvezető neve: *Klettner Péter*

2020. január 4.

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 8](#_Toc410323347)

[Tesztelés 11](#_Toc410323348)

[Érvényes tesztesetek 11](#_Toc410323349)

[Érvénytelen tesztesetek 12](#_Toc410323350)

[Fejlesztési lehetőségek 13](#_Toc410323351)

2. Felhasználói dokumentáció
   1. Feladat
3. A meteorológiai intézet az ország N településére adott M napos időjárás előrejelzést, az adott településen az adott napra várt legmagasabb hőmérsékletet. Egy napon az a leghűvösebb település, amelyre azon a napon a legkisebb hőmérsékletet jósolják. A program megadja azokat a napokat, amelyeken a leghűvösebb településen a lehető legnagyobb az előre jelzett hőmérséklet!
   1. Futási környezet

IBM PC, exe futtatására alkalmas, 32-bites operációs rendszer (pl. Windows 10). Nem igényel egeret.

* 1. Használat
     1. A program indítása

A program az rnyr2f\bin\Release\rnyr2f.exe néven található a tömörített állományban. A rnyr2f.exe fájl kiválasztásával indítható.

* + 1. A program bemenete

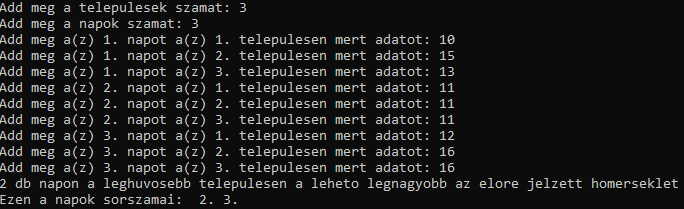
A program az adatokat a **billentyűzet**ről olvassa be a következő sorrendben:

|  |  |  |
| --- | --- | --- |
| # | Adat | Magyarázat |
| **1.** | *N* | A települések száma (1≤N≤1000) |
| **2.** | *M* | A napok száma (1≤M≤1000) |
| **3.** | *Hőmérséklet11* | 1. településen jósolt 1. napi hőmérséklet értéke található (-50≤Hi,j≤50). |
| **4.** | *Hőmérséklet12* | 1. településen jósolt 2. napi hőmérséklet értéke található (-50≤Hi,j≤50). |
| **…** | *…* |  |
| **N×M+2.** | *HőmérsékletNM* | N. településen jósolt M. napi hőmérséklet értéke található (-50≤Hi,j≤50). |

* + 1. A program kimenete

Az első sorban azon napok K száma, amelyeken a leghűvösebb településen a lehető legnagyobb az előre jelzett hőmérséklet. Ezt ezen napok sorszámai követik, növekvő sorrendben.

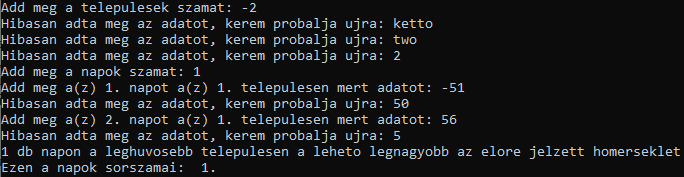
* + 1. Minta bemenet és kimenet



* + 1. Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha a városok vagy napok száma nem egész szám, vagy nem esik a **1..1 000** intervallumba; vagy valamely hőmérséklet érték nem egész szám, vagy nem esik a **-50..50** intervallumba. Hiba esetén a program azzal jelzi a hibát, hogy hibát ír ki és újra kérdezi azt.

* + - 1. Mintafutás hibás bemeneti adatok esetén:



1. Fejlesztői dokumentáció
   1. Feladat
2. A meteorológiai intézet az ország N településére adott M napos időjárás előrejelzést, az adott településen az adott napra várt legmagasabb hőmérsékletet. Egy napon az a leghűvösebb település, amelyre azon a napon a legkisebb hőmérsékletet jósolják. Készíts programot, amely azokat a napokat, amelyeken a leghűvösebb településen a lehető legnagyobb az előre jelzett hőmérséklet!
   1. Specifikáció

**Bemenet**:

**Kimenet**:

**Előfeltétel**:

**Utófeltétel**:

* 1. Fejlesztői környezet

IBM PC, exe futtatására alkalmas operációs rendszer (pl. Windows 10). mingw32-g++.exe c++ fordítóprogram (v4.7), Code::Blocks (v17.12) fejlesztői környezet.

