

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 egyféle boráról évente tárolja, hogy mennyit készített belőle és hány forintért árusítja.

Írj programot, amely megadja azon évek számát és sorszámain, 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 sorszámaint 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

6  
500 1500  
1900 2000  
2000 1500  
200 3800  
300 1000  
2100 3000

Kimenet

3 2 3 6

#### Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

## Specifikáció

<https://progalap.elte.hu/specifikacio/?data=H4sIAAAAAAAAAAC-pWSzUrDQBSFX%2BVyVymMYfJX9UIWLYhltW5UhDSLpJnKBjtIEotSunDX9%2FJNf-BKZ2ox6t6RBdhMyQb86Zc26WWL%2BIqZzJadLlSkDCoSDIUvhar2HMIC2rQZY0m%2B2wrC-LHtrM0ZmodWnNRFO%2ByFk80hjdIKhr3YFJNipEkyFuFppxv1dThPFbAxYzgRC3uZ2TlP0wPQh-hdPXx%2BXN9eDu4sGW6sGFhse4dlxrZ2VPTN4NFahK5tS33PaLHLMJA9QlaNqJ-saKVpiljQJEmYpQV%2F5b48RRGoHALD8DQUB50zFAifgfMWOLc55y7jcxKgvXT-pui3hnJsTTKmYnR8soRiGbvtWTE3jqrBomiFwGHoN%2BjAy-LZC6QEBlWon59bpD4iu115Xd35Wtn39iVu9OVA-flDRr7HHGZ09zP6xwM6BwGD7oDerq9hQk5nQF9PsX9qZLSTb%2FxfAv7PSQemluLVN57CMOCeBAAA>

## 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:  $e \leq u$

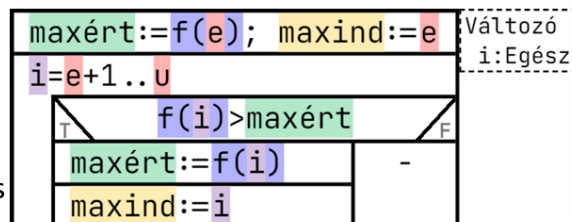
Uf: **maxind**  $\in [e..u]$  és

$\forall i \in [e..u]: (f(\mathbf{maxind}) \geq f(i))$  és  
**maxért**  $= f(\mathbf{maxind})$

Rövidítve:

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

### Algoritmus



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

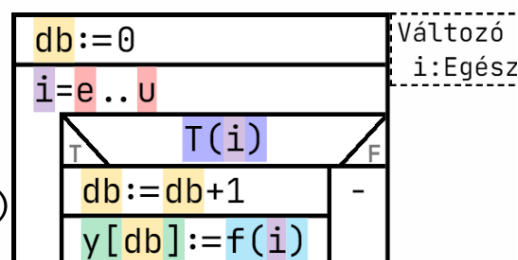
Ef: -

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



### 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=H4slAAAAAAAAACq1V72%2FaMBD9V6KrtC%2B1siSFULi-pVDStY936pT9XAfHvkAg2OAYWlrlf58ciDErSotUvuT0uLz3znfnvELKo-QWBf1Rv1IJ6PQijqBk06yEQGMoFqi6HlphnGYEcM2Qa%2BQYRkmMOrVeYDsZ6NG88mrj-kswCBKVUo9OalYco5CmglnMuRgF5OEVqQ42yOgiEQYMM04wpFl%2BfQ6sF4OV3ks%2FgY-CPyY%2FjzuXNxdQ78gG7zSdBl3mq6TPcqaapyg0EBA47OGFvCYeLFU55zqHo%2F7%2FgS-FWKY5DrZhqog38tpeSDwtJ3Ev9P1Rn3gT%2BoxK49hr2%2BzQ4YCCQO0%2B7t4277X1boEt74u>

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

b0XDa7Hzd733vqc0e4mXlb7E8KstTyTnEjlnZ-  
nwwQ3a7V%2B5pmGallhQ81akUhqod%2BX4KBGLJl9drP92tmgoCkTzJli%2BP36wDC2w5sK7%  
2Fo%2BCr55vw6EKWBU%2BuoocXPfpt5SywJcfV3fyxM7vcKyeK-  
nBr7NrHic978GPvWoLFveSo5h9iRc4Zmt1ya%2FNWsaH4Xzjx6nwb6ix1aswRqjh1FBRtutdPpWK  
myl8Weiqmh6oetoQK2anBY%2F3k9dyyYW9BprM7KfRz1vba3OOXp4mx0eHj62QTGmp3zypoz%  
2BI41e9u8v5t2v0w3LU8l5xC%2F56LatX1pO%2FR9Hu9aP1tUURDltZozLQeKToz-  
BVxB0Yjh%2F0VQAgQVVKY0zc5kDEFBSvGUD%2BXT3SZtbc2g35FyVGuo6BuD-  
mVTld2HVSB4cJEkQBAEQODDPxMz-  
lQZlwi8VB3UZJGTWCo4CWEUOanDSgv%2Bp%2Fh7LxQMm54GviOlojk2aeFTGyCsMGo1WUl-  
BGvsQAjQ2c%2BXx9l55zxG1bzW9fMGCurYoxVWBJVGCacrbBmc%2FVvEJy-  
wZt1EiFjix%2B6JFMUf62nakr8HAAA%3D

db, borAdat[db].mennyiseg, borAdat[db].ar, j = 1, tomb[1..j], maxertek = borAdat[1].mennyiseg	
i=1..db	
v=2..i	
	borAdat[v].mennyiseg > maxertek
maxertek = borAdat[v].mennyiseg tomb[j] = v j++	
-	
