

Neptun kód: UJI470 Név: **Nagy Levente**
Beadás verziószáma: 1.
Később kiegészítéssel(!)
2. (javított Uf...)
3. (javított Algoritmus)

Feladat

Programozási tételek összeépítése

*

Évek mikor több bor készült, mint azelőtt

Egy pincészet egyfélé boráról évente tárolja, hogy mennyit készített belőle és hány forintért árusítja.

Írj programot, amely megadja azon évek számát és sorszámait, amikor többet készítettek, mint bármely korábbi évben!

Bemenet

A standard bemenet első sorában az évek száma ($1 \leq N \leq 100$), alatta soronként egy-egy év bor mennyisége és ára van ($1 \leq M_i, A_i \leq 10\ 000$).

Kimenet

A standard kimenet egyetlen sorába azon évek számát és sorszámait kell kiírni, amikor többet készítettek, mint bármely korábbi évben (ha volt egyáltalán korábbi év)! A sorszámokat növekvő sorrendben, egy-egy szóközzel elválasztva kell kiírni! Ha egyetlen ilyen év sincs, akkor egyetlen 0 legyen a kimenet!

Példa

Bemenet	Kimenet
6	3 2 3 6
500 1500	
1900 2000	
2000 1500	
200 3800	
300 1000	
2100 3000	

Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

Megoldás sablon „C” beadandó 1. fázishoz (lecsérélendő a konkrét beadandó megnevezésére)

Specifikáció

[https://progalap.elte.hu/specifikacio/?data=H4sIAAAAAAAAC-pWSzUrDQBSFX%2BVyVymMYfJX9UIWLhItW5UhDSLpJnKBJtlEotSunDX9%2FJNf-BKZ2oxt6RBdhMyQb86Zc26WWL%2BIqZzJadLIskDCoSDIUvh...
LHtrM0ZmodWnNRFO%2ByFk80hjdIKhr3YFJNipEkyFuFppxv1dThPFbAxYzgRC3uZ2TlP0wPQh-hdPXx%2BXN9eDu4sGW6sGFhse4dlxrZ2VPTN4NFahK5tS33PaLHLMJA9QlaNqJ-saKVpiljQJEmYpQV%2F5b48RRGoHALD8DQUB50zFAifgfMWoIc55y7jcxKgvXT-pui3hnJsTTKmYnR8soRiGbvtWTE3jqrbomiFwGHoN%2BjAy-LZC6QEBlWon59bpD4iu115Xd35Wtn39iVu9OVA-fIDRr7HHGZ09zP6xwM6BwGD7oDerq9hQk5nQF9PsX9qZLSTb%2FxfAv7PSQemluLVN57CMOCeBAAA](https://progalap.elte.hu/specifikacio/?data=H4sIAAAAAAAAC-pWSzUrDQBSFX%2BVyVymMYfJX9UIWLhItW5UhDSLpJnKBJtlEotSunDX9%2FJNf-BKZ2oxt6RBdhMyQb86Zc26WWL%2BIqZzJadLIskDCoSDIUvh...)

Sablon

Feladat

Adott az egész számok egy $[e..u]$ intervalluma és egy $f:[e..u] \rightarrow H$ függvény. A H halmaz elemein értelmezett egy teljes rendezési reláció. Határozzuk meg, hogy az f függvény hol veszi fel az $[e..u]$ nem üres intervallumon a legnagyobb értéket, és mondjuk meg, mekkora ez a maximális érték!

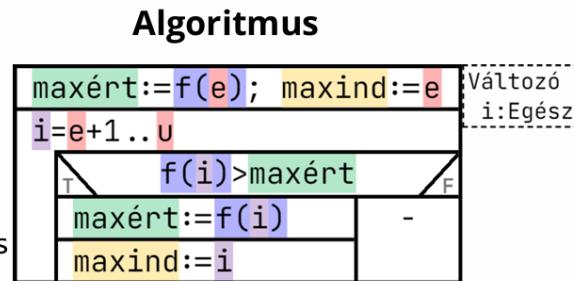
Specifikáció

Be: $e \in \mathbb{Z}$, $u \in \mathbb{Z}$
 Ki: $\maxind \in \mathbb{Z}$, $\maxért \in H$
 Ef: $e <= u$
 Uf: $\maxind \in [e..u]$ és
 $\forall i \in [e..u]: (f(\maxind) \geq f(i))$ és
 $\maxért = f(\maxind)$

Rövidítve:

Uf: $(\maxind, \maxért) = \text{MAX}(i=e..u, f(i))$

Kiválogatás sablon



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!

