„Programozási alapismeretek”  
beadandó feladat

Készítette: *Gulyás Simone*Neptun-azonosító: *H2E9AQ*E-mail: [h2e9aq@inf.elte.hu](mailto:h2e9aq@inf.elte.hu)

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

2019. december 27.

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 6](#_Toc410323341)

[Megoldás 6](#_Toc410323342)

[Programparaméterek 6](#_Toc410323343)

[Programfelépítés 6](#_Toc410323344)

[Függvénystruktúra 6](#_Toc410323345)

[Algoritmus 7](#_Toc410323346)

[A kód 7](#_Toc410323347)

[Tesztelés 9](#_Toc410323348)

[Érvényes tesztesetek 9](#_Toc410323349)

[Érvénytelen tesztesetek 10](#_Toc410323350)

[Fejlesztési lehetőségek 10](#_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

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

* 1. Használat
     1. A program indítása

A program a h2e9aq\bin\Debug\assignment\_complex.exe néven található a tömörített állományban. A assignment\_complex.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.** | *H0 0* | Az első bevitt hőmérsékleti adat (-50≤H0≤50). A kódban ez az adat egy változóban tárolódik el a vizsgálat idejére. |
| **4.** | *H0 1* | Az második bevitt hőmérsékleti adat (-50≤H1≤50). A kódban ez az adat egy változóban tárolódik el a vizsgálat idejére. |
| **...** | *…* |  |
| **N\*M+2.** | *HN-1 M-1* | Az M-edik bevitt magasság (-50≤HM-1≤50). A kódban ez az adat egy változóban tárolódik el a vizsgálat idejére. |
| **N\*M +3.** | *He* | Az előzően bevitt hőmérsékleti adat tárolódik. Hi-1 tárolódik benne, ahol az i 1 és M-1 közé eső egész szám (-50≤He≤50). |
| **N\*M +4.** | *Hu* | Az aktuálisan bevitt hőmérsékleti adat tárolódik. Hi tárolódik benne, ahol az i 0 és M-1 közé eső egész szám (-50≤Hu≤50). |

* + 1. A program kimenete

A program kiírja a települések számát és a településeket növekvő sorrendben, ahol egymás után közvetlenül 7 alkalommal fog csökkeni a hőmérséklet.

