**Hibalehetőségek**

A versenyzők, illetve a versenyek száma nem **1** és **100** közé esik. A minimum pontszámok nem **0** és **50** közé esnek. A versenyenkénti indulók száma, illetve a versenyzők sorszáma nem **1** és **induló versenyzők száma** közé esnek. Végül az elért pontszám nem **1** és **100** közé esik, vagy ha bármelyik bekéréskor karaktereket, vagy karakterláncokat adunk meg. Hiba esetén a program kiírja a hibát, majd utána lehetőséget ad a javításra, ismét bekéri az adatot.

**Fejlesztői dokumentáció**

**Feladat**

Egy iskolában egyéni és összetett tanulmányi versenyt tartottak. A versenyekben összesen N tanuló vett részt. A versenyek száma M. Ismerjük versenyenként az induló tanulókat és elért pontszámukat. Az összetett versenyben csak azon tanulók eredményét értékelik, akik az összes egyéni versenyen indultak és elérték a versenyenként adott minimális pontszámot.  
Készíts programot, amely megadja azon versenyen induló tanulókat, akik a minimális pontszámot mindenhol elérték, ahol indultak!

**Specifikáció**

Bemenet: N ∈ N; M ∈ N;Min ∈ NM ;Ind ∈ NM; P ∈ NN,M; S ∈ NN,M

Kimenet: T ∈ N; tomb ∈ NT

Előfeltétel: N ∈ [1…100] ⋀ M ∈ [1..100] ⋀ ∀i : (Mini ∈ [0...50] ⋀ Indi ∈ [1..N] ⋀ ∀j : (Si,j ∈ [1…N] ⋀ Pi.j ∈ [1…100]))

Utófeltétel:

**Fejlesztői környezet**

IBM PC, exe futtatására alkalmas operációs rendszer (pl. Windows 98). mingw32-g++.exe c++ fordítóprogram (v4.7), Code::Blocks (v16.01) fejlesztői környezet.

**Forráskód**

A teljes fejlesztői anyag –kicsomagolás után– az B1V655 nevű könyvtárban található meg. A fej­lesztés során használt könyvtár-struktúra:

|  |  |
| --- | --- |
| Állomány | Magyarázat |
| B1V655\bin\Release\B1V655.exe | futtatható kód |
| B1V655\obj\Release\main.o | félig lefordított kód |
| B1V655\main.cpp | C++ forráskód |
| B1V655\teszt1.txt | teszt-bemeneti fájl1 |
| B1V655\teszt2.txt | teszt-bemeneti fájl2 |
| B1V655\teszt3.txt | teszt-bemeneti fájl3 |
| B1V655\teszt4.txt | teszt-bemeneti fájl4 |
| B1V655\dokumentacio\B1V655.doc | dokumentáció |

**Megoldás**

***Programparaméterek***

***Típus***

S : Tömb(0…99;0…99:Egész)

P : Tömb(0…99;0…99:Egész)

Min : Tömb(0…99:Egész)

Ind : Tömb(0…99:Egész)

tomb : Tömb(0…99:Egész)

***Változó***

N : Egész

M : Egész

T : Egész

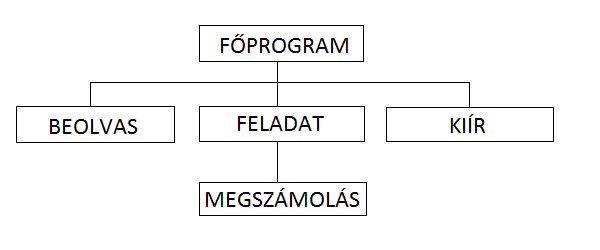
hiba : Logikai

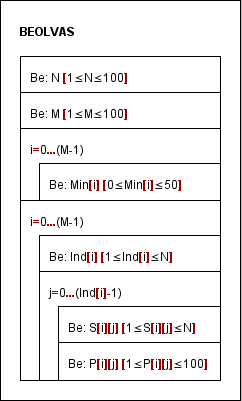
CIN : Karakterlánc

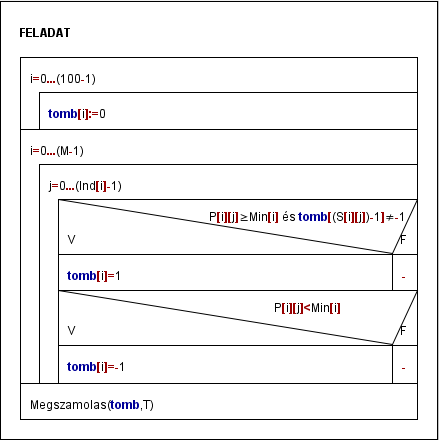
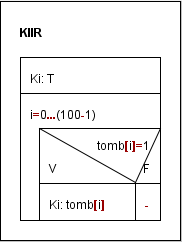
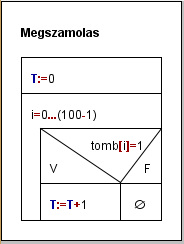
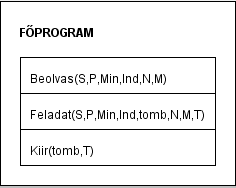
***Programfelépítés***

