

MEGOLDÁS SABLON

C. BEADANDÓ HÁZI FELADAT.

Név: Golyha Gergő

Neptunkód: A7MMZ1

Feladat: Gazdaságos üzemeltetés

Programozási tételek – Tételek összeépítése

*

Gazdaságos üzemeltetés

A Budapest-Székesfehérvár vasútvonalon egy vonat kalauza minden állomáson feljegyezte, hogy hányan szálltak fel a vonatra, illetve hányan szálltak le. (Budapesten biztos nincs leszálló, Székesfehérváron biztos nincs felszálló, aki leszállt, az nem száll vissza.)

Készíts programot, amely eldönti, hogy gazdaságos-e a vonat üzemeltetése, ha egy utasnak egy állomásnyi távolság N Ft-ba kerül, a vonat egy állomásnyi útja pedig M Ft-ba kerül!

Bemenet

A standard bemenet első sorában az állomások száma van ($1 \leq \text{állomásszám} \leq 1000$), második sorában az egy állomásnyi távolság ára személyenként ($0 < N \leq 100$) és az egy állomásnyi vonatút költsége ($0 < M \leq 100\,000$), majd soronként szóközzel elválasztva az egyes állomásokon leszállók ($0 \leq \text{leszállók} \leq 800$) és felszállók ($0 \leq \text{felszállók} \leq 800$) száma.

Kimenet

A standard kimenet első sorába egyetlen egész számot kell írni, amely 1, ha gazdaságos az üzemeltetés, és 0, ha nem!

Példa

Bemenet	Kimenet
6	1
100 1000	
0 15	
10 30	
0 32	
48 0	
20 27	
26 0	

Korlátok

Időlimit: 0.1 mp.


Memórialimit: 32 MB

Pontozás: A tesztek 40%-ában a bemenet hossza ≤ 500 , a leszállók és felszállók száma ≤ 400

Specifikáció		max: 20 pont
<div>Link</div> <div>Be: $adb \in \mathbb{N}, u \in \mathbb{N}, v \in \mathbb{N}, all \in U_{tas}[1..adb], U_{tas} = Le \times Fel, Le = \mathbb{N}, Fel = \mathbb{N}$</div> <div>Ki: $gazd \in L$</div> <div>Sa: $s \in \mathbb{N}$</div> <div>Fv: $vonaton: \mathbb{N} \rightarrow \mathbb{N}, vonaton(x) = SZUMMA(i=1..x, all[i].fel-all[i].le)$</div> <div>Ef: -</div> <div>Uf: $s = SZUMMA(i=1..adb-1, vonaton(i))$ és $gazd = s * u > (adb-1) * v$</div>		Itt megalkotva, vagy másolva más környezetből vagy kézzel írt megoldás fényképe
Sablon + forrásmegjelölés		max: 5 pont
<div>Összegzés (mind a kettő)</div> <div>programozási minták.pdf / 4. oldal (canvas)</div> <div><div>Specifikáció</div><div>Be: $e \in \mathbb{Z}, u \in \mathbb{Z}$</div><div>Ki: $s \in \mathbb{H}$</div><div>Ef: -</div><div>Uf: $s = SZUMMA(i=e..u, f(i))$</div></div> <div><div>Algoritmus</div><div><div>s:=0</div><div>i=e..u</div><div>s:=s+f(i)</div></div></div>		Lehet képernyőkép vagy másolat az előadás anyagából. Forrásmegjelölés: a forrásfájl neve (publikálás útvonala) és oldalszám/dia sorszáma.
Visszavezetési táblázat (megfeleltetés)		max: 5 pont
<div><div>Összegzés (fő)</div><div>e..u ~ 1..adb-1</div><div>f(i) ~ vonaton(i)</div></div> <div><div>Összegzés (vonaton)</div><div>e..u ~ 1..x</div><div>f(i) ~ all[i].fel-all[i].le</div></div>		A specifikáció és a minta közötti kapcsolat leírása
Algoritmus		max: 20 pont
<div><div>Változó s:Egész</div><div>s:=0</div><div>i=1..adb-1</div><div><div>s:=s+vonaton(i)</div></div><div>gazd:=s*u>(adb-1)*v</div></div> <div><div>vonaton(x:Egész):Egész</div><div>Változó: s,i:Egész</div><div>s:=0</div><div>i=1..x</div><div><div>s:=s+all[i].fel-all[i].le</div></div><div>vonaton:=s</div></div>		Itt megalkotva vagy képernyőkép vagy kézzel írt megoldás fényképe

Módosítások**max 10 pont**

Itt kell dokumentálni, ha a beadott tervet a visszajelzés vagy a kódolás és tesztelés során kiderülő részletek miatt módosítani kell. A dokumentáció tartalmazza