* + 1. Minta bemenet és kimenet

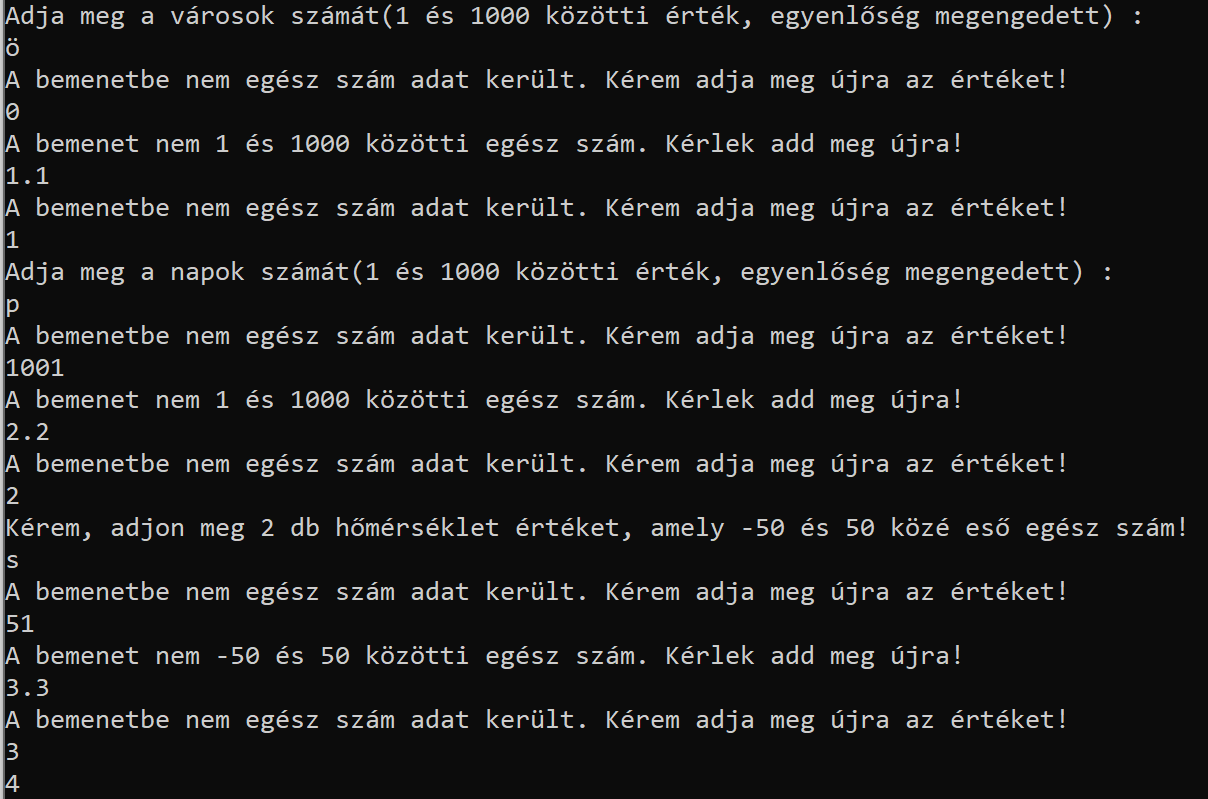
A képen képernyőkép, fekete látható

Automatikusan generált leírás

* + 1. Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha a települések vagy a napok száma nem egész szám, vagy nem esik a **1..1000** intervallumba; vagy valamely hőmérsékleti érték 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:



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. Készíts programot, amely megadja azokat a településeket, ahol a hőmérséklet legalább egy héten át fog csökkenni!
   1. Specifikáció

**Bemenet**: N ∈N, M ∈N, H0..M-1∈NM, He∈N, Hu∈N, db∈N, csokkenE∈L

**Kimenet**: T ∈N, homerseklet\_csokkeno\_varosok0..T-1∈NT

**Előfeltétel**: N ∈[1..1000] ∧ M ∈[1..1000] ∧∀i∈[0.. M-1]: H 0..M∈[-50**..** 50], He=-50, db=0

**Utófeltétel**: i∈[0.. N -1] ∧

csokkenE=Ǝj∈[1.. M -1] ∧ ∧ Hi j=Hu ∧ Hi j-1=He : He> Hu ∧

Ǝj∈[1.. M -1] ∧ ∧

db= ∧

T = ∧

k∈[0.. T -1] : homerseklet\_csokkeno\_varosokk=i+1

**Megjegyzés**: a „ha nincs ilyen” kitételt (a csokkenE=Hamis esetben) a program egyetlen 0 kiírásával fog­ja jelezni a kísérő szöveg után.

* 1. Fejlesztői környezet

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

* 1. Forráskód

A teljes fejlesztői anyag –kicsomagolás után– az H2E9Aq nevű könyvtárban található meg. A fej­lesztés során használt könyvtár-struktúra:

|  |  |
| --- | --- |
| Állomány | Magyarázat |
| H2E9AQ\bin\Debug\assignment\_complex.exe | futtatható kód |
| H2E9AQ\obj\Debug\main.o | félig lefordított kód |
|  |  |
| H2E9AQ\main.cpp | C++ forráskód |
|  |  |
| H2E9AQ\teszt1.txt | teszt-bemeneti fájl1 |
| H2E9AQ\teszt2.txt | teszt-bemeneti fájl2 |
| H2E9AQ\teszt3.txt | teszt-bemeneti fájl3 |
| H2E9AQ\teszt4.txt | teszt-bemeneti fájl4 |
| H2E9AQ\teszt5.txt | teszt-bemeneti fájl5 |
|  |  |
| H2E9AQ\dokumentum\h2e9aq.docx | dokumentáció |

