PROGRAMOZÁS

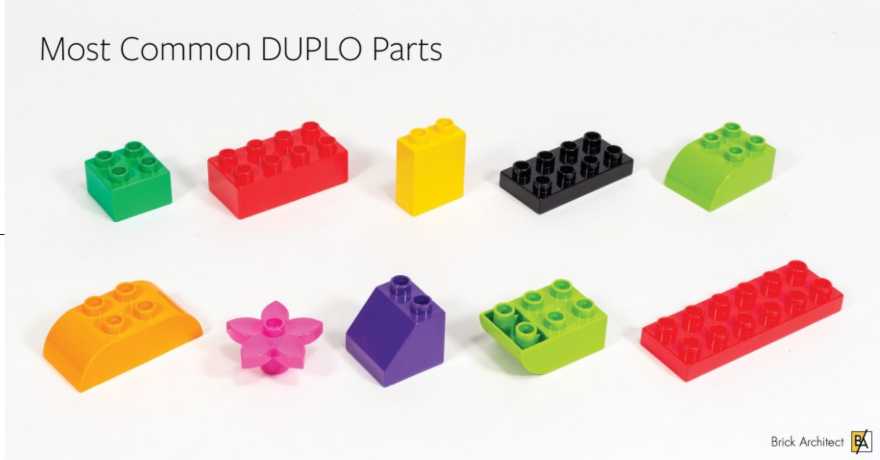
Visszavezetési esetek

Horváth Győző

Ismétlés

Programozási minták

1. Összegzés
2. Megszámolás
3. Maximumkiválasztás
   a. Minimumkiválasztás
4. Feltételes maximumkeresés
5. Keresés
6. Eldöntés
   a. Mind eldöntés
7. Kiválasztás
8. Másolás
9. Kiválogatás



Programozási minták

Összegzés

i		f(i)
e	→	f(e)
		+
e+1	→	f(e+1)
		+
e+2	→	f(e+2)
		+
…	→	…
		+
u-2	→	f(u-2)
		+
u-1	→	f(u-1)
		+
u	→	f(u)
		=
		s

Megszámolás

i		T(i)	érték
e	→	IGAZ	1
			+
e+1	→	HAMIS	0
			+
e+2	→	HAMIS	0
			+
…	→	…	…
			+
u-2	→	IGAZ	1
			+
u-1	→	IGAZ	1
			+
u	→	HAMIS	0
			=
			db

Maximum kiválasztás

i		f(i)
e	→	f(e)
e+1	→	f(e+1)
e+2	→	f(e+2)
…	→	…
u-2	→	f(u-2)
u-1	→	f(u-1)
u	→	f(u)

maxind, maxért

Feltételes maximumkeresés

i		T(i)	f(i)
e	→	HAMIS	f(e)
e+1	→	IGAZ	f(e+1)
e+2	→	IGAZ	f(e+2)
…	→		…
u-2	→	HAMIS	f(u-2)
u-1	→	IGAZ	f(u-1)
u	→	HAMIS	f(u)

van, maxind, maxért

Programozási minták

Keresés

i		T(i)
e	→	HAMIS
e+1	→	HAMIS
e+2	→	IGAZ
…	→	…
u-2	→	IGAZ
u-1	→	IGAZ
u	→	HAMIS

van, ind

Eldöntés

i		T(i)
e	→	HAMIS
e+1	→	HAMIS
e+2	→	IGAZ
…	→	…
u-2	→	IGAZ
u-1	→	IGAZ
u	→	HAMIS

van

Kiválasztás

i		T(i)
e	→	HAMIS
e+1	→	HAMIS
e+2	→	IGAZ
…	→	…
u-2	→	IGAZ
u-1	→	IGAZ
u	→	HAMIS

ind

Programozási minták

Másolás

i			f(i)
e	→	1	f(e)
e+1	→	2	f(e+1)
e+2	→	3	f(e+2)
…	→	…	…
u-2	→	u-e-1	f(u-2)
u-1	→	u-e	f(u-1)
u	→	u-e+1	f(u)

y

Kiválogatás

i		T(i)	f(i)			y
e	→	HAMIS	f(e)		1	f(e+1)
e+1	→	IGAZ	f(e+1)		2	f(e+2)
e+2	→	IGAZ	f(e+2)	db=	3	f(u-1)
…	→		…			
u-2	→	HAMIS	f(u-2)			
u-1	→	IGAZ	f(u-1)			
u	→	HAMIS	f(u)			

db, y

Visszavezetési esetek

Problémafelvetés

Eddig az utófeltételt hasonlítottuk össze, noha a programozási minta sablonja adott adatokra és ezeken értelmezett függvényekre mond valamit.