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***



**A kód**

//Név: Horváth Milán

//Neptunkód: B1V655

//MAIL: hormilag@gmail.com

//Feladat: Minimális pont alatt nem voltak

/\*Egy iskolában egyéni és összetett tanulmányi versenyt tartottak. A versenyekben összesen N tanuló vett részt. A versenyek száma M. Ismerjük versenyenként az induló tanulókat és elért pontszámukat.Az összetett versenyben csak azon tanulók eredményét értékelik, akik az összes egyéni versenyen indultak és elérték a versenyenként adott minimális pontszámot.

Készíts programot, amely megadja azon versenyen induló tanulókat, akik a minimális pontszámot mindenhol elérték, ahol indultak!\*/

#include <iostream>

#include <stdlib.h>

using namespace std;

**void** beolvas(**int** S[100][100],**int** P[100][100],**int** Min[100],**int** Ind[100],**int** &N,**int** &M);

**void** feladat(**const int** S[100][100],**const int** P[100][100],**const int** Min[100],**const int** Ind[100],**int** tomb[100], **int** N, **int** M, **int** &T);

**void** megszamolas(**const int** tomb[100],**int** &T);

**void** kiir(**const int** tomb[100],**int** T);

**int** main()

{

**int** S[100][100],P[100][100];

**int** Min[100];

**int** N,M;

**int** Ind[100];

**int** tomb[100];

**int** T=0;

beolvas(S,P,Min,Ind,N,M);

feladat(S,P,Min,Ind,tomb,N,M,T);

kiir(tomb,T);

system("pause");

**return** 0;

}

**void** beolvas(**int** S[100][100],**int** P[100][100],**int** Min[100],**int** Ind[100], **int** &N, **int** &M)

{

string CIN;

**bool** hiba;

**do**{

cerr<<"Adja meg az indulok szamat: ";

getline(cin,CIN);

N=atoi(CIN.c\_str());

hiba=(N<1||N>100);

**if**(hiba)cerr<<"A beirt szam nem 1 es 100 kozott van!"<<endl;

}**while**(hiba);

**do**{

cerr<<"Adja meg a versenyek szamat: ";

getline(cin,CIN);

M=atoi(CIN.c\_str());

hiba=(M<1||M>100);

**if**(hiba)cerr<<"A beirt szam nem 1 es 100 kozott van!"<<endl;

}**while**(hiba);

**for**(**int** i=0;i<M;i++){

**do**{

cerr<<"Elvart pontszam az "<<i+1<<". versenyen: ";

getline(cin,CIN);

Min[i]=atoi(CIN.c\_str());

hiba=(Min[i]<0||Min[i]>50||(Min[i]==0&&CIN!="0"));

**if**(hiba)cerr<<"A beirt szam nem 0 es 50 kozott van!"<<endl;

}**while**(hiba);

}

**for**(**int** i=0;i<M;i++){

**do**{

cerr<<"Adja meg a(z) "<<i+1<<". versenyen indulok szamat: ";

getline(cin,CIN);

Ind[i]=atoi(CIN.c\_str());

hiba=(Ind[i]<1||Ind[i]>N);

**if**(hiba)cerr<<"A beirt szam nem 1 es "<<N<<" kozott van!"<<endl;

}**while**(hiba);

**for**(**int** j=0;j<Ind[i];j++){

**do**{

cerr<<"Adja meg az indulo versenyzo sorszamat: ";

getline(cin,CIN);

S[i][j]=atoi(CIN.c\_str());

hiba=(S[i][j]<1||S[i][j]>N);

**if**(hiba)cerr<<"A beirt szam nem 1 es "<<N<<" kozott van!"<<endl;

}**while**(hiba);

**do**{

cerr<<"Adja meg az indulo versenyzo elert pontszamat: ";

getline(cin,CIN);

P[i][j]=atoi(CIN.c\_str());

hiba=(P[i][j]<1||P[i][j]>100);

