

**„Programozási alapismeretek”
beadandó feladat:
„ProgAlap beadandó feladatok” téma 99. feladat ¹**

*Készítette: Gipsz Jakab
Neptun-azonosító: A1B2C3
E-mail: gipszjakab@vilaghalo.hu*

*Kurzuskód: **IP-o8PAED**
Gyakorlatvezető neve: ???*

2015. november 24.

¹ A fenti feladatsor itt érhető el: <http://biro.inf.elte.hu/> .
Értelemszerűen töltendő ki itt ki a szerzőre vonatkozó adatok.
A lábjegyzetek a végső dokumentációból törölendő! Csak az Ön segítségét szolgálja.

Tartalom

Felhasználói dokumentáció.....	3
Feladat.....	3
Futási környezet	3
Használat.....	3
A program indítása	3
A program bemenete	3
A program kimenete.....	3
Minta bemenet és kimenet	4
Hibalehetőségek	4
Fejlesztői dokumentáció	5
Feladat.....	5
Specifikáció.....	5
Fejlesztői környezet	5
Forráskód	6
Megoldás.....	6
Programparaméterek	6
Programfelépítés	6
Függvénystruktúra	6
Algoritmus	7
A kód	7
Tesztelés	9
Érvényes tesztesetek	9
Érvénytelen tesztesetek	10
Fejlesztési lehetőségek.....	10

Felhasználói dokumentáció

Feladat

Egy repülőút során egyenlő távolságokként 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 sziget-pár, ha nincs ilyen, akkor ezt egyetlen 0-val jelezze!

Futási környezet

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

Használat

A program indítása

A program az `A1B2C3\bin\Release\A1B2C3.exe` néven található a tömörített állományban. A `A1B2C3.exe` fájl kiválasztásával indítható.

A program bemenete

A program az adatokat a **billentyűzetről** olvassa be a következő sorrendben:

#	Adat	Magyarázat
1.	N	A magasságmérés hossza ($2 \leq N \leq 10000$).
2.	$Magasság_1$	Az első magasság ($0 \leq Magasság_1 \leq 9000$).
3.	$Magasság_2$	A második magasság ($0 \leq Magasság_2 \leq 9000$).
...	...	
N+1.	$Magasság_N$	Az N-edik magasság ($0 \leq Magasság_N \leq 9000$).

A program kimenete

A program kiírja az egymáshoz legközelebbi két sziget bal és jobb partját jelző mérés sorszámát. A kimenet első sorába az első szigetet, a második sorba a második szigetet azonosító két adat kerül. Ha nem lenne legalább két sziget, akkor egyetlen 0 a kimenet.

Minta bemenet és kimenet

```
Legközelebbi szigetek
Mérések száma [2..10000]:12
 1. mérés [0..9000]:3
 2. mérés [0..9000]:0
 3. mérés [0..9000]:2
 4. mérés [0..9000]:0
 5. mérés [0..9000]:4
 6. mérés [0..9000]:3
 7. mérés [0..9000]:0
 8. mérés [0..9000]:0
 9. mérés [0..9000]:3
10. mérés [0..9000]:0
11. mérés [0..9000]:2
12. mérés [0..9000]:0
A legközelebbi szigetpár elsője: 3 3
... párja: 5 6
```

Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha a mérések száma nem egész szám, vagy nem esik a 2..10 000 intervallumba; vagy valamely magassági érték nem szám, vagy nem esik a 0..9 000 intervallumba. Hiba esetén a program azzal jelzi a hibát, hogy újra kérdezi azt.

Minta futás hibás bemeneti adatok esetén:

```
Legközelebbi szigetek
Mérések száma [2..10000]:sok
Mérések száma [2..10000]:1
Mérések száma [2..10000]:1.1
Mérések száma [2..10000]:2
 1. mérés [0..9000]:kevés
 1. mérés [0..9000]:-1
 1. mérés [0..9000]:9001
 1. mérés [0..9000]:0
 2. mérés [0..9000]:
```

Fejlesztői dokumentáció

Feladat

Egy repülőút során egyenlő távolságoként 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 sziget-pár, ha nincs ilyen, akkor ezt egyetlen 0-val jelezze!