Mi van akkor, ha a mintafeladat és a konkrét feladat adatai eltérnek, pl. hiányzik vagy más jellegű adat van? Abban az esetben miért vezet helyes működésre a dolog?

Feladatsablon
(mintafeladat)

```
Be: e∈Z, u∈Z
Ki: van∈L, ind∈Z
Ef: -
Uf: (van,ind)=KERES(i=e..u,
                    T(i))
```

???

Előzőnél gyorsabb vonat
(konkrét feladat)

```
Be: n∈N, midő∈N[1..n]
Ki: van∈L, melyik∈N
Ef: -
Uf: (van,melyik)=KERES(i=2..n,
          midő[i]<midő[i-1])
```

```
ind  ~ melyik
```

Rövidítés = helyettesítés

- A rövidített utófeltétel az eredeti hosszabb (és logikailag helyes) változatot helyettesíti
- A konkrét feladatot felírva csak akkor írhatjuk fel a rövidített formátumot, ha az minden elemében megfeleltethető a hosszabb formátumnak.
- Ekkor a visszavezetés során előállt algoritmus is biztosan helyes megoldása lesz a feladatnak.

```
Uf: db=SZUMMA(i=e..u, 1, T(i))
```

⇕

```
Uf: db=DARAB(i=e..u, T(i))
```

Természetes visszavezetés
amikor minden stimmel

Hány páros szám van a és b intervallumban?

Feladatsablon

(mintafeladat)

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

Uf: db=DARAB(i=e..u, T(i))
```

Páros számok száma

(konkrét feladat)

```
Be: a∈Z, b∈Z
Ki: pdb∈N
Ef: -
Uf: pdb=SZUMMA(i=a..b,1,i mod 2=0)
```

Természetes visszavezetés
amikor minden stimmel

Hány páros szám van a és b intervallumban?

Feladatsablon

(mintafeladat)

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

Uf:  db=DARAB(i=e..u, T(i))
```

Páros számok száma

(konkrét feladat)

```
Be: a∈Z, b∈Z
Ki: pdb∈N
Ef: -
Uf: pdb=SZUMMA(i=a..b,1,i mod 2=0)
```

Minden stimmel = az átnevezéseken kívül teljesen megegyezik
- adatok, futóindexek neve
- általános (jelentés nélküli) kifejezések, mint f(i), T(i)
- állandó értékű kifejezések, mint intervallum határ

Természetes visszavezetés
amikor minden stimmel

Hány páros szám van a és b intervallumban?

Feladatsablon
(mintafeladat)

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

Uf: db=DARAB(i=e..u, T(i))
```

Páros számok száma
(konkrét feladat)

```
Be: a∈Z, b∈Z
Ki: pdb∈N
Ef: -
Uf: pdb=SZUMMA(i=a..b,1,i mod 2=0)

Uf: pdb=DARAB(i=a..b, i mod 2=0)
```

Minden stimmel = az átnevezéseken kívül teljesen megegyezik
- adatok, futóindexek neve
- általános (jelentés nélküli) kifejezések, mint f(i), T(i)
- állandó értékű kifejezések, mint intervallum határ

Általános visszavezetés
szigorúbb előfeltétel

Hány prímszám van a és b intervallumban?

Feladatsablon

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

Uf: db=DARAB(i=e..u, T(i))
```

i		T(i)	érték
e	→	IGAZ	1 +
e+1	→	HAMIS	0 +
e+2	→	HAMIS	0 +
…	→	…	… +
u-2	→	IGAZ	1 +
u-1	→	IGAZ	1 +
u	→	HAMIS	0
			=
			db

