Megoldás a „Település valamikor maximális hőmérséklettel” beadandó 1. fázishoz

Neptun kód: **F8U9I2** Név: **Restye János Barnabás**
Beadás verziószáma: 2.
Segédfüggvény törlése, 2 max egyesítése sorfolytonosság használatával

## 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 ($1 \le N \le 1000$) és a napok száma ($1 \le M \le 1000$) van. Az ezt követő N sorban az egyes napokra jósolt M hőmérséklet értéke található ($-50 \le H_{i,j} \le 50$).

#### 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

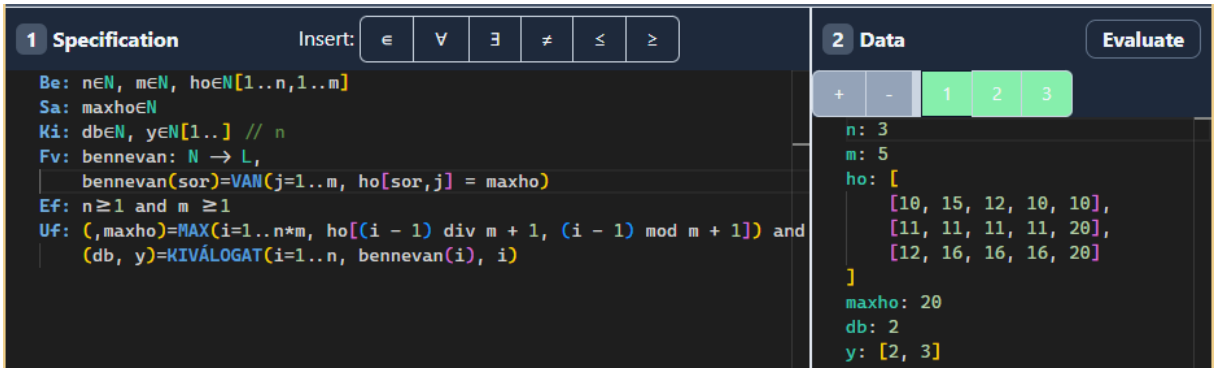
```
2 2 3
```

#### Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

## Specifikáció

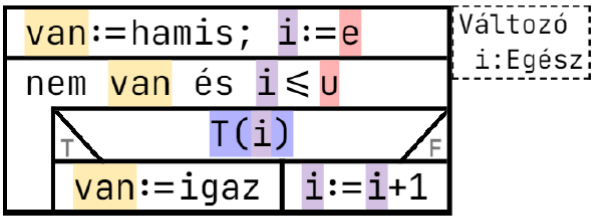


## Sablon

### Specifikáció

Be: e∈Z, u∈Z
Ki: van∈L
Ef: -
Uf: van=∃i∈[e..u]:(T(i))
Rövidítve:
Uf: van=**VAN**(i=e..u,T(i))

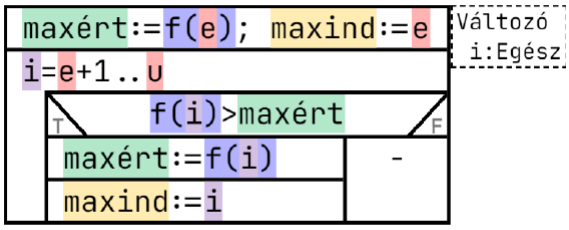
### Algoritmus




### Specifikáció

Be: e∈Z, u∈Z
Ki: maxind∈Z, maxért∈H
Ef: e<=u
Uf: maxind∈[e..u] és
∀i∈[e..u]:(f(maxind)>=f(i)) és
maxért=f(maxind)
Rövidítve:
Uf: (maxind,maxért)=**MAX**(i=e..u, f(i))

