Programozási alapismeretek komplex beadandó:
Sokáig meleg települések

Készítette: Hartyányi Kevin Attila

2017.november.30

Tartalom

Felhasználói dokumentáció

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 30 fok felett lesz!

Futási környezet:

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

Használat:

A program indítása

A program az C0S0RJ\bin\Debug\C0S0RJ.exe néven található a tömörített állományban. A C0S0RJ.exe fájl kiválasztásával indítható.

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,j	Az első napra jósolt M hőmérséglet értéke. (-50≤H1,j≤50)
4.	H2,j	Az második napra jósolt M hőmérséglet értéke. (-50≤H2,j≤50)
…	…	
N+2.	Hi,j	Az N-edik napra jósolt M hőmérséglet értéke. (-50≤Hi,j≤50)

A program kimenete

A standard kimenet első sorába azon települések T számát kell kiírni, ahol a hőmérséklet legalább egy héten át (7 alkalommal közvetlenül egymás után) 30 fok felett lesz! Ezt ezen települések sorszáma kövesse, növekvő sorrendben!

Minta bemenet és kimenet

Bemenet

Települések száma[1…1000]: 3
Napok száma[1…1000]: 8
1.nap hőmérsékletei[-50…50]: 16; 15; 12; 10; 9; 8; 3; -1
2.nap hőmérsékletei[-50…50]: 30; 30; 30; 30; 30; 30; 30; 30
3.nap hőmérsékletei[-50…50]: 32; 36; 37; 38; 31; 31; 32; 33

Kimenet

Azon települések száma, ahol a hőmérséklet legalább egy héten át 30 fok felett volt: 1
… ezen település sorszáma: 3

Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha a települések száma, vagy a napok száma nem egész szám, vagy nem esik az 1…1000 intervallumba; vagy valamely hőmérsékleti érték nem szám, vagy nem esik a -50…50 intervallumba. Hiba esetén a program azzal jelzi a hibát, hogy újra kérdezi azt. És egy hiba üzenetet jelenít meg.

Mintafutás hibás bemeneti adatok esetén:

Települések száma[1…1000]: 12alma
Hibas szamot adtal meg!
Települések száma[1…1000]: egy és ezer közötti szám
Hibas szamot adtal meg!
Települések száma[1…1000]: 0
Hibas szamot adtal meg!
Települések száma[1…1000]: 9
Napok száma[1…1000]: 5000
Hibas szamot adtal meg!
Napok száma[1…1000]: ötös dimat
Hibas szamot adtal meg!
Napok száma[1…1000]: 8
1.nap hőmérsékletei[-50…50]: 100; 15; 12; -80; 9; 8; 3; -1
Hibas szamot adtal meg!
1.nap hőmérsékletei[-50…50]: 16; 15; 12; 10; 9; 8; 3; -1
2.nap hőmérsékletei[-50…50]: 30; 30; 30; 30; 30; 30; 30; 30

Fejlesztői dokumentáció

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 30 fok felett lesz!

Specifikáció

Bemenet: $N,M \in N$
$Hőmérséklet \in Z^{NxM}$

Kimenet: $Db \in N$, $30felett \in N^{Db}$

Előfeltétel: $N \in [1…1000] \wedge M \in [1…1000] \wedge \forall i \in [1…N]: \forall j \in [1…M]: Hőmérséklet_{i,j} \in [-50…50]$

Utófeltétel: $Db = \sum_{\substack{i=1 \\ nagyobb30(Hőmérséklet\ i)}}^{N} 1 \wedge \forall i \in [1…Db]: nagyobb30(30felett_i)$

Definíció: nagyobb30: $N \rightarrow L$
$nagyobb30 := \exists j \in [1…(M-6)]: \forall d \in [j…(j+6)]: Hőmérséklet_{i,d} > 30$

Fejlesztői környezet

IBM PC, exe futtatására alkalmas operációs rendszer (pl. Windows 10). mingw32-g++.exe c++ fordítóprogram (v4.7), Code::Blocks (v16.01) fejlesztői környezet.