* 1. Megoldás
     1. Programparaméterek
        1. Konstans

1. MaxNM : **Egész**(1000) [maximális település és napok száma]
2. MinNM : **Egész**(1) [minimális település és napok száma]
3. MaxH : **Egész**(50) [maximális hőmérséklet értéke]
4. MinH : **Egész**(-50) [minimális hőmérséklet értéke]
   * + 1. Típus
5. Kimenet = **Struct**(T: **Egész**, homerseklet\_csokkeno\_varosok:**Egészeket tartalmazó vektor**)
   * + 1. Változó

N : **Egész**  
M : **Egész**  
He : **Egész**

Hu : **Egész**

szam : **Egész** [szám vizsgálathoz]

sstr : **StringStream** [szám vizsgálathoz]

megfelelosseg : **bool** [tartomány vizsgálathoz]

bemenet : **String** [szám vizsgálathoz]

kapcsolo : **String** [tartomány vizsgálathoz]

T, homerseklet\_csokkeno\_varosok : **Kimenet**

* + 1. Programfelépítés

A program által használt modulok (és helyük):

main.cpp – program, a forráskönyvtárban  
iostream – képernyő-, és billentyűkezelés, a C++ rendszer része  
sstream – a bemenetek helyességének vizsgálatához

vector – a kimenet megfelelő elemszám eltárolásához

* + 1. Függvénystruktúra



* + 1. A teljes program algoritmusa

Főprogram:

A képen képernyőkép látható

Automatikusan generált leírás

Alprogramok:

A képen képernyőkép látható

Automatikusan generált leírás

**A képen képernyőkép látható

Automatikusan generált leírás**

**A képen képernyőkép látható

Automatikusan generált leírás**

**A képen képernyőkép látható

Automatikusan generált leírás**

* + 1. A kód

A main.cpp fájl tartalma:

/\*

Készítette: Gulyás Simone

Neptun: H2E9AQ

E-mail: h2e9aq@inf.elte.hu

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

\*/

#include <iostream>

#include <sstream>

#include <vector>

#define MaxNM 1000

#define MinNM 1

#define MaxH 50

#define MinH -50

//#define BIRO

using namespace std;

struct Kimenet

{

int T;

vector<int> homerseklet\_csokkeno\_varosok;

};

int SzamVizsgalat(string u, string kapcsolo);

bool ErtekkeszletVizsgalat(int bemenet, string kapcsolo);

Kimenet HomersekletCsokkenesVizsgalat(int N, int M);

void Kiir(Kimenet kimenet);

int main()

{

//Deklaráció

int N, M;

string bemenet;

cerr << "Adja meg a városok számát(1 és 1000 közötti érték, egyenlőség megengedett) : " << endl;

#ifdef BIRO

cin >> N;

#else

//Városok számának megadása és bemenet helyességének vizsgálata

cin >> bemenet;

N=SzamVizsgalat(bemenet, "települések száma");

#endif // BIRO

cerr << "Adja meg a napok számát(1 és 1000 közötti érték, egyenlőség megengedett) : " << endl;

#ifdef BIRO

cin >> M;

#else

//Városok számának megadása és bemenet helyességének vizsgálata

cin >> bemenet;

M=SzamVizsgalat(bemenet, "napok száma");

#endif // BIRO

Kimenet kimenet;

cerr << "Kérem, adjon meg " << N\*M << " db hőmérséklet értéket, amely -50 és 50 közé eső egész szám!" << endl;

kimenet=HomersekletCsokkenesVizsgalat(N, M);

Kiir(kimenet);

return 0;

}

int SzamVizsgalat(string bemenet, string kapcsolo)

{

int szam;

stringstream sstr;

sstr.str("");

sstr.clear();

sstr << bemenet;

sstr >> szam;

while(sstr.fail() || !sstr.eof() || !ErtekkeszletVizsgalat(szam, kapcsolo))

{

if(sstr.fail() || !sstr.eof())

{

cerr << "A bemenetbe nem egész szám adat került. Kérem adja meg újra az értéket! " << endl;

