„Programozási alapismeretek”  
beadandó feladat

Készítette: *Beregi Bence Zsolt*Neptun-azonosító: *DQK6TE*E-mail: beregi.bence@hotmail.com

Kurzuskód: IP-18PROGEGGyakorlatvezető neve: *Daiki Tennó*

2020. január 4.

Tartalom

[Felhasználói dokumentáció 3](#_Toc29080251)

[Feladat 3](#_Toc29080252)

[Futási környezet 3](#_Toc29080253)

[Használat 3](#_Toc29080254)

[A program indítása 3](#_Toc29080255)

[A program bemenete 3](#_Toc29080256)

[A program kimenete 3](#_Toc29080257)

[Minta bemenet és kimenet 3](#_Toc29080258)

[Hibalehetőségek 4](#_Toc29080259)

[Fejlesztői dokumentáció 5](#_Toc29080260)

[Feladat 5](#_Toc29080261)

[Specifikáció 5](#_Toc29080262)

[Fejlesztői környezet 5](#_Toc29080263)

[Forráskód 6](#_Toc29080264)

[Megoldás 6](#_Toc29080265)

[Programparaméterek 6](#_Toc29080266)

[Programfelépítés 6](#_Toc29080267)

[Függvénystruktúra 6](#_Toc29080268)

[A teljes program algoritmusa 7](#_Toc29080269)

[A kód 7](#_Toc29080270)

[Tesztelés 10](#_Toc29080271)

[Érvényes tesztesetek 10](#_Toc29080272)

[Érvénytelen tesztesetek 11](#_Toc29080273)

[Fejlesztési lehetőségek 11](#_Toc29080274)

2. Felhasználói dokumentáció
   1. Feladat

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

Készíts programot, amely megadja azokat a településeket, ahol az átlagtól a maximum jobban el-tér, mint a minimum!

* 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 az meterologia\bin\Debug\meterologia.exe néven található a tömörített állományban. A meterologia.exe fájl kiválasztásával indítható.

* + 1. A program bemenete

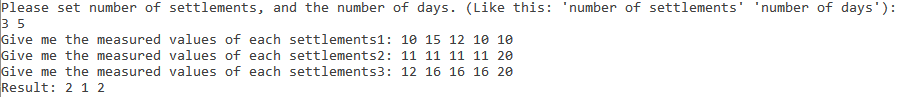
A program az adatokat a **billentyűzet**ről olvassa be a következő sorrendben:

|  |  |  |
| --- | --- | --- |
| # | Adat | Magyarázat |
| **1.** | *N* | Települések szám (1≤N≤1000). |
| **2.** | *M* | Napok száma (1≤M≤1000). |
| **3.** | *Település1* | Első település mérései naponként (H1, H2…Hm)(-50≤H≤50) |
| **4.** | *Település2* | Második település mérései naponként (H1, H2…Hm)(-50≤H≤50) |
| **...** | *…* |  |
| **N+1.** | *TelepülésN* | Az N-edik település mérései naponként (H1, H2…Hm)(-50≤H≤50) |

* + 1. A program kimenete

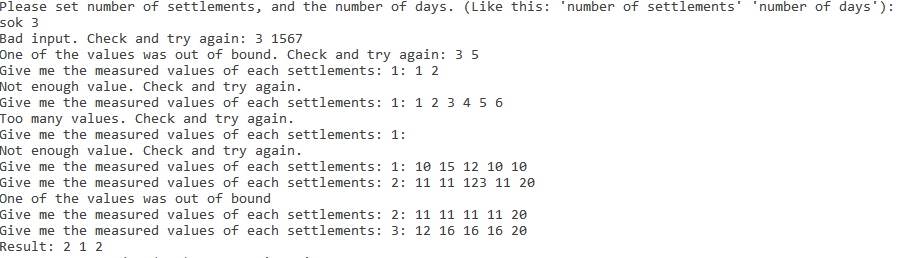
A program kiírja, hogy hány darab településen volt tért el az átlagtól jobban a maximum, mint a minimum, és ezután a települések számát, növekvő sorrendben.

* + 1. Minta bemenet és kimenet



* + 1. Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha a napok/települések száma nem egész szám, vagy nem esik a **1..10 00** intervallumba; vagy valamely jósolt hőmérséklet nem egész szám, vagy nem esik a **-50..50** intervallumba. Hiba esetén a program azzal jelzi a hibát, hogy kiírja azt és újra kérdezi azt.

* + - 1. Mintafutás hibás bemeneti adatok esetén:

1. Fejlesztői dokumentáció
   1. Feladat

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

Készíts programot, amely megadja azokat a településeket, ahol az átlagtól a maximum jobban el-tér, mint a minimum!