Forráskód

A teljes fejlesztői anyag –kicsomagolás után– a C0S0RJ nevű könyvtárban található meg. A fejlesztés során használt könyvtár-struktúra:

Állomány	Magyarázat
C0S0RJ\bin\Debug\C0S0RJ.exe	futtatható kód
C0S0RJ\obj\Debug\main.o	félig lefordított kód
C0S0RJ\main.cpp	C++ forráskód
C0S0RJ \teszt1.txt	teszt-bemeneti fájl$_1$
C0S0RJ \teszt2.txt	teszt-bemeneti fájl$_2$
C0S0RJ \teszt3.txt	teszt-bemeneti fájl$_3$
C0S0RJ \teszt4.txt	teszt-bemeneti fájl$_4$
C0S0RJ \C0S0RJ.docx	dokumentáció (ez a fájl)

Megoldás

Programparaméterek

Konstans

MaxN	: **Egész**(1000)	[a települések és napok maximális száma]

Típus

ki	= **Tömb**(1..MaxN:**Egész**)

Változó

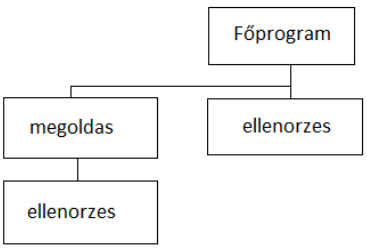
db	: **Egész**
telepulesek	: **Egész**
napok	: **Egész**
homersegletek	: **Egész**
hiba	: **logikai**

Programfelépítés

A program által használt modulok (és helyük):

main.cpp	– program, a forráskönyvtárban
iostream	– képernyő-, és billentyűkezelés, a C++ rendszer része
cstdlib	– ebben található az atoi függvény, a C++ rendszer része

Függvénystruktúra



A teljes program algoritmusa

Főprogram:

Main

Db:= 0, MAXN:= 1000
telepulesek:= ellenorzes(1, MAXN)
napok:= ellenorzes(1, MAXN)
Db:= megoldas(telepulesek, napok, ki)
Ki: Db
i:= 1...Db
Ki: ki[i]

Alprogramok:

Ellenorzes

Be: min, max		
hiba:= true		
szam		
hiba		
hiba:= false		
Be: seged		
i= 1...seged.length		
(seged[i] nem szám és seged[i] != '-') vagy (i > 0 és seged[i] = '-')		
hiba:= true		
hiba = false		
szam:= seged		
hiba:= szam < min vagy szam > max		
hiba		
Ki: "Hibas szamot adtal meg!"		
Ki: szam		

Megoldas

```
Be: telepulesek, napok
Be: ki
db:= 0
    i=1...telepulesek
        vanharmincfeletti:= false
        harmincfelett:= 0
            j=1...napok
                homersekletek:= ellenorzes(-50, 50)
                
                vanharmincfeletti = false és homersekletek > 30
                harmincfelett:= harmincfelett + 1 | harmincfelett:= 0
                harmincfelett = 7
                vanharmincfeletti = true |
        vanharmincfeletti
        ki[db]:= i |
        db:= db + 1 |
Ki: db
```