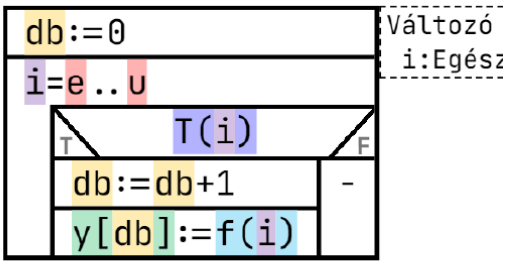
### Algoritmus




### Specifikáció

Be: e∈Z, u∈Z
Ki: db∈N, y∈H[1..db]
Ef: -
Uf: db=**DARAB**(i=e..u,T(i)) és
∀i∈[1..db]:(
∃j∈[e..u]:T(j) és y[i]=f(j))
és y⊆(f(e),f(e+1),...,f(u))
Rövidítve:
Uf: (db,y)=**KIVÁLOGAT**(i=e..u,T(i),f(i))

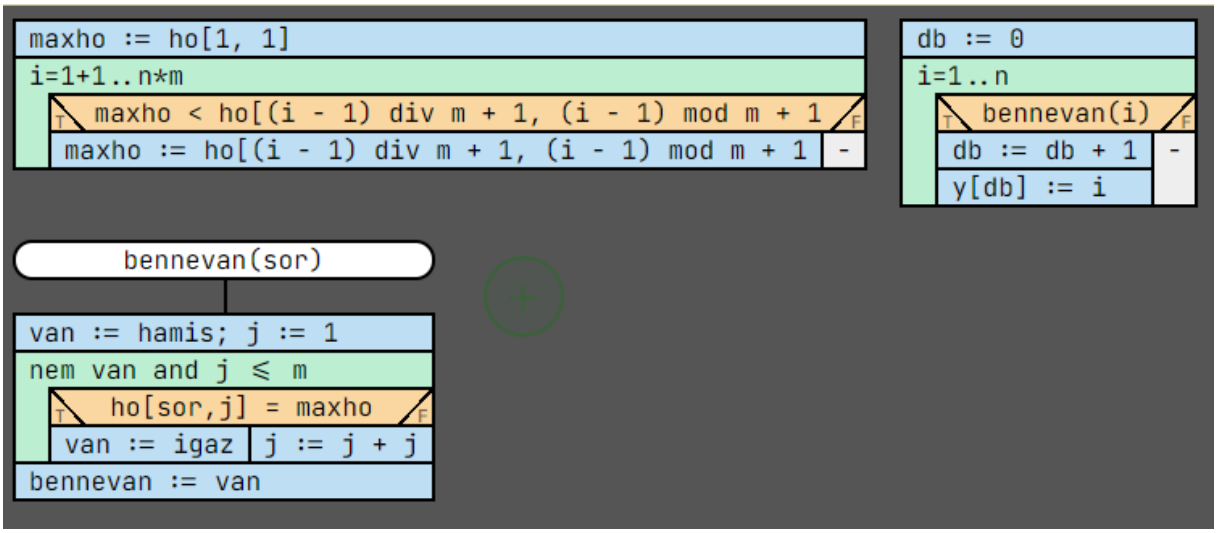
### Algoritmus




## Visszavezetés

| max | | bennevan(sor) | | kivalogatas | |
|---|---|---|---|---|---|
| maxert | maxho | van | van | db | db |
| maxind | , | i | j | i | i |
| i | i | e | 1 | e | 1 |
| e | 1 | u | m | u | n |
| u | n * m | T(i) | ho[sor, j] = maxho | T(i) | bennevan(i) |
| f(i) | ho[(i - 1) div m + 1, (i - 1) mod m + 1] | | | y | y |

## Algoritmus


maxho := ho[1, 1]
i=1+1..n*m
maxho < ho[(i - 1) div m + 1, (i - 1) mod m + 1
maxho := ho[(i - 1) div m + 1, (i - 1) mod m + 1
-
db := 0
i=1..n
bennevan(i)
db := db + 1
y[db] := i
-
bennevan(sor)
van := hamis; j := 1
nem van and j ⩽ m
ho[sor,j] = maxho
van := igaz
j := j + j
bennevan := van