

Neptun kód: **F8U9I2**
 Beadás verziószáma: 1.

Név: **Restye János Barnabás**

Feladat

Időjárás előrejelzés

**

Település valamikor maximális hőmérséklettel

A meteorológiai intézet az ország N településére adott M napos időjárás előrejelzést, az adott településen az adott napra várt legmagasabb hőmérsékletet.

Készíts programot, amely megadja azokat a településeket, amelyeken előfordul valamelyik napi előrejelzések maximuma!

Bemenet

A *standard bemenet* első sorában a települések száma ($1 \leq N \leq 1000$) és a napok száma ($1 \leq M \leq 1000$) van. Az ezt követő N sorban az egyes napokra jósolt M hőmérséklet értéke található ($-50 \leq H_{i,j} \leq 50$).

Kimenet

A *standard kimenet* első sorába azon települések T számát kell kiírni, amelyeken előfordul valamelyik napi előrejelzések maximuma! Ezt kövesse ezen települések sorszáma, növekvő sorrendben!

Példa

Bemenet

```
3 5
10 15 12 10 10
11 11 11 11 20
12 16 16 16 20
```

Kimenet

```
2 2 3
```

Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

Specifikáció

1 Specification
Insert:

€

∇

∃

≠

≤

≥

```

Be: n ∈ N, m ∈ N, ho ∈ N[1..n, 1..m]
Sa: maxho ∈ N
Ki: db ∈ N, y ∈ N[1..] // n * m
Fv: bennevan: N → L,
    bennevan(sor) = VAN(i = 1..m, ho[sor, i] = maxho)
Fv: telepulesmax: N → N x N,
    telepulesmax(telep) = MAX(i = 1..m, ho[telep, i])
Ef: n ≥ 1 and m ≥ 1
Uf: (, maxho) = MAX(i = 1..n, telepulesmax(i).2) and
    (db, y) = KIVÁLOGAT(i = 1..n, bennevan(i), i)

```

2 Data

+

-

1

2

3

Evaluate

```

n: 3
m: 5
ho: [
  [10, 15, 12, 10, 10],
  [11, 11, 11, 11, 20],
  [12, 16, 16, 16, 20]
]
maxho: 20
db: 2
y: [2, 3]

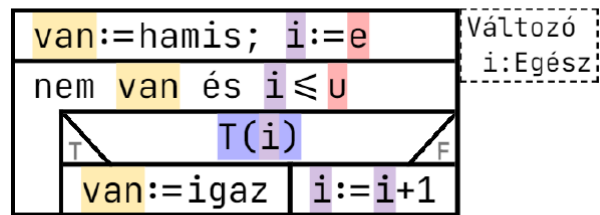
```

Sablon

Specifikáció

Be: $e \in \mathbb{Z}$, $u \in \mathbb{Z}$
 Ki: $van \in \mathbb{L}$
 Ef: -
 Uf: $van = \exists i \in [e..u] : (T(i))$
 Rövidítve:
 Uf: $van = VAN(i=e..u, T(i))$

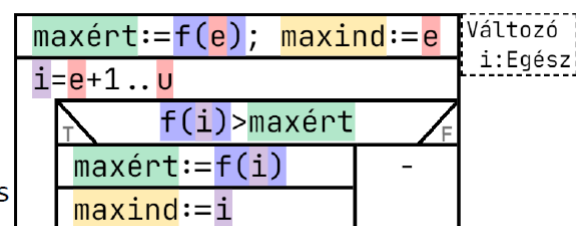
Algoritmus



Specifikáció

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

Algoritmus

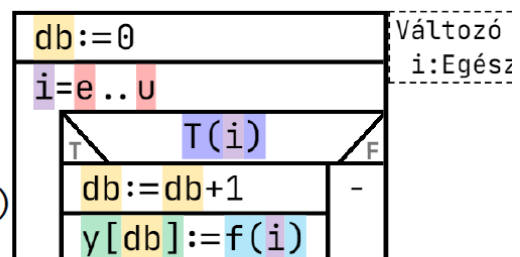


Rövidítve:
 Uf: $(maxind, maxért) = MAX(i=e..u, f(i))$

Specifikáció

Be: $e \in \mathbb{Z}$, $u \in \mathbb{Z}$
 Ki: $db \in \mathbb{N}$, $y \in \mathbb{H}[1..db]$
 Ef: -
 Uf: $db = DARAB(i=e..u, T(i))$ és
 $\forall i \in [1..db] : ($
 $\exists j \in [e..u] : T(j) \text{ és } y[i] = f(j))$
 és $y \subseteq (f(e), f(e+1), \dots, f(u))$

Algoritmus



Rövidítve:
 Uf: $(db, y) = KIVÁLOGAT(i=e..u, T(i), f(i))$

Visszavezetés

telepulesmax(telep)		max		bennevan(sor)		kivalogatas	
maxert	maxert	maxert	maxho	van	van	db	db
maxind	maxind	maxind	,	i	i	i	i
i	i	i	i	e		1	1
e		1	e	u	m	u	n
u	m	u	n	T(i)	ho[sor, i] = maxho	T(i)	bennevan(i)
f(i)	ho[telep, i]	f(i)	telepulesmax(i).2			y	y

Algoritmus

