Neptun kód: **EBYPPB** Név: **Pőcze Máté**

Beadás verziószáma: 1.

Feladat

Programozási tételek összeépítése

*

Drága beszerzések

A Tesó™ üzletlánc N féle terméket árul. Ismerjük az összes termék árát és egy heti eladásai számát.

Irj programot, amely megadja azt a legnagyobb árat, amelynél az egy héten eladott termékek legalább felének az ára nagyobb vagy egyenlő!

Bemenet

A standard bemenet első sorában a termékek száma (1≤N≤100) található. A következő N sorban soronként az egyes termékek ára (1≤Ár_i≤200) és egy heti eladási száma (0≤Db_i≤1000) van. Az adatok ár szerint csökkenően rendezettek

Kimenet

A standard kimenet egyetlen sorába azt a legnagyobb árat kell írni, amelynél az egy héten eladott termékek legalább felének az ára nagyobb vagy egyenlő!

Példa

Bemenet Kimenet
6 6
9 1
8 2
8 0
7 3

5 5

6 4

Korlátok

Időlimit: 0.1 mp. Memórialimit: 32 MiB

Specifikáció

```
Be: n \in N, t \in Termek[1..n], Termek = (ar:N \times elad:N)
```

Ki: legnagyobbAr ∈ N

Fv: elemekKozepe: N -> R,

elemekKozepe(i) = **SZUMMA**(=1..n, t[].elad) / 2

Fv: osszEladott: N -> N,

osszEladott(i) = **SZUMMA**(j=1..i, t[j].elad)

Ef: $1 \le n \le 100$ and $\forall i \in [1..n]$: $(1 \le t[i].ar \le 200$ and $0 \le t[i].elad \le 1000)$

Uf: (,,legnagyobbAr) = MAXHA(i=1..n, t[i].ar, elemekKozepe(n) <= osszEladott(i))

Sablon

Összegzés sablon

Feladat

Adott az egész számok egy [e..u] intervalluma és egy f:[e..u]→H függvény. A H halmaz elemein értelmezett az összeadás művelet. Határozzuk meg az f függvény [e..u] intervallumon felvett értékeinek az összegét, azaz a $\sum_{i=e}^{u} f(i)$ kifejezés értékét! (e>u esetén ennek az értéke definíció szerint a nulla elem)

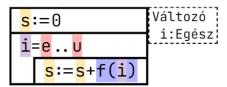
Specifikáció

Be: e∈Z, u∈Z Ki: s∈H

Ef: -

Uf: s=SZUMMA(i=e..u, f(i))

Algoritmus





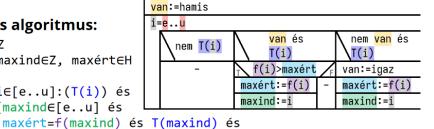
Feltételes maximumkeresés sablon

Feladat

Adott az egész számok egy [e..u] intervalluma, egy f:[e..u]→H függvény és egy T:[e..u]→Logikai feltétel. A H halmaz elemein értelmezett egy teljes rendezési reláció. Határozzuk meg, hogy az [e..u] intervallum T feltételt kielégítő elemei közül az f függvény hol veszi fel a legnagyobb értéket, és mondjuk meg, Változó mekkora ez az érték! i:Egész

Specifikáció és algoritmus:

```
Be: e∈Z, u∈Z
Ki: van∈L, maxind∈Z, maxért∈H
Ef: -
Uf: van = \exists i \in [e..u]:(T(i)) és
    van -> (maxind∈[e..u] és
```



Uf: (van, maxind, maxért) = FELTMAX(i=e..u, f(i), T(i))

 $\forall i \in [e..u]: (T(i) \rightarrow maxért \rightarrow = f(i)))$



Visszavezetés

```
Összegzés – elemekKozepe:
e..u ~ 1..n

T(i) ~ t[]].elad

Összegzés – osszEladott:
e..u ~ 1..i

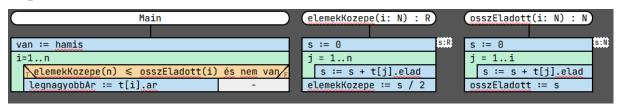
T(i) ~ t[j].elad

Feltételes maximumkeresés:
e..u ~ 1..n

f(i) ~ t[i].ar

T(i) ~ elemekKozepe(n) < osszEladott(i)
```

Algoritmus



Megjegyzés: az 1. fázishoz képest a feltételes maximumkeresést kiegészítettem egy van változóval, az elemekKozepe függvényben 'i' helyett 'j' változót használok, és az 's' típusa N helyett R.

Kód (C#)

```
/*

* Pőcze Máté

* EBYPPB

* ebyppb@inf.elte.hu

*

* Feladat: Drága beszerzések

*/

using System;
using System.Collections.Generic;

namespace beadando2
{

internal class Program
{

static int n;

static Termek[] t;

struct Termek
{

public int ar;

public int elad;
```

```
}
static void beolvas()
  string bemenet;
  string[] darabok;
  bemenet = Console.ReadLine();
  n = int.Parse(bemenet);
  t = new Termek[n + 1];
  for (int i = 1; i <= n; i++)
    bemenet = Console.ReadLine();
    darabok = bemenet.Split(' ', StringSplitOptions.RemoveEmptyEntries);
    t[i].ar = int.Parse(darabok[0]);
    t[i].elad = int.Parse(darabok[1]);
  }
}
static double elemekKozepe()
  double s = 0;
  for (int j = 1; j \le n; j++)
    s += t[j].elad;
  return s / 2;
static int osszEladott(int i)
  int s = 0;
  for (int j = 1; j \le i; j++)
    s += t[j].elad;
  return s;
```

```
static void Main(string[] args)
      // Beolvasás
       beolvas();
      // Deklaráció
      int legnagyobbAr = 0;
       bool van = false;
      // Feldolgozás
      for (int i = 1; i <= n; i++)
         if (elemekKozepe() <= osszEladott(i) && !van)
           legnagyobbAr = t[i].ar;
           van = true;
         }
      }
      // Kiírás
       Console.WriteLine(legnagyobbAr);
    }
  }
}
```

Bíró pontszám és képernyőkép



Saját tesztfájlok

73

61

5 2

5 3

4 2

ki2.txt:

5

be3.txt:

8

8 4

73

6 2 58

43 38

2 1

16

ki3.txt:

4