**if**(hiba)cerr<<"A beirt szam nem 1 es 100 kozott van!"<<endl;

}**while**(hiba);

}

}

}

**void** feladat(**const int** S[100][100],**const int** P[100][100],**const int** Min[100],**const int** Ind[100],**int** tomb[100], **int** N, **int** M,**int** &T)

{

**for**(**int** i=0;i<100;i++)tomb[i]=0;

**for**(**int** i=0;i<M;i++){

**for**(**int** j=0;j<Ind[i];j++){

**if**(P[i][j]>=Min[i]&&tomb[(S[i][j])-1]!=-1)tomb[(S[i][j])-1]=1;

**if**(P[i][j]<Min[i])tomb[(S[i][j])-1]=-1;

}

}

megszamolas(tomb,T);

}

**void** megszamolas(**const int** tomb[100],**int** &T)

{

**for**(**int** i=0;i<100;i++)**if**(tomb[i]==1)T++;

}

**void** kiir(**const int** tomb[100],**int** T)

{

cerr<<"A versenyeken minimalis pontszamot elert tanulok szama, es azok sorszama novekvo sorrendben: "<<endl;

cout<<T;

**for**(**int** i=0;i<100;i++)**if**(tomb[i]==1)cout<<" "<<i+1;

}

**Tesztelés**

***Érvényes testesetek***

1. ***teszteset: teszt1.txt***

|  |
| --- |
| Bemenet – |
| N = 3  M = 3  Max0 =10  Max1 = 5  Max2 = 10  Ind0 = 3  S0,0 = 1  P0,0 = 20  S0,1 = 2  P0,1 = 10  S0,2 = 3  P0,2 = 10  Ind1 = 3  S1,0 = 1  P1,0 = 8  S1,1 = 2  P1,1 = 9  S1,2 = 3  P1,2 =10  Ind2 =3  S1,0 = 1  P1,0 = 20  S1,1 = 2  P1,1 = 30  S1,2 = 3  P1,2 = 40 |
| Kimenet |
| 3 1 2 3 |

1. ***teszteset: teszt2.txt***

|  |
| --- |
| Bemenet – csak 1 indul |
| N = 3  M = 3  Max0 =10  Max1 = 10  Max2 = 10  Ind0 = 1  S0,0 = 1  P0,0 = 20  Ind1 = 1  S1,0 = 1  P1,0 = 20  Ind2 =1  S1,0 = 1  P1,0 = 20 |
| Kimenet |
| 1 1 |

1. ***teszteset: teszt3.txt***

|  |
| --- |
| Bemenet – minimum pontszám |
| N = 3  M = 3  Max0 =0  Max1 = 0  Max2 = 0  Ind0 = 3  S0,0 = 1  P0,0 = 1  S0,1 = 2  P0,1 = 1  S0,2 = 3  P0,2 = 1  Ind1 = 3  S1,0 = 1  P1,0 = 1  S1,1 = 2  P1,1 = 1  S1,2 = 3  P1,2 =1  Ind2 =3  S1,0 = 1  P1,0 = 1  S1,1 = 2  P1,1 = 1  S1,2 = 3  P1,2 = 1 |
| Kimenet |
| 3 1 2 3 |

1. ***teszteset: teszt4.txt***

|  |
| --- |
| Bemenet – nem érte el senki |
| N = 3  M = 3  Max0 = 50  Max1 = 50  Max2 = 50  Ind0 = 3  S0,0 = 1  P0,0 = 10  S0,1 = 2  P0,1 = 20  S0,2 = 3  P0,2 = 30  Ind1 = 3  S1,0 = 2  P1,0 = 10  S1,1 = 3  P1,1 = 20  S1,2 = 1  P1,2 =30  Ind2 =3  S1,0 = 3  P1,0 = 10  S1,1 = 1  P1,1 = 20  S1,2 = 2  P1,2 = 30 |
| Kimenet |
| 3 1 2 3 |

***Érvénytelen tesztesetek***

1. ***teszteset***

|  |
| --- |
| Bemenet – rossz N |
| N = 11tizenegy |
| Kimenet |
| Újrakérdezés: N = |

1. ***teszteset***

|  |
| --- |
| Bemenet – rossz indulók száma |
| N = 11  M = 0 |
| Kimenet |
| Újrakérdezés: M = |

**Fejlesztési lehetőségek**

1. A program fusson le többször felhasználó kívánsága szerint.
2. Grafikus kezelőfelület.
3. Tömbök mérete szükségesre csökkentése, akár vektorok használata.