

Neptun kód: UW0FDO
Beadás verziószáma: 1

Név: Szász Roland

Feladat

Díjazottak

Egy pontverseny N résztvevője közül I. díjat kap mindenki, aki az elérhető pontok legalább 90%-át eléri; II. díjat kapnak azok, akik legalább 80%-ot teljesítettek, de nem érték el a 90%-ot, III. díjat pedig azok, akiknek teljesítménye legalább 70%, de kevesebb 80%-nál. A tanulók eredményeit egy névsor tartalmazza.

Készítsd el az I., a II., illetve a III. díjasok és a nem díjazottak névsorát!

Bemenet

A *standard bemenet* első sorában található a résztvevők száma ($1 \leq N \leq 100$) és a maximálisan elérhető pontszám ($1 \leq P \leq 100$). Az ezt követő N sor mindegyikében pontosvesszővel elválasztva szerepel egy név majd egy pontszám ($0 \leq P_i \leq 100$).

Kimenet

A *standard kimenet* első sorába az I., a második sorába a II., a harmadik sorába a III. míg a negyedik sorába a többiek névsorát kell írni a bemenetbeli előfordulások sorrendjében. Minden sor elején a sorban lévő elemek száma legyen. Minden elem után pontosvesszőt kell írni!

Adatreprezentáció

1	2	3	4	5
<pre># Bemenet n: 6 p: 100 r: [{nev: "Angyal Pofa", pont: 90}, {nev: "Bozontos Beata", pont: 73}, {nev: "Cukor Falat", pont: 85}, {nev: "Dohongo Diana", pont: 78}, {nev: "Eros Pista", pont: 83}, {nev: "Fickos Ferenc", pont: 64}] # Kimenet db1: 1 y1: [Angyal Pofa] db2: 2 y2: [Cukor Falat, Eros Pista] db3: 2 y3: [Bozontos Beata, Dohongo Diana] db4: 1 y4: [Fickos Ferenc]</pre>	<pre># Bemenet n: 2 p: 100 r: [{nev: "Szabályos Szabolcs", pont: 85}, {nev: "Betűs Bertalan", pont: "nyolcvan"}] # Kimenet HIBA: Nem megfelelő formátum!</pre>	<pre># Bemenet n: 2 p: 100 r: [{nev: "Balszerencses Bori", pont: 69}, {nev: "Gyenge Gary", pont: 10}] # Kimenet db1: 0 y1: [] db2: 0 y2: [] db3: 0 y3: [] db4: 2 y4: [Balszerencses Bori, Gyenge Gary]</pre>	<pre># Bemenet n: 2 p: 100 r: [{nev: "Balszerencses Bori", pont: 69}, {nev: "Gyenge Gary", pont: 10}] # Kimenet db1: 0 y1: [] db2: 0 y2: [] db3: 0 y3: [] db4: 2 y4: [Balszerencses Bori, Gyenge Gary]</pre>	<pre># Bemenet n: 150 p: 200 r: [... 150 db adat ...] # Kimenet HIBA: Előfeltétel sértés!</pre>
helyes	hibás	helyes	helyes	előfeltétel hiba

Specifikáció

Be:

$n \in \mathbb{N}$,

$p \in \mathbb{N}$,

$r \in \text{Versenyző}[1..n]$,

$\text{Versenyző} = (\text{nev}: S \times \text{pont}: \mathbb{N})$

Ki:

```
db1 ∈ N,  
y1 ∈ S[1..], // I. kategória  
  
db2 ∈ N,  
y2 ∈ S[1..], // II. kategória  
  
db3 ∈ N,  
y3 ∈ S[1..], // III. kategória  
  
db4 ∈ N,  
y4 ∈ S[1..] // IV. kategória
```

Ef:

```
1 <= n <= 100 és  
1 <= p <= 100 és  
∀i∈[1..n]:(0 <= r[i].pont <= 100)
```