* 1. Specifikáció

**Bemenet**: N∈N, M∈N, Napok∈NN\*M

**Kimenet**: dbIgaz∈N, telepId∈N\*

**Előfeltétel**: N=Hossz(Napok első dimenziója) ∧ M = Hossz(Napok második dimenziója) ∧ N,M∈[1..1000] ∧ ∀i∈[0..N-1] ∀j∈[0..M-1]: Napoki,j∈[-50..50]

**Utófeltétel**: out, avg, dailySum, minarr, maxarr∈NN ∧

∀i∈[0..N]: dailySumi = ∧

∀i∈[0..N]: avgi = dailySumi / M ∧

∀i∈[0..N]: maxarri = (Napoki,j) ∧

∀i∈[0..N]: minarri = (Napoki,j) ∧

out = ∀i∈[0..N]: ha (|avgi – maxarri| > | avgi – minarri |) akkor outi = i+1 különben

outi = 0 ∧

dbIgaz = ∧

telepId = out értékei, ahol nem 0

* 1. Fejlesztői környezet

IBM PC, exe futtatására alkalmas operációs rendszer (pl. Windows 7). mingw32-g++.exe c++ fordítóprogram (v4.7), Code::Blocks (v13.12) fejlesztői környezet.

* 1. Forráskód

A teljes fejlesztői anyag –kicsomagolás után– a *meterológia* nevű könyvtárban található meg. A fej lesztés során használt könyvtár-struktúra:

|  |  |
| --- | --- |
| Állomány | Magyarázat |
| A1B2C3\bin\Debug\meterologia.exe | futtatható kód |
| A1B2C3\obj\Debug\main.o | félig lefordított kód |
|  |  |
| meterologia\main.cpp | C++ forráskód |
|  |  |
| meterologia\teszt1.txt | teszt-bemeneti fájl1 |
| meterologia\teszt2.txt | teszt-bemeneti fájl2 |
| meterologia\teszt3.txt | teszt-bemeneti fájl3 |
| meterologia\teszt4.txt | teszt-bemeneti fájl4 |
| meterologia\teszt5.txt | teszt-bemeneti fájl5 |
|  |  |
| meterologia\meterologia.docx | dokumentációk (ez a fájl) |

* 1. Megoldás
     1. Programparaméterek
        1. Változó

N, M, dailySum, start, startCheck : Egész

input : Szöveg  
out,avg, maxarr, minarr : Tömb(0..N-1)  
weather : Mátrix(0..N-1)(0..M-1)

* + 1. Programfelépítés

A program által használt modulok (és helyük):

main.cpp – program, a forráskönyvtárban

string – szövegkezeléses műveletek  
iostream – képernyő-, és billentyűkezelés, a C++ rendszer része  
cstdlib – általános rutinok, a C++ rendszer része

cmath – matematikai műveletek

* + 1. Függvénystruktúra

splitter

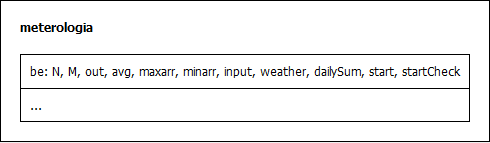
Főprogram

numberOfNotNull

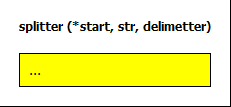
clearCin

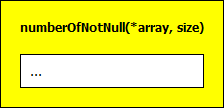
* + 1. A teljes program algoritmusa

Főprogram:



Alprogramok:

****

****

* + 1. A kód

A main.cpp fájl tartalma:

/\*

Készítette: Beregi Bence Zsolt

Neptun: DQK6TE

E-mail: beregi.bence@hotmail.com

Feladat: „ProgAlap beadandó feladatok” téma „Az átlag kisebb a minimum és maximum átlagánál” feladat

\*/

#include <iostream>

#include <cstdlib>

#include <string>

#include <cmath>

using namespace std;

double splitter(int\* start, string str, char delimetter)

{

\*start += str.find(delimetter) + 1;

str = str.substr(0, str.find(delimetter));

int output = atoi(str.c\_str());

return output;

}

int numberOfNotNull(int\* array, int size)

{

int output = 0;

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

if (array[i] != 0) output++;

return output;

}

void clearCin()

{

cin.clear();

cin.ignore(256, '\n');

}

int main()

{

int N, M;

cerr << "Please set number of settlements, and the number of days. (Like this: 'number of settlements' 'number of days'): " << endl;

cin >> N >> M;

while (N < 1 || N > 1000 || M < 1 || M > 1000 || cin.fail())

{

cin.fail() == true ? cerr << "Bad input. Check and try again: " : cerr << "One of the values was out of bound. Check and try again: ";

clearCin();

cin >> N >> M;

