

”Programozási alapismeretek”
beadandó feladat:
”ProgAlap beadandó” téma 1. feladat

Készítette: Bárdosi Bence
Neptun-azonosító: VY9NJN
E-mail: bardosi.bence@gmail.com

Kurzuskód: IP-08PAEG
Gyakorlatvezető neve: Pap Gábor Sándorné

2016-11-31

Tartalom

1	Felhasználói dokumentáció	2
1.1	Feladat	2
1.2	Futási környezet	2
1.3	Használat	2
1.3.1	A program indítása	2
1.3.2	A program bemenete	2
1.3.3	A program kimenete	3
1.3.4	Minta bemenet és kimenet	3
1.3.5	Hibalehetőségek	3
2	Fejlesztői dokumentáció	4
2.1	Feladat	4
2.2	Specifikáció	4
2.3	Fejlesztői környezet	4
2.4	Forráskód	5
2.5	Megoldás	5
2.5.1	Programparaméterek	5
2.5.2	Programfelépítés	5
2.5.3	Függvénystruktúra	5
2.5.4	Algoritmus	5
2.5.5	A kód	5
2.6	Tesztelés	5
2.6.1	Érvényes tesztesetek	5
2.6.2	Érvénytelen tesztesetek	5
2.7	Fejlesztési lehetőségek	5

1 Felhasználói dokumentáció

1.1 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 az egyéni versenyek győzteseinek rangsorát!

1.2 Futási környezet

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

1.3 Használat

1.3.1 A program indítása

A program a `.\VY9NJJN\bin\Release\VY9NJJN.exe` néven található a tömörített állományban. A `VY9NJJN.exe` fájl kiválasztásával indítható.

1.3.2 A program bemenete

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

Table 1: Bemenet

#	Adat	Magyarázat
1	N	Tanulók száma ($1 \leq N \leq 100$)
2	M	Versenyszek száma ($1 \leq M \leq 100$)
3	Min_1	Az 1. verseny minimum ponthatára ($0 \leq Min_1 \leq 50$)
4	Min_2	A 2. verseny minimum ponthatára ($0 \leq Min_2 \leq 50$)
.		
.		
.		
$M + 2$	Min_M	Az M . verseny minimum ponthatára ($0 \leq Min_M \leq 50$)
$M + 2 + 1$	$Para_1$	Az 1. verseny paraméterei (lásd: table2)
$M + 2 + 2$	$Para_2$	A 2. verseny paraméterei (lásd: table2)
.		
.		
.		
$M + 2 + M$	$Para_M$	Az M . verseny paraméterei (lásd: table2)

Table 2: Egy adott verseny paramétereit

#	Adat	Magyarázat
1	Ind_i	A versenyen indulók száma ($1 \leq Ind_i \leq N$)
2	$S_{i,1}$	Az első tanuló sorszáma ($1 \leq S_{i,1} \leq N$)
3	$P_{i,1}$	Az első tanuló által elért pont ($1 \leq P_{i,1} \leq 100$)
4	$S_{i,2}$	A második tanuló sorszáma ($1 \leq S_{i,2} \leq N$)
5	$P_{i,2}$	A második tanuló által elért pont ($1 \leq P_{i,2} \leq 100$)
.		
.		
.		
$2 * Ind_i$	S_{i,Ind_i}	Az Ind_i . tanuló sorszáma ($1 \leq S_{i,1} \leq N$)
$2 * Ind_i + 1$	P_{i,Ind_i}	Az Ind_i . tanuló által elért pont ($1 \leq P_{i,1} \leq 100$)

1.3.3 A program kimenete

A program kiírja az egyéni versenyek győzteseinek rangsorát. A kimenet első sorába az egyéni győzelmet elért tanulók számát, amelyet győztesek sorszáma követi, győzelmek száma szerint csökkenő, azon belül sorszám szerint növekvő sorrendben.

1.3.4 Minta bemenet és kimenet

1.3.5 Hibalehetőségek

Az egyes bemeneti adatokat a fenti mintának megfelelően kell megadni. Hiba, ha a bármely bemenő adat nem egész szám, nem esik az adott intervallumba, vagy ha nem szám. Hiba esetén a program "HIBA!"-t jelez és újrakérdezi az adott adatot.

Mintafutás hibás bemeneti adatok esetén:

2 Fejlesztői dokumentáció

2.1 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 az egyéni versenyek győzteseinek rangsorát!

2.2 Specifikáció

Be : $N \in \mathbb{N}$

$M \in \mathbb{N}$

$Min \in \mathbb{N}^M$

$P \in \mathbb{N}^{N \times M}$

Ki : $T \in \mathbb{N}$

$Nyert \in \mathbb{N}^T$

EF : $N \in [1..100]$

$M \in [1..100]$

$\forall i \in [1..M] : Min_i \in [0..50]$

UF : $T = \sum_{i=1}^N 1 \wedge$

$Db(i) \geq 1$

$Nyert \subseteq [1..N] \wedge$

$\forall i \in [1..T] : Db(Nyert_i) \geq 1 \wedge$

$\forall i \in [1..(T-1)] : Nyert_i < Nyert_{i+1}$

Def : $Db(x) = \sum_{j=1}^M 1$

$P_{x,j} = Max(P_{[1..N],j})$

$x < y \Leftrightarrow \begin{cases} Db(x) \neq Db(y) : Db(x) > Db(y) \\ Db(x) = Db(y) : x < y \end{cases}$

$Max(P_{X,y}) = (k > 0 \mid \exists i \in X : k = P_{i,y} \wedge \forall i \in X : k \geq P_{i,y})$

2.3 Fejlesztői környezet

IBM PC, exe futtatására alkalmas operációs rendszer (pl. Windows 10).
mingw32-g++.exe c++ fordítóprogram (v4.9.2), Code::Blocks (v16.01) fejlesztői környezet.

2.4 Forráskód

A teljes fejlesztői anyag –kicsomagolás után– a **ProgAlap_{be}** *eadnevű könyvtárban* található meg. A fejlesztéssorán *struktúra* :

2.5 Megoldás

2.5.1 Programparaméterek

2.5.2 Programfelépítés

A program által használt modulok (és helyük):

main.cpp	-	program, a forráskönyvtárban
iostream	-	képernyő-, és billentyűkezelés, a C++ rendszer része

2.5.3 Függvénystruktúra

2.5.4 Algoritmus

2.5.5 A kód

2.6 Tesztelés

2.6.1 Érvényes tesztesetek

2.6.2 Érvénytelen tesztesetek

2.7 Fejlesztési lehetőségek

- Adatok - a felhasználó igénye szerint - akár fájlból is fogadása.
- Hibás fájl-bemenetek felismerése, és a hiba helyének (sor számának) kiírása.
- Többszöri futtatás megszervezése.
- Külső adatbázis alapján tanulók sorszámának megfeleltetése neveikkel és neveik kiírása.