Neptun kód:	**AZXX1Z**	Név:	Soós Csaba
Beadás verziószáma:	1.		

Feladat

Programozási tételek összeépítése *

Új őrség küldése a Kínai Nagy Falra

A Kínai Nagy Falon N őrhelyet létesítettek. Közülük azonban csak M helyen van őrség. Két szomszédos őrhely közötti fal őrzött, ha legalább az egyik végén van őrség.

Készíts programot, amely megadja, hogy minimum hány helyre kell még őrséget küldeni, hogy minden fal őrzött legyen!

Bemenet

A *standard bemenet* első sorában az őrhelyek száma ($1 \leq N \leq 100$) és az őrségek száma ($1 \leq M \leq N$) van, egy szóközzel elválasztva. A következő M sor az őrségek leírását tartalmazza, közülük az i-edik annak az őrhelynek a sorszáma, ahol az i-edik őrség van. Tudjuk, hogy minden helyen legfeljebb 1 őrség van.

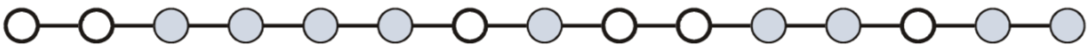
Kimenet

A *standard kimenet* egyetlen sorába egyetlen egész számot kell írni: az új őrségek minimális számát, amivel elérhető, hogy minden fal őrzött legyen!

Példa

Bemenet

```
15 9
6
3
12
11
4
5
8
15
14
```



Kimenet

```
2
```

Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

Specifikáció

Be: n∈N, m∈N, lista∈N[1..m]

Sa: orsegek∈N[1..n], orsegszamok∈N[1..m], kul∈N[1..m], hiany∈N[1..db], db∈N

Ki: uj∈N

Ef: (1<=n és n<=100) és (1<=m és m<=n)

Uf: ∀i∈[1..m]: (orsegek[lista[i]] = 1) és

orsegszamok = KIVÁLOGAT(i=1..n, orsegek[i] != 0, i).2 és

kul[1] = orsegszamok[1] - 1 és

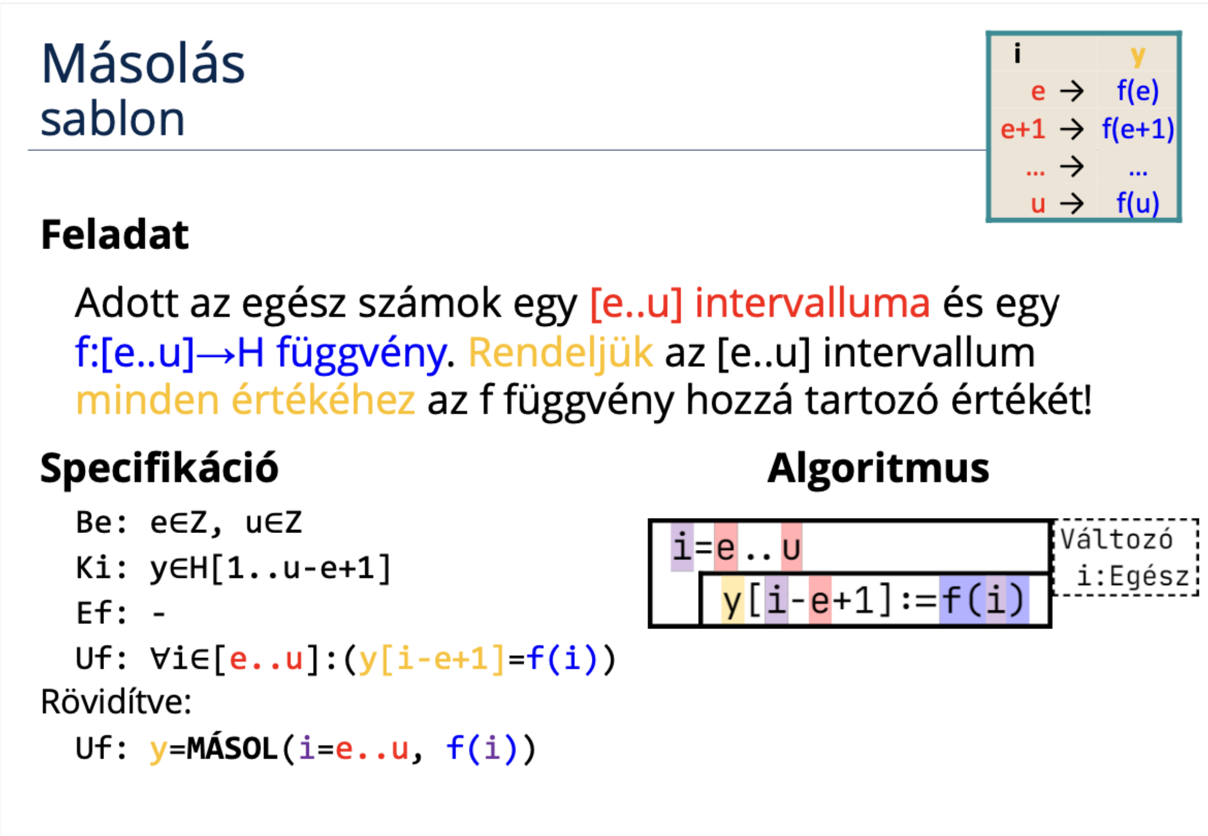
∀i∈[2..m]: (kul[i] = orsegszamok[i] - orsegszamok[i-1] - 1) és