* 1. Forráskód

A teljes fejlesztői anyag –kicsomagolás után– az rnyr2f nevű könyvtárban található meg. A fej­lesztés során használt könyvtár-struktúra:

|  |  |
| --- | --- |
| Állomány | Magyarázat |
| rnyr2f\bin\Release\rnyr2f.exe | futtatható kód |
| rnyr2f\obj\Release\main.o | félig lefordított kód |
|  |  |
| rnyr2f\main.cpp | C++ forráskód |
|  |  |
| rnyr2f\teszt1.txt | teszt-bemeneti fájl1 |
| rnyr2f\teszt2.txt | teszt-bemeneti fájl2 |
| rnyr2f\teszt3.txt | teszt-bemeneti fájl3 |
| rnyr2f\teszt4.txt | teszt-bemeneti fájl4 |
| rnyr2f\teszt5.txt | teszt-bemeneti fájl5 |
|  |  |
| rnyr2f\doc\rnyr2f.docx | dokumentációk (ez a fájl) |

* 1. Megoldás
     1. Programparaméterek
        1. Konstans

1. Max : **Egész**(1000)
   * + 1. Változó

N : **Egész**

M : **Egész**

db : **Egész**

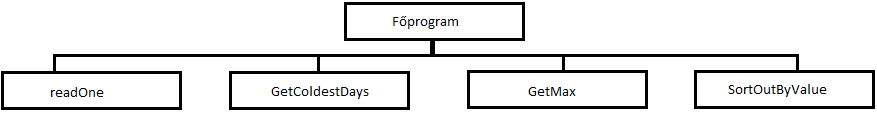
tempetures : **Mátrix**(1..N:**Egész,**Max)  
colds = **Tömb**(1..M:**Egész**)  
hotcolds = **Tömb**(1..M:**Egész**)

* + 1. Programfelépítés

A program által használt modulok (és helyük):

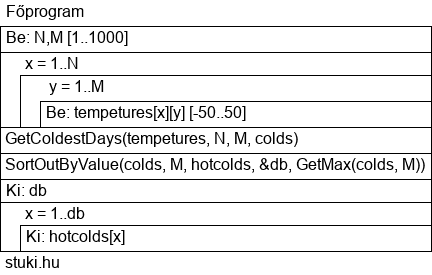
main.cpp – program, a forráskönyvtárban  
iostream – képernyő-, és billentyűkezelés, a C++ rendszer része  
sstream – Stringstream kezelő könyvtár, a C++ rendszer része

* + 1. Függvénystruktúra



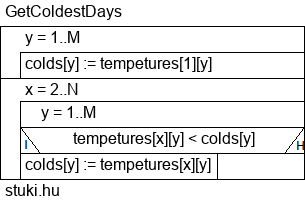
* + 1. A teljes program algoritmusa

**Főprogram**:

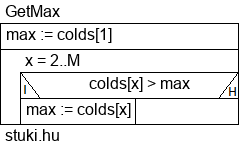


**Alprogramok**:

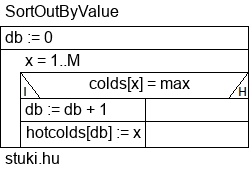
**GetColdestDays (tempetures, N, M, colds)**

****

**GetMax (colds, M)**

****

**SortOutByBalue (colds, M, hotcolds, db, max)**

****

* + 1. A kód

A main.cpp fájl tartalma:

/\*

 Készítette: Magyar Tamás

 Neptun: RNYR2F

 E-mail: rnyr2f@inf.elte.hu

 Feladat:

    A meteorológiai intézet az ország N településére adott M napos időjárás előrejelzést, az adott településen az adott napra várt legmagasabb hőmérsékletet.

    Egy napon az a leghűvösebb település, amelyre azon a napon a legkisebb hőmérsékletet jósolják.

    Készíts programot, amely azokat a napokat, amelyeken a leghűvösebb településen a lehető legnagyobb az előre jelzett hőmérséklet!

\*/

#include <iostream> //Standart Input/Output

#include <sstream> //Stringstream a kiírások miatt

//#define BIRO

using namespace std;

typedef short int SInt;

const SInt MAX = 1000;

void readOne(SInt \*num, string text, int min, int max);

void GetColdestDays(SInt tempetures[][MAX], SInt N, SInt M, SInt colds[]);

int GetMax(SInt colds[], SInt M);

void SortOutByValue(SInt colds[], SInt M, SInt hotcolds[], SInt \*db, SInt max);

int main()

{

    ios\_base::sync\_with\_stdio(false);

    SInt N,M;

    readOne(&N, "Add meg a telepulesek szamat: ", 1, 1000);

    readOne(&M, "Add meg a napok szamat: ", 1, 1000);

    SInt tempetures[N+1][MAX];

    SInt colds[M+1];

    SInt hotcolds[M+1];

    SInt db;

    //beolvasás

    for(int x=1;x<=N;++x)

        for(int y=1;y<=M;++y)

        {

            #ifdef BIRO

                readOne(&tempetures[x][y], "", -50, 50);