Uf:

```
(db1,y1) = KIVÁLOGAT(i=1..n, (r[i].pont >= (p*0.90)), r[i].nev) és  
(db2,y2) = KIVÁLOGAT(i=1..n, ((p*0.90) > r[i].pont és r[i].pont >= (p*0.80)),  
r[i].nev) és  
(db3,y3) = KIVÁLOGAT(i=1..n, ((p*0.80) > r[i].pont és r[i].pont >= (p*0.70)),  
r[i].nev) és  
(db4,y4) = KIVÁLOGAT(i=1..n, ((p*0.70) > r[i].pont), r[i].nev)  
  
// I: r[i].pont >= (p*0.90)  
// II: (p*0.90) > r[i].pont >= (p*0.80)  
// III: (p*0.80) > r[i].pont >= (p*0.70)  
// IV: (p*0.70) > r[i].pont
```

Link: [Specification editor](#)

Kiválogatás sablon

i	T(i)	f(i)
e	→ HAMIS	
e+1	→ IGAZ	1 f(e+1)
e+2	→ IGAZ	2 f(e+2)
u	→ HAMIS	

Feladat

Adott az egész számok egy $[e..u]$ intervalluma, egy ezen értelmezett $T:[e..u] \rightarrow \text{Logikai feltétel}$ és egy $f:[e..u] \rightarrow H$ függvény. Határozzuk meg az f függvény az $[e..u]$ intervallum azon értékeinél felvett értékeit, amelyekre a T feltétel teljesül!

Specifikáció

Be: $e \in \mathbb{Z}$, $u \in \mathbb{Z}$

Ki: $db \in \mathbb{N}$, $y \in H[1..db]$

Ef: -

Uf: $db = \text{DARAB}(i=e..u, T(i))$ és
 $\forall i \in [1..db]:$
 $\exists j \in [e..u]: T(j) \text{ és } y[i] = f(j)$
 és $y \subseteq (f(e), f(e+1), \dots, f(u))$

Rövidítve:

Uf: $(db, y) = \text{KIVÁLOGAT}(i=e..u, T(i), f(i))$

Algoritmus

db := 0	Változó
i = e .. u	i: Egész
T	T(i)
db := db + 1	-
y[db] := f(i)	



Visszavezetés

Kiválogatás		
y	~	y1
e..u	~	1..n
T(i)	~	(r[i].pont >= (p*0.90))
f(i)	~	r[i].nev

Kiválogatás		
y	~	y2
e..u	~	1..n
T(i)	~	((p*0.90) > r[i].pont és r[i].pont >= (p*0.80))
f(i)	~	r[i].nev

Kiválogatás		
y	~	y3
e..u	~	1..n

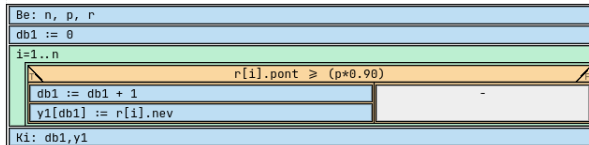
T(i)	~	((p*0.80) > r[i].pont és r[i].pont >= (p*0.70))
f(i)	~	r[i].nev

Kiválogatás		
y	~	y4
e..u	~	1..n
T(i)	~	((p*0.70) > r[i].pont)
f(i)	~	r[i].nev

Algoritmus

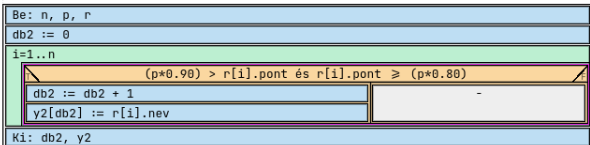
```
// I. kategória
n: Természetes
p: Természetes
r: Tömb[1..n: Versenyző]
Versenyző: Rekord(nev: Szöveg x pont: Természetes)
```

i: Egész



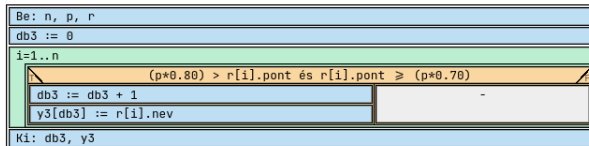
```
// II. kategória
n: Természetes
p: Természetes
r: Tömb[1..n: Versenyző]
Versenyző: Rekord(nev: Szöveg x pont: Természetes)
```