}

sstr.clear();

sstr.str("");

bemenet="";

cin >> bemenet;

sstr << bemenet;

sstr>>szam;

}

return szam;

}

bool ErtekkeszletVizsgalat(int bemenet, string kapcsolo)

{

bool megfelelosseg=true;

if(kapcsolo.compare("települések száma")==0 || kapcsolo.compare("napok száma")==0 )

{

if(MinNM>bemenet || MaxNM<bemenet)

{

cerr << "A bemenet nem 1 és 1000 közötti egész szám. Kérlek add meg újra!" << endl;

megfelelosseg=false;

}

}

else if(kapcsolo.compare("hőmérséklet")==0)

{

if(MinH>bemenet || MaxH<bemenet)

{

cerr << "A bemenet nem -50 és 50 közötti egész szám. Kérlek add meg újra!" << endl;

megfelelosseg=false;

}

}

else

{

cerr << "Nem megfelelő kapcsoló, kódhiba" << endl;

}

return megfelelosseg;

}

Kimenet HomersekletCsokkenesVizsgalat(int N, int M)

{

Kimenet kimenet;

kimenet.T=0;

int He, Hu;

string bemenet;

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

{

int db=0;

He=-50;

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

{

#ifdef BIRO

cin >> Hu;

#else

//Városok számának megadása és bemenet helyességének vizsgálata

cin >> bemenet;

Hu=SzamVizsgalat(bemenet, "hőmérséklet");

#endif // BIRO

if(db<7)

{

bool csokkenE=He>Hu;

if(csokkenE)

{

db++;

}

else

{

db=0;

}

if(db==7)

{

kimenet.homerseklet\_csokkeno\_varosok.push\_back(i+1);

kimenet.T++;

}

He=Hu;

}

}

}

return kimenet;

}

void Kiir(Kimenet kimenet)

{

cerr << "7 alkalommal közvetlenül egymás után csökkenő hőmérsékletű városok száma es felsorolása: " << endl;

cout << kimenet.T << " ";

for (int i = 0; i<kimenet.T; i++)

{

cout << kimenet.homerseklet\_csokkeno\_varosok[i] << " ";

}

}

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

|  |
| --- |
| Bemenet – *Minimális települések száma, minimális napok száma, maximális hőmérséklet* |
| N = 1  M = 1  H2 = 50 |
| Kimenet |
| 0 |

* + - 1. teszteset: teszt2.txt

|  |
| --- |
| Bemenet – *Néhány település, legalább 7 a napok száma, van közöttük egy minimális hőmérséklet, nincs 7 napon keresztül hőmérséklet csökkenés* |
| N=3  M=8  H0 0=16  H0 1=15  H0 2=12  H0 3=10  H0 4=19  H0 5=8  H0 6=3  H0 7=-50  H1 0=11  H1 1=23  H1 2=22  H1 3=21  H1 4=20  H1 5=13  H1 6=12  H1 7=11  H2 0=12  H2 1=16  H2 2=17  H2 3=18  H2 4=20  H2 5=21  H2 6=22  H2 7=23 |
| Kimenet |
| 0 |

* + - 1. teszteset: teszt3.txt

|  |
| --- |
| Bemenet – *van 1 városban 7 napon keresztül hőmérséklet csökkenés* |
| N=3  M=8  H0 0=16  H0 1=15  H0 2=12  H0 3=10  H0 4=29  H0 5=8  H0 6=3  H0 7=-1  H1 0=24  H1 1=23  H1 2=22  H1 3=21  H1 4=20  H1 5=13  H1 6=12  H1 7=11  H2 0=12  H2 1=16  H2 2=17  H2 3=18  H2 4=20  H2 5=21  H2 6=22  H2 7=23 |
|  |
| Kimenet |
| T=1  homerseklet\_csokkeno\_varosok0=2 |