Specifikáció

Bemenet: $N \in \mathbb{N}$, Magasságok $\in \mathbb{N}^*$

Kimenet: $\text{VanE} \in \{L, 0\}$, $\text{Szig1}, \text{Szig2} \in \text{Sziget}$, $\text{Sziget} = \text{Bal} \times \text{Jobb}$, $\text{Bal}, \text{Jobb} \in \mathbb{N}$

Előfeltétel: $N = \text{Hossz}(\text{Magasságok}) \wedge N \in [2..10000] \wedge \forall i \in [1..N]: \text{Magasságok}_i \in [0..9000]$

Utófeltétel: $\text{db} = \sum_{i=2}^{N-1} 1 \wedge$
 $\text{SzigetKezdet}(i)$
 $\text{szigek} \in \text{Sziget}^{\text{db}} \wedge$
 $\forall i \in [1..\text{db}]: (\text{szigek}_i.\text{Bal} \in [2..N-1] \wedge \text{szigek}_i.\text{Jobb} \in [2..N-1] \wedge$
 $\text{szigek}_i.\text{Bal} \leq \text{szigek}_i.\text{Jobb} \wedge \text{SzigetKezdet}(\text{szigek}_i.\text{Bal}) \wedge \text{SzigetVég}(\text{szigek}_i.\text{Jobb}) \wedge$
 $\forall j \in [\text{szigek}_i.\text{Bal}..\text{szigek}_i.\text{Jobb}]: \text{Magasságok}_j > 0) \wedge$
 $\text{db} < 2 \rightarrow \text{VanE} = \text{Hamis} \wedge$
 $\text{db} \geq 2 \rightarrow \text{VanE} = \text{Igaz} \wedge$
 $\exists i \in [1..\text{db}-1]: \text{Szig1} = \text{szigek}_i \wedge \text{Szig2} = \text{szigek}_{i+1} \wedge$
 $\forall i \in [1..\text{db}-1]: \text{szigek}_{i+1}.\text{Bal} - \text{szigek}_i.\text{Jobb} \geq \text{Szig2}.\text{Bal} - \text{Szig1}.\text{Jobb}$

Definíció: $\text{SzigetKezdet}: \mathbb{N} \rightarrow \{L, 0\}$
 $\text{SzigetKezdet}(i) := \text{Magasságok}_i > 0 \wedge \text{Magasságok}_{i-1} = 0$
 $\text{SzigetVég}: \mathbb{N} \rightarrow \{L, 0\}$
 $\text{SzigetVég}(i) := \text{Magasságok}_i > 0 \wedge \text{Magasságok}_{i+1} = 0$

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

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.

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	félíg lefordított kód
A1B2C3\main.cpp	C++ forráskód
A1B2C3\teszt1.txt	teszt-bemeneti fájl ₁
A1B2C3\teszt2.txt	teszt-bemeneti fájl ₂
A1B2C3\teszt3.txt	teszt-bemeneti fájl ₃
A1B2C3\teszt4.txt	teszt-bemeneti fájl ₄
A1B2C3\teszt5.txt	teszt-bemeneti fájl ₅
A1B2C3\doksi\A1B2C3.docx	dokumentációk (ez a fájl)

Megoldás

Programparaméterek

Konstans

MaxN : **Egész** (10000) [a mérések maximális száma]
MaxMagasság : **Egész** (9000) [a maximális magasság]

Típus

TMagasságok = **Tömb** (1..MaxN:**Egész**)
TSziget = **Rekord** (bal, jobb:**Egész**)

Változó

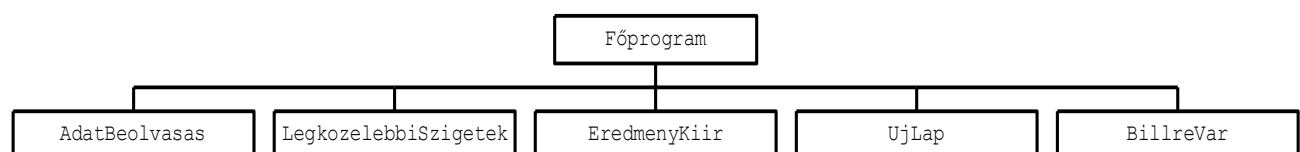
N : **Egész**
Magasságok : TMagasságok
Szig1, Szig2 : TSziget

Programfelépítés

A program által használt modulok (és helyük):

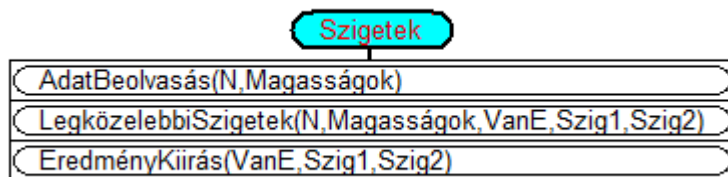
main.cpp – program, a forráskönyvtárban
iostream – képernyő-, és billentyűkezelés, a C++ rendszer része
stdlib.h – általános rutinok, a C++ rendszer része