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

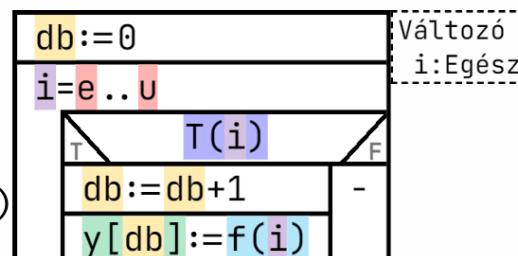
Specifikáció

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



Megoldás sablon „C” beadandó 1. fázishoz (lecsérélendő a konkrét beadandó megnevezésére)

Visszavezetés

(db,y)	(j,tomb)
i=e..u	i=1..db
T(i)	(,borAdat[i].mennyiseg) = MAX(v=2..i, borAdat[v].mennyiseg)
f(i)	i

(maxind,maxért)	(,borAdat[i].mennyiseg)
i=e..u	v=2..i
f(i)	borAdat[i].mennyiseg

Algoritmus

<https://progalap.elte.hu/stuki/?data=H4sIAAAACq1V72%2FaMBD9V6KrtC%2B1siSFUli-pVDStY936pT9XxAfHvkAg2OAYWlrlf58ciDERSOtUvuT0uLz3znfnvELKo-QWBf1Rv1IJ6PQijqBk06yEQGMMoFqi6HlphnGYEcM2Qa%2BQYRkmMOrVeYDsZ6NG88mrj-kswCBKVUo9OaIYco5CmgLNMuRgF5OEVqQ42yOgiEQYMM04wpFl%2BfQ6sF4OV3ks%2FgY-CPyY%2FjzuXNxdQ78gG7zSdBl3mq6TPcqappyg0EBA47OGFvCYeLFU55zqHo%2F7%2FgS-FWKY5DrZhqog38tpeSDwtJ3Ev9P1Rn3gT%2BoxK49hr2%2BzQ4YCCQO0%2B7t4277X1boEt74ub0XDa7Hzd733vqc0e4mXlb7E8KstTyTnEjlznZ-nwuQ3a7V%2B5pmGallhQ81akUhqod%2BX4KBGLJl9drP92tmgoCkTzJli%2BP36wDC2w5sK7%2Fo%2BCr55vw6EKWBU%2BuocXPfpt5SywJcfV3fyxM7vcKyek-nBr7NrHic978GPvWoLFveSo5h9iRc4Zmt1ya%2FNWsah4Xzjx6nwb6ix1aswRqjh1FBRtutdPpWKmyl8Weiqmh6oetoQK2anBY%2F3k9ddyYW9BprM7KfRz1vba3OOXp4mx0eHj62QTGmp3zypoz%2BI41e9u8v5t2v0w3LU8l5xC%2F56LatX1pO%2FR9Hu9aP1tUURDIltZozLQeKToz-BVxB0Yjh%2F0VQAgQVVKY0zc5kDEFBSvvGUD%2BXT3SZtbc2g35FyVGuo6BuD-mVTld2HVsB4cJEkQBAEQODDPxMz-IQZlwi8VB3UZJGTWCo4CWEUOanDSgv%2Bp%2Fh7LxQMm54GviOlojk2aeFTGyCsMGo1WUI-BGvsQAjQ2c%2BXx9I55zxG1bzW9fMGCurYoxVWBjVGCacrbBmc%2FVvEJy-wZt1EiFjjx%2B6JFMUf62nakr8HAAA%3D>

Kód (C#)

using System;

Megoldás sablon „C” beadandó 1. fázishoz (lecsérélendő a konkrét beadandó megnevezésére)

```
/*
Nagy Levente
UJI470
levente0517@gmail.com
*/



class Program
{
    public struct Bor
    {
        public int mennyiseg;
        public int ar;
        public Bor(int a, int b)
        {
            mennyiseg = a;
            ar = b;
        }
    }

    static void Main(string[] args)
    {
        int db = int.Parse(Console.ReadLine());

        Bor[] borAdat = new Bor[db];

        string[] temp = new string[db];

        for (int i = 0; i < db; i++)
        {
            temp = Console.ReadLine().Split(' ');