Prímszámok száma

```
Be: a∈Z, b∈Z
Ki: pdb∈N
Ef: a>0
Uf: pdb=SZUMMA(i=a..b, 1, prím(i))

Uf: pdb=DARAB(i=a..b, prím(i))
```

A konkrét feladat előfeltétele lehet szigorúbb a mintáénál. Ha sok adatra helyes, akkor nyilván a szűkítettre is helyes lesz.

i		T(i)	érték
e	→	IGAZ	1 +
e+1	→	HAMIS	0 +
e+2	→	HAMIS	0 +
…	→	…	… +
u-2	→	IGAZ	1 +
u-1	→	IGAZ	1 +
u	→	HAMIS	0
			=
			db

Általános visszavezetés
gyengébb utófeltétel

Adj meg egy prímszámot az a és b intervallumban!

Feladatsablon

```
Be: e∈Z, u∈Z
Ki: van∈L, ind∈Z
Ef: -
Uf: van=∃i∈[e..u]:(T(i)) és
     van->(ind∈[e..u] és T(ind) és
        ∀i∈[e..ind-1]:(nem T(i)))

Uf: (van,ind)=KERES(i=e..u,T(i))
```

Prímszám keresése

```
Be: a∈Z, b∈Z
Ki: van∈L, p∈N
Ef: a>0
Uf: van=∃i∈[a..b]:(prím(i)) és
     van->(p∈[a..b] és prím(p))
```

i		T(i)
e	→	HAMIS
e+1	→	HAMIS
e+2	→	IGAZ
…	→	…
u-2	→	IGAZ
u-1	→	IGAZ
u	→	HAMIS

A konkrét feladat utófeltétele lehet gyengébb a mintáénál. Ha a feladat több megoldást is elfogad, akkor nyilván a minta szűkített megoldása is helyes lesz. → Szűkítsük le a feladatot!

i		T(i)
e	→	HAMIS
e+1	→	HAMIS
e+2	→	IGAZ
…	→	…
u-2	→	IGAZ
u-1	→	IGAZ
u	→	HAMIS

Általános visszavezetés
gyengébb utófeltétel

Adj meg egy prímszámot az a és b intervallumban!

Feladatsablon

```
Be: e∈Z, u∈Z
Ki: van∈L, ind∈Z
Ef: -
Uf: van=∃i∈[e..u]:(T(i)) és
     van->(ind∈[e..u] és T(ind) és
        ∀i∈[e..ind-1]:(nem T(i)))

Uf: (van,ind)=KERES(i=e..u,T(i))
```

Prímszám keresése

```
Be: a∈Z, b∈Z
Ki: van∈L, p∈N
Ef: a>0
Uf: van=∃i∈[a..b]:(prím(i)) és
     van->(p∈[a..b] és prím(p) és
        ∀i∈[e..p-1]:(nem prím(i)))

Uf: (van,p)=KERES(i=a..b,prím(i))
```

i		T(i)
e	→	HAMIS
e+1	→	HAMIS
e+2	→	IGAZ
…	→	…
u-2	→	IGAZ
u-1	→	IGAZ
u	→	HAMIS

A konkrét feladat utófeltétele lehet gyengébb a mintáénál. Ha a feladat több megoldást is elfogad, akkor nyilván a minta szűkített megoldása is helyes lesz. → Szűkítsük le a feladatot!

i		T(i)
e	→	HAMIS
e+1	→	HAMIS
e+2	→	IGAZ
…	→	…
u-2	→	IGAZ
u-1	→	IGAZ
u	→	HAMIS

Alteres visszavezetés
kevesebb kimeneti adat

Az a és b intervallumbeli egész értékeken hol veszi fel a sin(x) függvény a legnagyobb értékét?