            #else

                string str;

                stringstream ss;

                ss.str("");

                ss << "Add meg a(z) "; ss << x;

                ss << ". napot a(z) "; ss << y;

                ss << ". telepulesen mert adatot: ";

                getline(ss, str);

                readOne(&tempetures[x][y], str, -50, 50);

            #endif

        }

    //feldolgozás

    GetColdestDays(tempetures, N, M, colds);

    SortOutByValue(colds, M, hotcolds, &db, GetMax(colds, M));

    //kiírása

    #ifdef BIRO

        cout << db;

        for(int x=1;x<=db;++x)

            cout << " " << hotcolds[x];

    #else

        cout << db;

        cout << " db napon a leghuvosebb telepulesen a leheto legnagyobb az elore jelzett homerseklet" << endl << "Ezen a napok sorszamai: ";

        for(int x=1;x<=db;++x)

            cout << " " << hotcolds[x] << ".";

    #endif

    return 0;

}

//alprogramok

void readOne(SInt \*num, string text, int min, int max)

{

    #ifdef BIRO

        cin >> \*num;

    #else

        cout << text;

        while(!(cin >> \*num) || \*num > max || \*num < min)

        {

            cin.clear();

            cin.ignore(1000, '\n');

            cout << "Hibasan adta meg az adatot, kerem probalja ujra: ";

        }

    #endif

}

void GetColdestDays(SInt tempetures[][MAX], SInt N, SInt M, SInt colds[])

{

    for(int y=1;y<=M;++y)

        colds[y] = tempetures[1][y];

    for(int x=2;x<=N;++x)

        for(int y=1;y<=M;++y)

            if(tempetures[x][y] < colds[y])

                colds[y] = tempetures[x][y];

}

int GetMax(SInt colds[], SInt M)

{

    SInt max = colds[1];

    for(int x=2;x<=M;++x)

        if(colds[x] > max)

            max = colds[x];

    return max;

}

void SortOutByValue(SInt colds[], SInt M, SInt hotcolds[], SInt \*db, SInt max)

{

    \*db = 0;

    for(int x=1;x<=M;++x)

        if(colds[x] == max)

            hotcolds[++(\*db)] = x;

}

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

|  |
| --- |
| Bemenet – *legalább kettő település* |
| N = 3  M = 5  Hőmérséklet11 = 10  Hőmérséklet12 = 15  Hőmérséklet13 = 12  Hőmérséklet14 = 10  Hőmérséklet15 = 10  Hőmérséklet21 = 11  Hőmérséklet22 = 11  Hőmérséklet23 = 11  Hőmérséklet24 = 11  Hőmérséklet25 = 20  Hőmérséklet31 = 12  Hőmérséklet32 = 16  Hőmérséklet33 = 16  Hőmérséklet34 = 16  Hőmérséklet35 = 20 |
| Kimenet |
| 2 2 3 |

* + - 1. teszteset: teszt2.txt

|  |
| --- |
| Bemenet – *maximás bemenet* |
| N = 1000  M = 1000  Hőmérséklet11 = 36  Hőmérséklet12 = 34  **…** |
| Kimenet |
| 1000 1 2 3 **…** |

* + - 1. teszteset: teszt3.txt

|  |
| --- |
| Bemenet – *minimális bemenet* |
| N = 1  M = 1  Hőmérséklet11 = 3 |
| Kimenet |
| 1 1 |

* + - 1. teszteset: be4.txt

|  |
| --- |
| Bemenet – *csak egy település* |
| N = 2 |
| M = 2  Hőmérséklet11 = 2  Hőmérséklet12 = 1  Hőmérséklet21 = 2  Hőmérséklet22 = 2 |
| Kimenet |
| 1 1 |

* + - 1. teszteset: be5.txt

|  |
| --- |
| Bemenet – *teljes intervallumot betöltő adatok* |
| N = 5 |
| M = 5  Hőmérséklet11 = 13  Hőmérséklet12 = -41  Hőmérséklet13 = -28  Hőmérséklet14 = 5  **…** |
| Kimenet |
| 1 5 |

* + 1. Érvénytelen tesztesetek
       1. teszteset

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

* + - 1. teszteset

|  |
| --- |
| Bemenet – *Rossz napok száma* |
| N = 11  M = 1001 |
| Kimenet |
| Újrakérdezés: M = |

* + - 1. teszteset

|  |
| --- |
| Bemenet – *Rossz hőmérséklet* |
| N = 11  M = 12  Magasság1 = -55 |
| Kimenet |
| Újrakérdezés:  Magasság1 = |

* 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. Algoritmus opimalizálása: az adatok beolvasás utáni azzonali feldolgozása
5. Grafikus felhasználóbarátabb felület megszervezése