<p>Uf: $s = \text{SZUMMA}(i=1..adb-1, \text{vonaton}(i))$ és $\text{gazd} = s * u > adb-1 * v$</p> <p>$\text{gazd} := s * u > (adb-1) * v$</p> 	<p>A biro-ban 100-at kaptam, ha adb-ot nem csökkentettem, de találtam olyan esetet, ahol helytelen az eredmény.</p> <p>Mivel adb az állomások száma, a vonat adb-1 állomásnyi utat tett meg, ezért a csökkentett értékkel kell számolni.</p> <p>A biro-ba lehet érdemes egy olyan tesztesetet beletenni, mint pl nálam a harmadik (a biro elfogadta a kódot (-1) nélkül is de itt nem működne).</p>
--	---

(A módosítások jellegétől függően, lehet szövegdoboz nyilakkal vagy lábjegyzet is az indoklás.)

Maximális pontszámot ér az összes módosítás dokumentálása. Ha nem volt módosítás és a kód a korábban beadott tervet képezi le, akkor megkapható a maximális pontszám (nem a mennyiség, hanem a minőség számít.) Ebben az esetben is szerepeljen itt az első részteljesítés.

Kód forrása**max 11 pont**

biro.elte.hu téma: Tételek összeépítése feladat: Gazdaságos üzemeltetés beadás (feltöltés) sorszáma: 4 pontszám: 100	Break, exit vagy try-catch használata esetén a 11 pontból legfeljebb 1 pont adható.
Ha volt: hibás tesztesetek sorszáma és a mellettük megjelenő hibaüzenetek	A hibaüzenetek és a kód függvényében, akár 80%-os eredmény is lehet maximális pontszámú

Kód**max 26 pont**

2-2-2 pont: megjegyzésbe írt információk, külön választott deklaráció és kiírás.

8 pont a külön választott beolvasás/adatok eltárolása Ezen belül break, exit vagy try-catch használata esetén a 8 pontból 0 pont adható.

12 pont az algoritmus szabványos kódolása. Ezen belül break, exit vagy try-catch használata esetén a 26 pontból 0 pont adható.

```

using System;

// Golyha Gergő
// A7MMZ1
// golhyagergo@gmail.com
// Tételek összeépítése: Gazdaságos üzemeltetés

namespace beadando2
{
    internal class Program
    {
        struct Utas
        {
            public int le;
            public int fel;
        }

        static void Main(string[] args)
        {
            (int u, int v, Utas[] all) = beolvasas();
            bool gazd = feldolgozas(u, v, all);
            kiiras(gazd);
        }

        static (int u, int v, Utas[] all) beolvasas()
        {
            int u, v;
            int n;
            Utas[] all;
            Console.Error.Write("n = ");
            // Ef: 1 <= n <= 100
            int.TryParse(Console.ReadLine(), out n);
            all = new Utas[n];
            Console.Error.Write("u = , v = ");
            string[] be1 = Console.ReadLine().Split();
            int.TryParse(be1[0], out u);
            int.TryParse(be1[1], out v);
            // Ef: 0 <= u <= 100
            //      0 <= v <= 1000000
            for (int i = 0; i < n; i++)
            {
                Console.Error.Write("{0}. állomáson le = , fel = ", i + 1);
                string[] be2 = Console.ReadLine().Split();
                int.TryParse(be2[0], out all[i].le);
                int.TryParse(be2[1], out all[i].fel);
                // Ef: 0 <= le <= 800
                //      0 <= fel <= 800
            }
            return (u, v, all);
        }

        static bool feldolgozas(int u, int v, Utas[] all)
        {
            int vonaton(int x)
            {
                int s = 0;
                for (int i = 0; i <= x; i++)
                {
                    s += all[i].fel - all[i].le;
                }
                return s;
            }

            bool gazd;
            int s = 0;
            for (int i = 0; i < all.Length - 1; i++)
            {
                s += vonaton(i);
            }
            return s * u > (all.Length - 1) * v;
        }

        static void kiiras(bool gazd)
        {
            Console.Error.WriteLine("Az üzemeltetés {0} gazdaságos", gazd ? "" : "nem");
            Console.WriteLine(gazd ? 1 : 0);
        }
    }
}

```

Teszt**max 3 pont**

A feladathoz adott 2 teszten túl még 3 teszt.

(A tervmódosítást alátámasztó tesztek, speciális esetek tesztjei.)

	Saját 1.	Saját 2.	Saját 3
Bemenet:	2 100 100 0 1 1 0	5 50 1000 0 100 10 50 31 0 15 7 0 101	3 100 99 0 1 0 0 1 0
Kimenet:	0	1	1