Függvénystruktúra

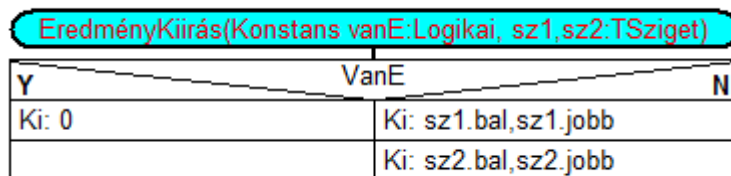
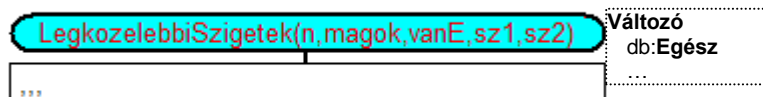
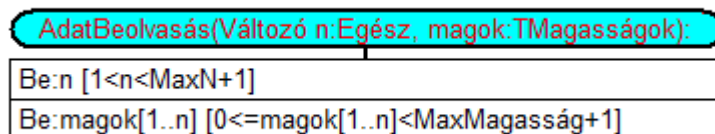


A teljes program algoritmusa

Főprogram:



Alprogramok:



A kód

A main.cpp fájl tartalma:

```
/*
Készítette: Gipsz Jakab
Neptun: A1B2C3
E-mail: gipszjakab@vilaghalo.hu
Feladat: „ProgAlap beadandó feladatok” téma 99. 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()
{
    cout << Cim << endl << endl;
    AdatBeolvasas(N,Magassagok);
    LegkozelebbiSzigetek(N,Magassagok,VanE,Szig1,Szig2);
    EredmenyKiir(VanE,Szig1,Szig2);
    BillreVar();
    return 0;
}

void AdatBeolvasas(int& n, TMagassagok magok)
{
    do{
        cerr << "..."; cin >> ...;
        ...
    }While (...);
    ...
}

void LegkozelebbiSzigetek(int n, const TMagassagok magok,
                           bool& vanE, TSziget& sz1, TSziget& sz2)
{
    ...
}

void EredmenyKiir(bool VanE, TSziget sz1, TSziget sz2)
{
    cerr << "..."; cout << ...;
    ...
}

void BillreVar()
{
    ...
}

```


Tesztelés

Érvényes tesztesetek

1. *teszteset: be1.txt*

Bemenet – nincs sziget; minimális hossz
N = 2 Magasság ₁ = 0 Magasság ₂ = 0
Kimenet
0

2. *teszteset: be2.txt*

Bemenet – kontinenssel kezdődik, van legalább 2 sziget
N = 12 Magasság ₁ = 3 Magasság ₂ = 0 Magasság ₃ = 2 Magasság ₄ = 0 Magasság ₅ = 4 Magasság ₆ = 3 Magasság ₇ = 0 Magasság ₈ = 0 Magasság ₉ = 3 Magasság ₁₀ = 0 Magasság ₁₁ = 2 Magasság ₁₂ = 0
Kimenet
Szig1 = 3 3 Szig1 = 5 6

3. *teszteset: be3.txt*

Bemenet – kontinenssel végződik, van legalább 2 sziget
N = ... Magasság ₁ =
Kimenet
...

4. *teszteset: be4.txt*

Bemenet – nincs kontinens, egy sziget van
N = ...
Magasság ₁ = ...
...
Kimenet
...

5. *teszteset: be5.txt*

Bemenet – csak kontinens van
N = ...
Magasság ₁ = ...
...
Kimenet
...

Érvénytelen tesztesetek

6. *teszteset*

Bemenet – Rossz hossz
N = 11tizenegy
Kimenet
Újrakérdezés: N =

7. *teszteset*

Bemenet – Rossz magasság
N = 11
Magasság ₁ = -1
Kimenet
Újrakérdezés: Magasság ₁ =
...

8. *teszteset*

...

Fejlesztési lehetőségek

1. Adatok –a felhasználó igénye szerint– akár fájlból is fogadása.
2. Hibás fájl-bemenetek felismerése, és a hiba helyének (sor sorszámanak) kiírása.
3. Többszöri futtatás megszervezése
4. A bemeneti sorozat grafikus megjelenítése, s az eredmény-szigetek elütő színű kijelzése.