```

Megoldás sablon „C” beadandó 1. fázishoz (lecsérélendő a konkrét beadandó megnevezésére)

```
borAdat[i] = new Bor(int.Parse(temp[0]), int.Parse(temp[1]));

}

int maxertek = borAdat[0].mennyiseg;
int[] tomb = new int[db];
int j = 0;

for (int i = 1; i < db; i++)
{
    if (borAdat[i].mennyiseg > maxertek)
    {
        maxertek = borAdat[i].mennyiseg;
        tomb[j] = i + 1;
        j++;
    }
}

Console.WriteLine(j + " ");

for (int i = 0; i < j; i++)
{
    Console.WriteLine(tomb[i] + " ");
}

}
```

Bíró

pontszám

és

képernyőkép



Eötvös Loránd Tudományegyetem
Informatikai Kar

User: Nagy Levente, Topic: Programozás 2. beadandó, Deadline: 2024-12-08 23:59:59

CHANGE TOPIC
SUBMIT
RESULT
DOWNLOAD
BACKUP
SUBMITTED
MAIN PAGE
EXIT/LOGIN

Result of the last submission

Total points: 100/

Test#	Point	Verdict	CPU time
1.1	1/1	Helyes	0.032 sec
1.2	2/2	Helyes	0.032 sec
2.1	1/1	Helyes	0.033 sec
2.2	2/2	Helyes	0.033 sec
3.1	1/1	Helyes	0.032 sec
3.2	2/2	Helyes	0.032 sec
4.1	1/1	Helyes	0.030 sec
4.2	2/2	Helyes	0.030 sec
5.1	1/1	Helyes	0.031 sec
5.2	2/2	Helyes	0.031 sec
6.1	1/1	Helyes	0.033 sec
6.2	2/2	Helyes	0.033 sec
7.1	1/1	Helyes	0.033 sec
7.2	2/2	Helyes	0.033 sec
8.1	1/1	Helyes	0.033 sec
8.2	2/2	Helyes	0.033 sec
9.1	1/1	Helyes	0.032 sec
9.2	3/3	Helyes	0.032 sec
10.1	1/1	Helyes	0.032 sec
10.2	3/3	Helyes	0.032 sec
11.1	1/1	Helyes	0.032 sec
11.2	3/3	Helyes	0.032 sec
12.1	1/1	Helyes	0.033 sec
12.2	3/3	Helyes	0.033 sec
13.1	1/1	Helyes	0.032 sec
13.2	3/3	Helyes	0.032 sec
14.1	1/1	Helyes	0.031 sec
14.2	3/3	Helyes	0.031 sec

Megoldás sablon „C” beadandó 1. fázishoz (lecsérélendő a konkrét beadandó megnevezésére)

14.2	3/3	Helyes	0.031 sec
15.1	1/1	Helyes	0.030 sec
15.2	3/3	Helyes	0.030 sec
16.1	1/1	Helyes	0.025 sec
16.2	3/3	Helyes	0.025 sec
17.1	1/1	Helyes	0.031 sec
17.2	3/3	Helyes	0.031 sec
18.1	1/1	Helyes	0.031 sec
18.2	3/3	Helyes	0.031 sec
19.1	1/1	Helyes	0.031 sec
19.2	3/3	Helyes	0.031 sec
20.1	1/1	Helyes	0.032 sec
20.2	3/3	Helyes	0.032 sec
21.1	1/1	Helyes	0.031 sec
21.2	3/3	Helyes	0.031 sec
22.1	1/1	Helyes	0.031 sec
22.2	3/3	Helyes	0.031 sec
23.1	1/1	Helyes	0.031 sec
23.2	3/3	Helyes	0.031 sec
24.1	1/1	Helyes	0.032 sec
24.2	3/3	Helyes	0.032 sec
25.1	1/1	Helyes	0.032 sec
25.2	3/3	Helyes	0.032 sec
26.1	1/1	Helyes	0.031 sec
26.2	3/3	Helyes	0.031 sec
27.1	1/1	Helyes	0.032 sec
27.2	3/3	Helyes	0.032 sec

Date of submission: 2024-12-02 10:52:11.0

Saját tesztfájlok

6		5	
500 1500		3000 100	
1900 2000	4	3100 200	
2000 1500	400 3400	4300 6700	
200 3800	1200 200	4000 4500	
300 1000	100 1000	5000 3000	
2100 3000	2000 1000		
3 2 3 6	2 2 4	3 2 3 5	