Feladatsablon

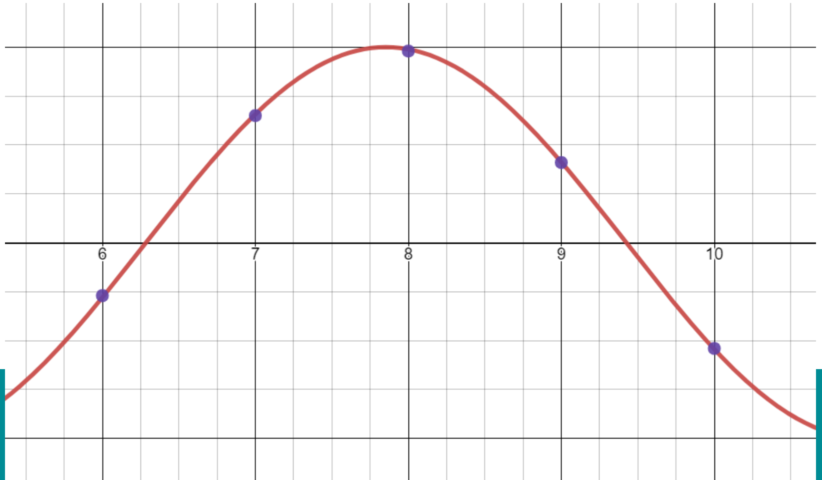
```
Be: e∈Z, u∈Z

Ki: maxind∈Z, maxért∈H
Ef: -
Uf: maxind∈[e..u] és
    ∀i∈[e..u]:(f(maxind)>=f(i)) és
    maxért=f(maxind)
```

↓

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

sin(x) maximuma

```
Be: a∈Z, b∈Z

Ki: hol∈Z
Ef: -
Uf: hol∈[a..b] és
    ∀i∈[a..b]:(sin(hol)>=sin(i))
```



Alteres visszavezetés
kevesebb kimeneti adat

Az a és b intervallumbeli egész értékeken hol veszi fel a sin(x) függvény a legnagyobb értékét?

Feladatsablon

```
Be: e∈Z, u∈Z
Sa: maxért∈H
Ki: maxind∈Z
Ef: -
Uf: maxind∈[e..u] és
    ∀i∈[e..u]:(f(maxind)>=f(i)) és
    maxért=f(maxind)

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

sin(x) maximuma

```
Be: a∈Z, b∈Z
Sa: maxért∈R
Ki: hol∈Z
Ef: -
Uf: hol∈[a..b] és
    ∀i∈[a..b]:(sin(hol)>=sin(i)) és
    maxért=f(hol)

Uf: (hol,maxért)=
                  MAX(i=a..b,sin(i))
```

A mintafeladat kimeneti adatait **segédadat**tá lehet minősíteni.

→ szigorúbb lesz az utófeltétel, hiszen a megmaradt kimeneti adaton túl másra is tesz megszorítást, de ez nem baj (ld. előzőleg)

→ segédváltozó az algoritmusban

Ezzel párhuzamosan a konkrét feladatba is segédadatot kell bevezetni és szigorítani kell az utófeltételét.

Alteres visszavezetés
kevesebb kimeneti adat

Az a és b intervallumbeli egész értékeken hol veszi fel a sin(x) függvény a legnagyobb értékét?

Feladatsablon

```
Be: e∈Z, u∈Z
Sa: maxért∈H
Ki: maxind∈Z
Ef: -
Uf: maxind∈[e..u] és
    ∀i∈[e..u]:(f(maxind)>=f(i)) és
    maxért=f(maxind)
```

⬇

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

sin(x) maximuma

```
Be: a∈Z, b∈Z

Ki: hol∈Z
Ef: -
```

maxind, maxért	~	hol, maxért
e..u	~	a..b
f(i)	~	sin(i)

```
Uf: (hol,)=
              MAX(i=a..b,sin(i))
```

A konkrét feladatnak nincs szüksége a segédadatra, csak azért vezettük be, hogy a rövidítés és visszavezetés alkalmazható legyen. Végső soron teljesen ki is hagyható a specifikációból.
A visszavezetési táblázatban ugyanolyan nevűnek tekintjük.

Alteres visszavezetés
kevesebb kimeneti adat

Az a és b intervallumbeli egész értékeken hol veszi fel a sin(x) függvény a legnagyobb értékét?

Feladatsablon

```
Be: e∈Z, u∈Z
Sa: maxért∈H
Ki: maxind∈Z
Ef: -
Uf: (maxind,maxért)=
        MAX(i=e..u,f(i))
```

sin(x) maximuma

```
Be: a∈Z, b∈Z

Ki: hol∈Z
Ef: -
Uf: (hol,)=
        MAX(i=a..b,sin(i))
```

maxind, maxért	~	hol, maxért
e..u	~	a..b
f(i)	~	sin(i)

