„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*

2019. december 20.

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
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. Készíts programot, amely 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 7). Nem igényel egeret.

* 1. Használat
     1. A program indítása

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

* + 1. A program bemenete

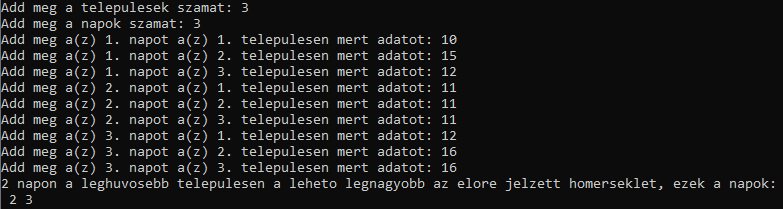
A program az adatokat a **billentyűzet**ről olvassa be a következő sorrendben:

|  |  |  |
| --- | --- | --- |
| # | Adat | Magyarázat |
| **1.** | *N* | A települések száma (1≤N≤1000) |
| **2.** | *M* | A napok száma (1≤M≤1000) |
| **3.** | *Hőmérsékletek11* | 1. sorban az egyes napokra jósolt 1. hőmérséklet értéke található (-50≤Hi,j≤50). |
| **4.** | *Hőmérsékletek12* | 1. sorban az egyes napokra jósolt 2. db hőmérséklet értéke (-50≤Hi,j≤50). |
| **…** | *…* |  |
| **NxM.** | *HőmérsékletekNM* | N. sorban az egyes napokra jósolt M. hőmérséklet értéke (-50≤Hi,j≤50). |

* + 1. A program kimenete

A első sorába azon napok K számát, amelyeken a leghűvösebb településen a lehető legnagyobb az előre jelzett hőmérséklet. Ezt ezen napok sorszámai követik, növekvő sorrendben.

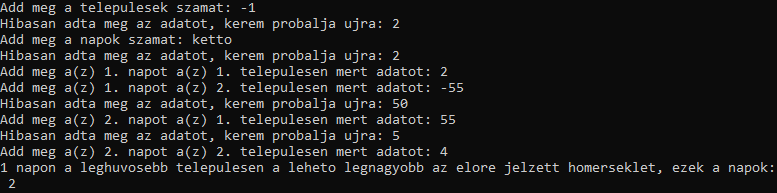
* + 1. Minta bemenet és kimenet



* + 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 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. Egy repülőút során egyenlő távolságonként mértük a felszín tengerszint feletti magasságát. Zéró magasságot ott mértünk, ahol állóvíz volt, pozitív magasságot pedig ott, ahol szárazföld. Három egymást követő mérési eredményt jelöljön A, B és C. Ekkor B

* bal oldali partot jelez, ha B>0 és A=0;
* jobb oldali partot jelez, ha B>0 és C=0.

Készítsen programot, amely meghatároz két szigetet, melyeknél nincs egymáshoz közelebbi sziget­pár, ha nincs ilyen, akkor ezt egyetlen 0-val jelezze!

* 1. Specifikáció

**Bemenet**: N∈N, Magasságok∈N\*

**Kimenet**: VanE∈L, Szig1,Szig2∈Sziget, Sziget=Bal×Jobb, Bal,Jobb=N

**Előfeltétel**: N=Hossz(Magasságok) ∧ N∈[2..10000] ∧ ∀i∈[1..N]: Magasságoki∈[0..9000]

**Utófeltétel**: db= ∧   
szigek∈Szigetdb ∧   
∀i∈[1..db]: (szigeki.Bal∈[2..N-1] ∧ szigeki.Jobb∈[2..N-1] ∧   
 szigeki.Bal≤szigeki.Jobb ∧ SzigetKezdet(szigeki.Bal) ∧ SzigetVég(szigeki.Jobb) ∧  
 ∀j∈[szigeki.Bal..szigeki.Jobb]: Magasságokj>0) ∧  
db<2 → VanE=Hamis ∧  
db≥2 → VanE=Igaz ∧   
 ∃i∈[1..db-1]: Szig1=szigeki ∧ Szig2=szigeki+1 ∧  
 ∀i∈[1..db-1]: szigeki+1.Bal-szigeki.Jobb≥Szig2.Bal-Szig1.Jobb

**Definíció**: SzigetKezdet: N→L  
SzigetKezdet(i):=Magasságoki>0 ∧ Magasságoki-1=0  
SzigetVég: N→L  
SzigetVég(i):=Magasságoki>0 ∧ Magasságoki+1=0

**Megjegyzés**: a „ha nincs ilyen” kitételt (a VanE=Hamis esetben) a program egyetlen 0 kiírásával fog­ja jelezni, nem pedig a logikai érték megjelenítésével (hűen a feladat eredeti kiírásához).