db = DARAB(i=1..m, kul[i]>1) és

hiany = KIVÁLOGAT(i=1..m, kul[i]>1, kul[i]).2 és

uj = SZUMMA(i=1..db, hiany[i]/2)

Sablon



Megszámolás
sablon

i	T(i)	érték
e	IGAZ	1 +
e+1	HAMIS	0 +
...	HAMIS	0 +
u	IGAZ	1
		=
	db=	2

Feladat

Adott az egész számok egy [e..u] intervalluma és egy T:[e..u]→Logikai feltétel. Határozzuk meg, hogy az [e..u] intervallumon a T feltétel hányszor veszi fel az igaz értéket!

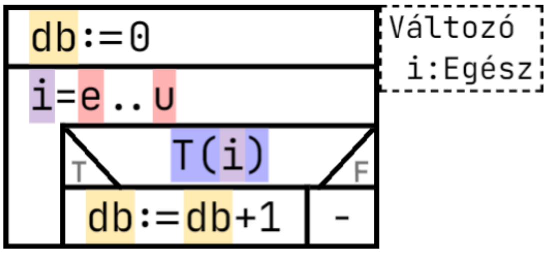
Specifikáció

```
Be: e∈Z, u∈Z
Ki: db∈N
Ef: -
Uf: db=SZUMMA(i=e..u, 1, T(i))
```