A kód

A main.cpp fájl tartalma:

```
/*
  Készítette: Hartyányi Kevin
  Neptun: C0S0RJ
  E-mail: hartyanyi.kevin@gmail.com
  Feladat: Sokáig meleg települések
*/

#include <iostream>
//#include <string>
#include <cstdlib>

using namespace std;

int ellenorzes(int min, int max) //Bemenet ellenorzes
{
	bool hiba = false;
	int szam;

	do
	{
		hiba = false;
		string seged;
		cin >> seged;

		for (int i = 0; i < seged.length(); ++i)
		{
			if ((!isdigit(seged[i]) && seged[i] != '-') || (i > 0 && seged[i] == '-'))
//Hogy '-' csak az elso karakter lehesen
			{
				hiba = true;
			}
		}

		if (!hiba)
		{
			szam = atoi(seged.c_str());
			hiba = szam < min || szam > max;
		}
```

```
		if (hiba)
		{
			cout << "Hibas szamot adtal meg!" << endl;
			cin.clear();
			cin.ignore(1024, '\n');
		}
	} while (hiba);

	return szam;
}
int megoldas(int telepulesek, int napok, int ki[])
{
	int db = 0;
	int homersegletek;

	for (int i = 0; i < telepulesek; ++i)
	{
		bool vanharmincfeletti = false;
		int harmincfelett = 0;

		for (int j = 0; j < napok; ++j)
		{
			homersegletek = ellenorzes(-50, 50);

			if (vanharmincfeletti == false && homersegletek > 30)
			{
				++harmincfelett;

				if (harmincfelett == 7)
				{
					vanharmincfeletti = true;
				}
			}
			else
			{
				harmincfelett = 0;
			}

		}
		if (vanharmincfeletti)
		{
			ki[db] = i + 1;
			++db;
		}
	}

	return db;
```

```
}

int main()
{
   ios::sync_with_stdio(false);
        const int MAXN = 1000;
        int ki[MAXN];
        int db = 0;
        int telepulesek;
        int napok;
        //Adatok bekerese es feladat elvegzese
        telepulesek = ellenorzes(1, 1000);
        napok = ellenorzes(1, 1000);

        db = megoldas(telepulesek, napok, ki);

        //Kimenet
        cout << db << " ";
        for (int i = 0; i < db; ++i)
        {
                cout << ki[i] << " ";
        }
        cout << endl;

        return 0;
}
```

Tesztelés

Érvényes tesztesetek

1. teszteset: be1.txt

Bemenet – minimális hossz
N = 3; M = 8
Hőmérsékletek$_1$ = 16 15 12 10 9 8 3 -1
Hőmérsékletek$_2$ = 30 30 30 30 30 30 30 30
Hőmérsékletek$_3$ = 32 36 37 38 31 31 32 33
Kimenet
1 3

2. teszteset: be2.txt

Bemenet – maximális hossz
N = 1000; M = 1000 Hőmérsékletek$_1$ = 43 41 46 46 50 50 49 46 48 47 47 … … … …
Kimenet
917 1 2 3 4 5 6 7 8 9 10 12 13 14 15 16 18 20 21 22 23 24 25 26 27 28 30 31 32 34 35 36 37 38 39 40 41 42 43 …

3. teszteset: be3.txt

Bemenet – 7-nél kevesebb nap
N = 1; M = 5 Hőmérsékletek$_1$ = 35 35 35 35 35
Kimenet
0

4. teszteset: be4.txt

Bemenet – sehol sincs 7 nap egymás után, ahol a hőmérséklet legalább 30 fok
N = 2; M = 7
Hőmérsékletek$_1$ = 12 41 -4 5 -1 6 0 Hőmérsékletek$_2$ = 34 -31 0 4 2 5 1
Kimenet
0

Érvénytelen tesztesetek

5. teszteset

Bemenet – Rossz település szám
N = 12alma
Kimenet
Újrakérdezés: Hibas szamot adtal meg! N =

6. teszteset

Bemenet – Rossz hőmérséklet
N = 4; M = 10 Hőmérsékletek$_1$ = 0 20 4 -100 9 8 3 -1
Kimenet
Újrakérdezés: Hibas szamot adtal meg! Hőmérsékletek$_1$ =

Fejlesztési lehetőségek:

1. Hibák fajtáinak felismerése és a hibának megfelelő hibakód kiírása esetleg a hiba helyének feltüntetése.
2. A kimenet egy fájlba való elmentése.
3. Felhasználói igényeknek megfelelően ajánlani, hogy mikor hova érdemes túrázni menni.
4. Az adatok felhasználásával különböző statisztikák kiszámítása.
5. Automatikus futtatás megszervezése.
6. Adatok fájlból fogadása (felhasználó igényei szerint).
7. A korlátozások szűkítése (pl.: 1000 nagyobb szám is megadható legyen napnak és településnek).