* + - 1. teszteset: teszt4.txt

|  |
| --- |
| Bemenet – *van több városban 7 napon keresztül hőmérséklet csökkenés* |
| N=3  M=8  H0 0=16  H0 1=15  H0 2=12  H0 3=10  H0 4=9  H0 5=8  H0 6=3  H0 7=-1  H1 0=24  H1 1=23  H1 2=22  H1 3=21  H1 4=20  H1 5=13  H1 6=12  H1 7=11  H2 0=12  H2 1=16  H2 2=17  H2 3=18  H2 4=20  H2 5=21  H2 6=22  H2 7=23 |
| Kimenet |
| T=1  homerseklet\_csokkeno\_varosok0=1  homerseklet\_csokkeno\_varosok1=2 |

* + - 1. teszteset: teszt5.txt

|  |
| --- |
| Bemenet – *van több városban 7 napon keresztül hőmérséklet csökkenés, az első, utolsó, és közbülső napokban* |
| N=3  M=8  H0 0=16  H0 1=15  H0 2=12  H0 3=10  H0 4=9  H0 5=8  H0 6=3  H0 7=-1  H0 8=9  H0 9=8  H1 0=8  H1 1=24  H1 2=24  H1 3=23  H1 4=22  H1 5=21  H1 6=20  H1 7=13  H1 8=12  H1 9=11  H2 0=12  H2 1=16  H2 2=15  H2 3=13  H2 4=12  H2 5=10  H2 6=9  H2 7=8  H2 8=7  H2 9=23 |
| Kimenet |
| T=3  homerseklet\_csokkeno\_varosok0=1  homerseklet\_csokkeno\_varosok1=2  homerseklet\_csokkeno\_varosok2=3 |

* + 1. Érvénytelen tesztesetek
       1. teszteset:

|  |
| --- |
| Bemenet – *Túl kevés települések száma* |
| N = 0 |
| Kimenet |
| Újrakérdezés: N = |

* + - 1. teszteset:

|  |
| --- |
| Bemenet – *Túl sok települések száma* |
| N = 1001 |
| Kimenet |
| Újrakérdezés:  N= |

* + - 1. teszteset:

|  |
| --- |
| Bemenet – *Nem egész szám a települések száma* |
| N = 1.1 |
| Kimenet |
| Újrakérdezés:  N= |

* + - 1. teszteset:

|  |
| --- |
| Bemenet – *Nem szám a települések száma* |
| N = 11valami |
| Kimenet |
| Újrakérdezés:  N= |

* + - 1. teszteset:

|  |
| --- |
| Bemenet – *Túl kevés a napok száma* |
| N=10  M = 0 |
| Kimenet |
| Újrakérdezés: M = |

* + - 1. teszteset:

|  |
| --- |
| Bemenet – *Túl sok a napok száma* |
| N=10  M = 1001 |
| Kimenet |
| Újrakérdezés:  M= |

* + - 1. teszteset:

|  |
| --- |
| Bemenet – *Nem egész szám a napok száma* |
| N=10  M = 1.1 |
| Kimenet |
| Újrakérdezés:  M= |

* + - 1. teszteset:

|  |
| --- |
| Bemenet – *Nem szám a napok száma* |
| N=10  M = 1valami0 |
| Kimenet |
| Újrakérdezés:  M= |

* + - 1. teszteset:

|  |
| --- |
| Bemenet – *Túl kicsi a hőmérséklet* |
| N=10  M = 10  H0 0=-51 |
| Kimenet |
| Újrakérdezés: H0 0= |

* + - 1. teszteset:

|  |
| --- |
| Bemenet – *Túl nagy a hőmérséklet* |
| N=10  M = 10  H0 0=51 |
| Kimenet |
| Újrakérdezés:  H0 0= |

* + - 1. teszteset:

|  |
| --- |
| Bemenet – *Nem egész szám a hőmérséklet* |
| N=10  M = 10  H0 0=-5.1 |
| Kimenet |
| Újrakérdezés:  H0 0= |

* + - 1. teszteset:

|  |
| --- |
| Bemenet – *Nem szám a hőmérséklet* |
| N=10  M = 10  H0 0=5ejha1 |
| Kimenet |
| Újrakérdezés:  H0 0= |

* 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.