Rövidítve:

```
Uf: db=DARAB(i=e..u, T(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]→Logikai feltétel és egy f:[e..u]→H függvény. Határozzuk meg az f függvény az [e..u] intervallum azon értékeinél felvett értékeit, amelyekre a T feltétel teljesül!

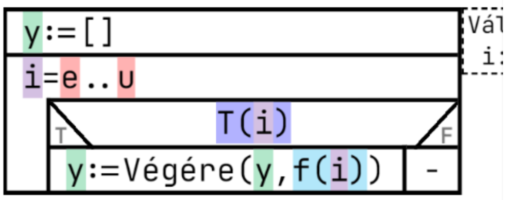
Specifikáció

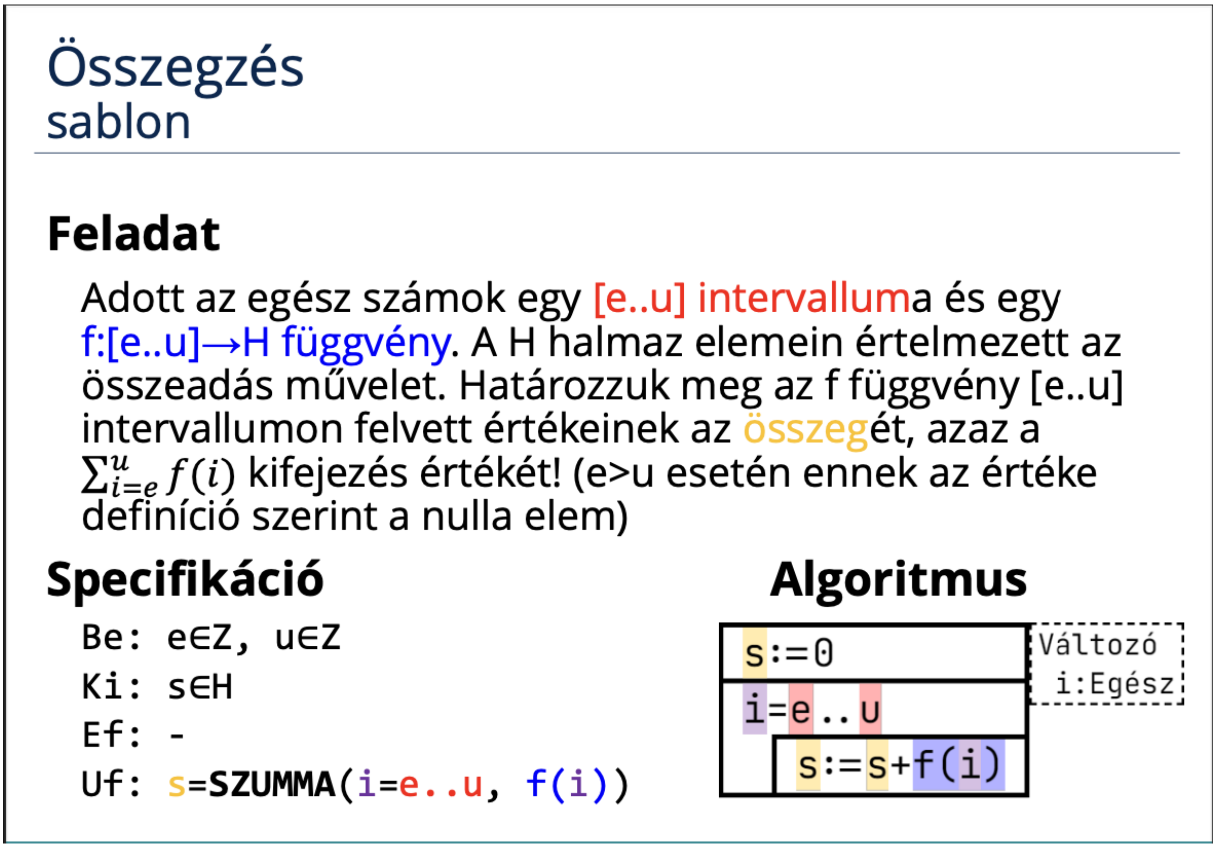
```
Be: e∈Z, u∈Z
Ki: y∈H[1..]
Ef: -
Uf: ∀i∈[1..hossz(y)]:(
        ∃j∈[e..u]:T(j) és y[i]=f(j))
  és y⊆(f(e),f(e+1),...,f(u))
```

Rövidítve:

```
Uf: (,y)=KIVÁLOGAT(i=e..u,T(i),f(i))
```

Algoritmus





Visszavezetés

MÁSOL: ∀i∈[e..u]:(y[i-e+1]=f(i))

e..u ~ 1..m

y[i-e+1] ~ orsegek[lista[i]]

f(i) ~ 1

KIVÁLOGAT:

y ~ orsegszamok

e..u ~ 1..n

T(i) ~ orsegek[i] != 0

f(i) ~ i

MÁSOL: ∀i∈[e..u]:(y[i-e+1]=f(i))

e..u ~ 2..m

y[i-e+1] ~ kul[i]

f(i) ~ orsegszamok[i] - orsegszamok[i-1] - 1

Megoldás sablon „Új őrség küldése a Kínai Nagy Falra” beadandó 1. fázishoz

DARAB:

db ~ db

e..u ~ 1..m

T(i) ~ kul[i] > 1

KIVÁLOGAT:

y ~ hiany

e..u ~ 1..m

T(i) ~ kul[i]>1

f(i) ~ kul[i]

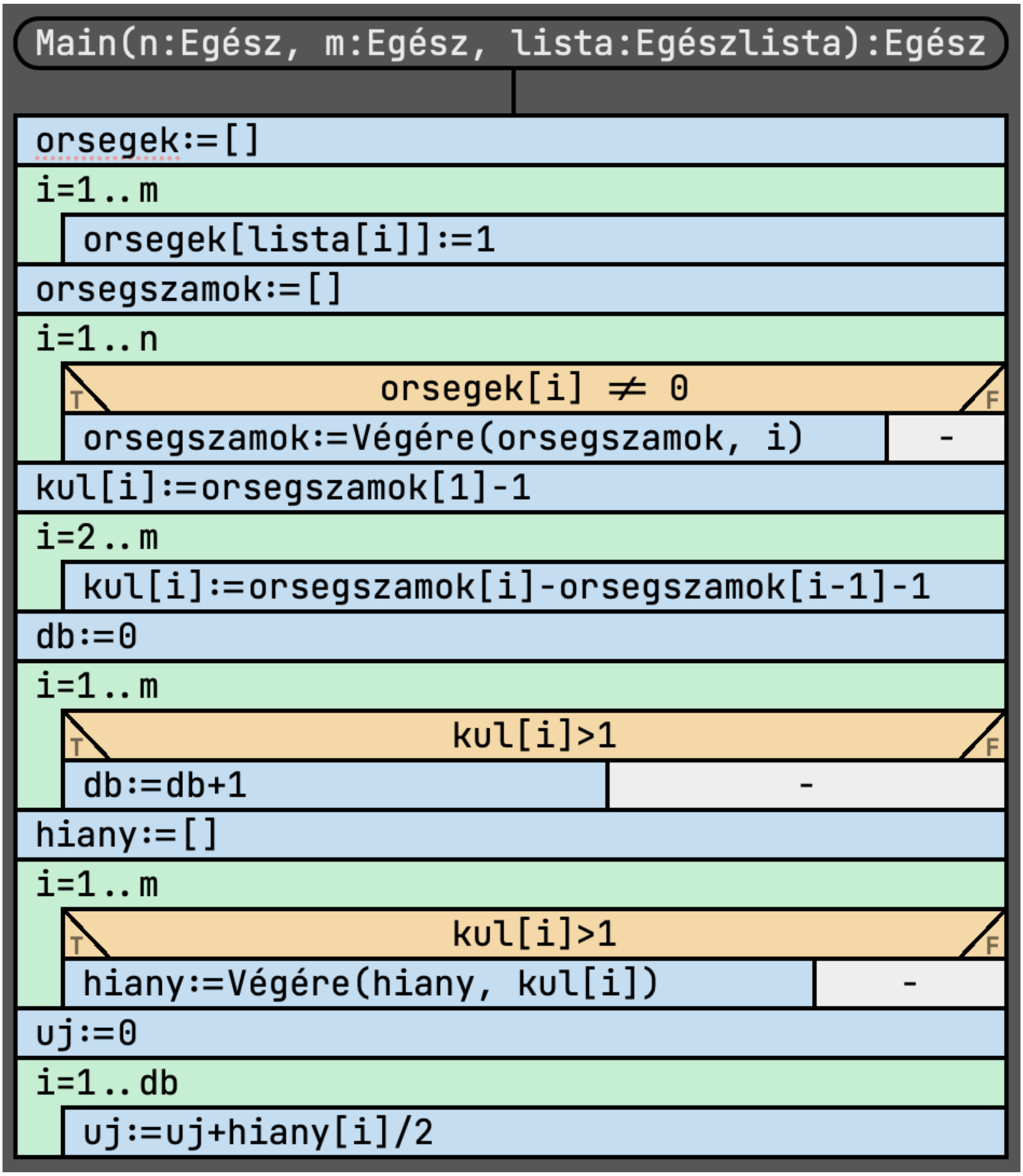
SZUMMA:

s ~ uj

e..u ~ 1..db

f(i) ~ hiany[i]/2

Algoritmus



Git repo:

https://github.com/csabisoos/elte/tree/main/1.%20felev/programozas%20gyak/beadando_2