```
maxért:=f(e); maxind:=e
i=e+1..u
  f(i)>maxért   (T / F)
    T: maxért:=f(i)
       maxind:=i
    F: -
```

Változó
i:Egész
maxért:H

```
maxért:=sin(a); hol:=a
i=a+1..b
  sin(i)>maxért   (T / F)
    T: maxért:=sin(i)
       hol:=i
    F: -
```

Változó
i:Egész,
maxért:Valós

A kihagyott adat segédváltozó lesz az algoritmusban.

Alteres visszavezetés
példa: eldöntés

Eldöntés

Keresés

```
Be: e∈Z, u∈Z
Sa: ind∈Z
Ki: van∈L, ind∈Z
Ef: -
Uf: (van,ind)=KERES(i=e..u,T(i))
```

Eldöntés

```
Be: e∈Z, u∈Z
Sa: ind∈Z
Ki: van∈L
Ef: -
Uf: (van,ind)=KERES(i=e..u,T(i))
Uf: (van,)=KERES(i=e..u,T(i))
Uf: van=VAN(i=e..u,T(i))
```

```
ind:=e
ind≤u és nem T(ind)
    ind:=ind+1
van:=ind≤u
```

```
Változó
  ind:Egész
```

```
i:=e
i≤u és nem T(i)
    i:=i+1
van:=i≤u
```

```
Változó
  i:Egész
```


A kihagyott adat segédváltozó lesz az algoritmusban.

Alteres visszavezetés
kevesebb bemeneti adat

Adjuk meg egy természetes szám valódi osztóinak számát!

Feladatsablon

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

⬇

```
Uf: db=DARAB(i=e..u, T(i))
```

Valódi osztók száma

```
Be:        n∈N
Ki: db∈N
Ef: -
Uf: db=SZUMMA(i=2..n div 2,1,i|n)
```

Ha a mintafeladat egy olyan bemeneti adata hiányzik a konkrét feladatból, amelyik nem változtatja értékét, akkor ezt helyettesíthetjük egy olyan adattal, amelyik értéke állandó (konstans). Ettől a feladat nem változik meg. Az értéket az előfeltételben határozzuk meg, azaz a konkrét feladat szigorítja azt. A bevezetett konstans értékű adatnak megfelelő változókat segédváltozóknak tekinthetjük.

Alteres visszavezetés
kevesebb bemeneti adat

Adjuk meg egy természetes szám valódi osztóinak számát!

Feladatsablon

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

Uf: db=DARAB(i=e..u, T(i))
```

Valódi osztók száma

```
Be: e∈N, n∈N
Ki: db∈N
Ef: e=2
Uf: db=SZUMMA(i=e..n div 2,1,i|n)

Uf: db=DARAB(i=e..n div 2, i|n)
```

Ha a mintafeladat egy olyan bemeneti adata hiányzik a konkrét feladatból, amelyik nem változtatja értékét, akkor ezt helyettesíthetjük egy olyan adattal, amelyik értéke állandó (konstans). Ettől a feladat nem változik meg. Az értéket az előfeltételben határozzuk meg, azaz a konkrét feladat szigorítja azt. A bevezetett konstans értékű adatnak megfelelő változókat segédváltozóknak tekinthetjük.

Paraméteres visszavezetés
több bemeneti adat

Osztja-e k az a és b közötti számok valamelyikét?

Feladatsablon

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

Uf: van=**VAN**(i=e..u, T(i))

Valódi osztók száma

Be: a∈Z, b∈Z, k∈N
Ki: van∈L
Ef: -
Uf: van=∃i∈[a..b]:(k|i)

Ha a konkrét feladat olyan plusz bemeneti adatot is tartalmaz, amelynek értéke állandó, akkor

1. rögzítsük az értékét (konstans, nem kell feltüntetni az adatok között)
2. a visszavezetés így adott érték mellett megtehető, hiszen az adatok egyeznek
3. ez k minden lehetséges értéke mellett megtehető, ezért ezt olyan adatként jelezzük, amely a program elején felvesz egy állandó értéket

Paraméteres visszavezetés
több bemeneti adat

Osztja-e k az a és b közötti számok valamelyikét?