* 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– az A1B2C3 nevű könyvtárban található meg. A fej­lesztés során használt könyvtár-struktúra:

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

* 1. Megoldás
     1. Programparaméterek
        1. Konstans

1. MaxN : **Egész**(10000) [a mérések maximális száma]  
   MaxMagasság : **Egész**(9000) [a maximális magasság]
   * + 1. Típus
2. TMaggasságok = **Tömb**(1..MaxN:**Egész**)  
   TSziget = **Rekord**(bal,jobb:**Egész**)
   * + 1. Változó

N : **Egész**  
Magasságok : TMagasságok  
Szig1,Szig2 : TSziget

* + 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

* + 1. Függvénystruktúra

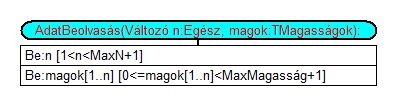


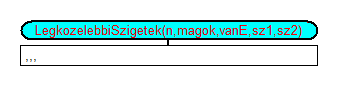
* + 1. A teljes program algoritmusa

Főprogram:

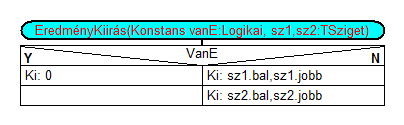


Alprogramok:





**Változó**  
 db:**Egész**  
 …



* + 1. A kód

A main.cpp fájl tartalma:

/\*

Készítette: Gipsz Jakab

Neptun: A1B2C3

E-mail: gipszjakab@vilaghalo.hu

Feladat: „ProgAlap beadandó feladatok” téma „Legközelebbi szigetek” feladat

\*/

#include <iostream>

#include <stdlib.h>

**using** **namespace** std**;**

const string Cim**=**"Legközelebbi szigetek"**;**

const int MaxN**=**10000**;**

const int MaxMagassag**=**9000**;**

**typedef** int TMagassagok**[**MaxN**];**

**typedef** struct**{**int bal**,**jobb**;}** TSziget**;**

//Bemenet:

int N**;**

TMagassagok Magassagok**;**

//Kimenet:

bool VanE**;**

TSziget Szig1**,**Szig2**;**

void AdatBeolvasas**(**int**&** n**,** TMagassagok magok**);**

void LegkozelebbiSzigetek**(**int n**,** const TMagassagok magok**,**

bool**&** vanE**,** TSziget**&** sz1**,** TSziget**&** sz2**);**

void EredmenyKiir**(**bool VanE**,** TSziget sz1**,** TSziget sz2**);**

void BillreVar**();**

int main**()**

**{**

clog **<<** Cim **<<** endl **<<** endl**;**

AdatBeolvasas**(**N**,**Magassagok**);**

LegkozelebbiSzigetek**(**N**,**Magassagok**,**VanE**,**Szig1**,**Szig2**);**

EredmenyKiir**(**VanE**,**Szig1**,**Szig2**);**

BillreVar**();**//ez a biro-s változatban kommentálva szerepel!!!

**return** 0**;**

**}**

void AdatBeolvasas**(**int**&** n**,** TMagassagok magok**)**

**{**

**do{**  
 clog **<<** "…"**;** cin **>>** **…;**

…  
 **}While** **(…);** …  
**}**

void LegkozelebbiSzigetek**(**int n**,** const TMagassagok magok**,**

bool**&** vanE**,** TSziget**&** sz1**,** TSziget**&** sz2**)**

**{**

…  
**}**

void EredmenyKiir**(**bool VanE**,** TSziget sz1**,** TSziget sz2**)**

**{**

clog **<<** "…"**;** cout **<<** **…;**

…  
**}**

void BillreVar**()**

**{**

…  
**}**

* 1. Tesztelés
     1. Érvényes tesztesetek
        1. teszteset: be1.txt

|  |
| --- |
| Bemenet – *nincs* *sziget; minimális hossz* |
| N = 2  Magasság1 = 0  Magasság2 = 0 |
| Kimenet |
| 0 |

* + - 1. teszteset: be2.txt

|  |
| --- |
| Bemenet – *kontinenssel kezdődik, van legalább 2 sziget* |
| N = 12  Magasság1 = 3  Magasság2 = 0  Magasság3 = 2  Magasság4 = 0  Magasság5 = 4  Magasság6 = 3  Magasság7 = 0  Magasság8 = 0  Magasság9 = 3  Magasság10 = 0  Magasság11 = 2  Magasság12 = 0 |
| Kimenet |
| Szig1 = 3 3  Szig1 = 5 6 |

* + - 1. teszteset: be3.txt

|  |
| --- |
| Bemenet – *kontinenssel végződik, van legalább 2 sziget* |
| N = … |
| Magasság1 = …  … |
| Kimenet |
| … |

* + - 1. teszteset: be4.txt

|  |
| --- |
| Bemenet – *nincs kontinens, egy sziget van* |
| N = … |
| Magasság1 = …  … |
| Kimenet |
| … |

* + - 1. teszteset: be5.txt

|  |
| --- |
| Bemenet – *csak kontinens van* |
| N = … |
| Magasság1 = …  … |
| Kimenet |
| … |

* + 1. Érvénytelen tesztesetek
       1. teszteset

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

* + - 1. teszteset

|  |
| --- |
| Bemenet – *Rossz magasság* |
| N = 11  Magasság1 = -1 |
| Kimenet |
| Újrakérdezés:  Magasság1 = |

…

* + - 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.