i: Egész



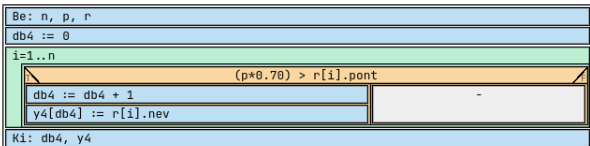
```
// III. kategória
n: Természetes
p: Természetes
r: Tömb[1..n: Versenyző]
Versenyző: Rekord(nev: Szöveg x pont: Természetes)
```

i: Egész



```
// IV. kategória
n: Természetes
p: Természetes
r: Tömb[1..n: Versenyző]
Versenyző: Rekord(nev: Szöveg x pont: Természetes)
```

i: Egész



Link: [Structogram editor](#)

2. fázishoz

Kód

```
// Név: Szász Roland
// Neptun kód: UW0FDO
// Inf-es e-mail: UW0FDO@inf.elte.hu
using System;
class Program { public struct Versenyző { public string nev; public int pont; }
static void Main(string[] args)
{
    #region
    int n;
    int p;
    Versenyző[]

    int db1 = 0;
    int db2 = 0;
    int db3 = 0;
    int db4 = 0;
    string[] y1;
    string[] y2;
    string[] y3;
    string[] y4;

    #endregion

    #region
    string[] elsoSor = Console.ReadLine().Split(' ');
    n = int.Parse(elsoSor[0]);
    p = int.Parse(elsoSor[1]);
    }
}
```

```

r = new Versenyző[n];
y1 = new string[n];
y2 = new string[n];
y3 = new string[n];
y4 = new string[n];

for (int i = 0; i < n; i++)
{
    string sor = Console.ReadLine();
    string[] adatok = sor.Split(';');
    r[i].nev = adatok[0];
    r[i].pont = int.Parse(adatok[1]);
}
#endregion

#region Feldolgozás

// 1. I. Kategória (>= 90%)
for (int i = 0; i < n; i++)
{
    if (r[i].pont >= (p * 0.90))
    {
        y1[db1] = r[i].nev;
        db1++;
    }
}

// 2. II. Kategória (80% - 90%)
for (int i = 0; i < n; i++)
{
    if (r[i].pont < (p * 0.90) && r[i].pont >= (p * 0.80))
    {
        y2[db2] = r[i].nev;
        db2++;
    }
}

// 3. III. Kategória (70% - 80%)
for (int i = 0; i < n; i++)
{
    if (r[i].pont < (p * 0.80) && r[i].pont >= (p * 0.70))
    {
        y3[db3] = r[i].nev;
        db3++;
    }
}

// 4. IV. Kategória (< 70%)
for (int i = 0; i < n; i++)
{
    if (r[i].pont < (p * 0.70))
    {
        y4[db4] = r[i].nev;
        db4++;
    }
}
#endregion

```

```

        #region
        KiirKategoria(db1,
        KiirKategoria(db2,
        KiirKategoria(db3,
        KiirKategoria(db4,
        #endregion
    }

    static void KiirKategoria(int db, string[] nevek)
    {
        Console.Write(db);
        for (int i = 0; i < db; i++)
        {
            Console.Write($"{{nevek[i]}}");
        }
        Console.WriteLine(";");
    }
}

```

Bíró eredmény

Összpont: 100/100

Teszt#	Pont	...Verdikt...	futási idő
1.1	10/10	HELYES	0.031 sec
2.1	10/10	HELYES	0.032 sec
3.1	10/10	HELYES	0.032 sec
4.1	10/10	HELYES	0.032 sec
5.1	10/10	HELYES	0.032 sec
6.1	10/10	HELYES	0.036 sec
7.1	10/10	HELYES	0.032 sec
8.1	10/10	HELYES	0.034 sec
9.1	10/10	HELYES	0.032 sec
10.1	10/10	HELYES	0.033 sec

Beadva: 2025-11-30 16:51:23.0