}

int out[N];

double weather[N][M];

double avg[N];

int maxarr[M], minarr[M];

string input;

getline(cin, input);

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

{

double dailySum = 0;

int start = 0;

int startCheck = -1;

cerr << "Give me the measured values of settlement " << i + 1 << ": ";

getline(cin, input);

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

{

int isIt = splitter(&start, input.substr(start), ' ');

if (startCheck == start && j != M - 1)

{

cerr << "Not enough value. Check and try again.\n";

i--;

break;

}

else if (j == M - 1 && startCheck != start)

{

cerr << "Too many values. Check and try again.\n";

i--;

break;

}

startCheck = start;

if (isIt >= -50 && isIt <= 50 && !(cin.fail()))

{

weather[i][j] = isIt;

dailySum += isIt;

}

else

{

cin.fail() == true ? cerr << "Bad input\n" : cerr << "One of the values was out of bound\n";

cin.clear();

i--;

break;

}

if (maxarr[i] < isIt || j == 0) maxarr[i] = isIt;

if (minarr[i] > isIt || j == 0) minarr[i] = isIt;

}

avg[i] = dailySum / M;

if (abs(avg[i] - maxarr[i]) > abs(avg[i] - minarr[i]))

out[i] = i + 1;

else

out[i] = 0;

}

cerr << "Result: ";

cout << numberOfNotNull(out, N);

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

if (out[i] != 0) cout << " " << out[i];

return 0;

}

* 1. Tesztelés
     1. Érvényes tesztesetek
        1. teszteset: teszt3.txt

|  |
| --- |
| Bemenet |
| N = 3, M = 3  Település1 = 10 10 15 (Szóközzel elválasztva hőmérséklet naponta)  Település2 = 10 10 10  Település3 = 15 12 10 |
| Kimenet |
| 2 1 3 |

* + - 1. teszteset: teszt1.txt

|  |
| --- |
| Bemenet |
| N = 2, M = 5  Település1 = 2 9 10 5 11  Település2 = 2 6 -5 36 2 |
| Kimenet |
| 1 2 |

* + - 1. teszteset: teszt2.txt

|  |
| --- |
| Bemenet |
| N = 5, M = 12 |
| Település1 = 6 9 8 12 45 32 -5 12 -10 25 41 11  Település2 = 2 6 8 49 -36 8 -4 -3 -5 45 21 36  Település3 = 1 5 9 -4 -49 -50 36 8 36 45 12 3  Település4 = 12 45 12 12 6 5 13 14 18 19 -50 1  Település5 = 32 4 -6 -9 45 12 13 14 14 14 20 26 |
| Kimenet |
| 2 1 5 |

* + - 1. teszteset: be4.txt

|  |
| --- |
| Bemenet |
| N = 3, M = 10 |
| Település1 = 11 11 11 11 12 15 14 19 23 50  Település2 = 12 12 12 12 -10 12 12 18 -40 5  Település3 = 15 15 15 15 15 15 15 15 15 15 10 |
| Kimenet |
| 1 1 |

* + - 1. teszteset: be5.txt

|  |
| --- |
| Bemenet – *csak kontinens van* |
| N = 10, M = 5 |
| Település1 = 12 15 7 36 -20  Település2 = 19 41 -40 5 36  Település3 = 12 36 41 -5 3  Település4 = 12 50 -50 1 4  Település5 = 21 20 1 2 6  Település6 = 21 40 36 -5 3  Település7 = 36 41 20 4 8  Település8 = 31 41 50 35 36  Település9 = -34 -50 -36 -40 -20  Település10 = -36 -20 -3 -4 -16 |
| Kimenet |
| 4 3 7 8 9 |

* + 1. Érvénytelen tesztesetek
       1. teszteset

|  |
| --- |
| Bemenet – *Rossz település szám* |
| N = 11tizenegy, M = 2 |
| Kimenet |
| Bad input. Check and try again: N = , M = |

* + - 1. teszteset

|  |
| --- |
| Bemenet – *Rossz hőmérséklet* |
| N = 3, M = 4  Település1 = 3 4 15 101 |
| Kimenet |
| One of the values was out of bound.  Település1 = |

* + - 1. 8. teszteset

|  |
| --- |
| Bemenet – *Túl sok hőmérséklet adat* |
| N = 3, M = 4  Település1 = 3 4 15 2 5 6 |
| Kimenet |
| Too many values. Check and try again.  Település1 = |

* + - 1. 9. teszteset

|  |
| --- |
| Bemenet – *Kevés hőmérséklet adat* |
| N = 3, M = 4  Település1 = 3 4 |
| Kimenet |
| Not enough value. Check and try again.  Település1 = |

* 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, és hőmérséklet különbségek grafikus megjelenítése.