Feladatsablon

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

Uf: van=**VAN**(i=e..u, T(i))

Valódi osztók száma

Be: a∈Z, b∈Z
Ki: van∈L
Ef: -
Uf: van=∃i∈[a..b]:(13|i)

k=13

Uf: van=**VAN**(i=a..b, 13|i)

Ha a konkrét feladat olyan plusz bemeneti adatot is tartalmaz, amelynek értéke állandó, akkor

1. rögzítsük az értékét (konstans, nem kell feltüntetni az adatok között)
2. a visszavezetés így adott érték mellett megtehető, hiszen az adatok egyeznek
3. ez k minden lehetséges értéke mellett megtehető, ezért ezt olyan adatként jelezzük, amely a program elején felvesz egy állandó értéket

Paraméteres visszavezetés
több bemeneti adat

Osztja-e k az a és b közötti számok valamelyikét?

Feladatsablon

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

```
Uf: van=VAN(i=e..u, T(i))
```

Valódi osztók száma

```
Be: a∈Z, b∈Z, k∈N
Ki: van∈L
Ef: -
Uf: van=∃i∈[a..b]:(k|i)
```

minden k-ra

```
Uf: van=VAN(i=a..b, k|i)
```

Ha a konkrét feladat olyan plusz bemeneti adatot is tartalmaz, amelynek értéke állandó, akkor

1. rögzítsük az értékét (konstans, nem kell feltüntetni az adatok között)
2. a visszavezetés így adott érték mellett megtehető, hiszen az adatok egyeznek
3. ez k minden lehetséges értéke mellett megtehető, ezért ezt olyan adatként jelezzük, amely a program elején felvesz egy állandó értéket

Példa

Ez miért is tehető meg, amikor nem is hasonlítanak?

Egy hőmérsékletstatisztika alapján add meg a legnagyobb hőmérsékletet!

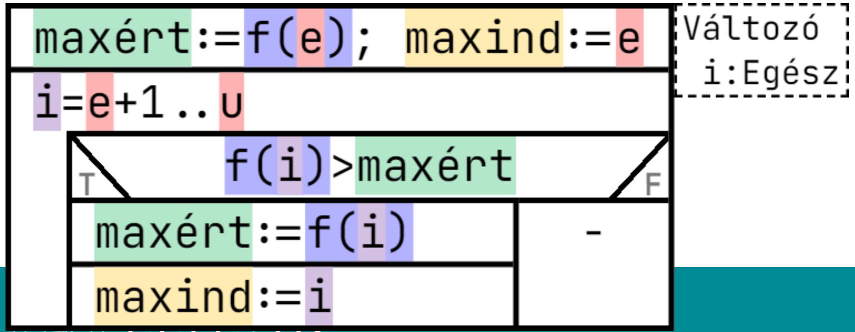
Feladatsablon

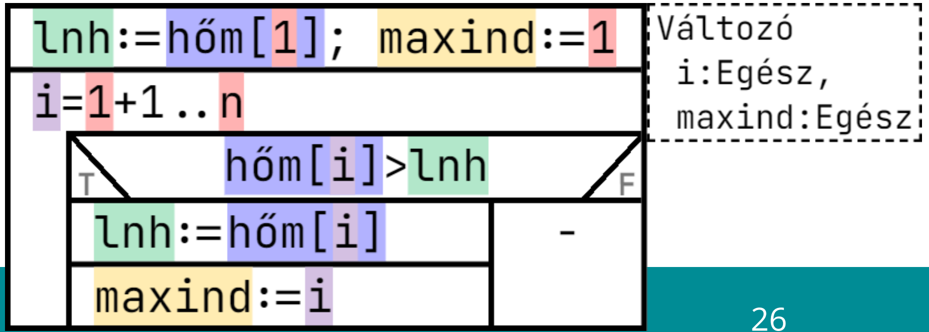
```
Be: e∈Z, u∈Z
Ki: maxind∈Z, maxért∈H
Ef: e<=u

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

Legnagyobb hőmérséklet

```
Be: n∈N, hőm∈R[1..n]
Ki: lnh∈R
Ef: n>0 és
    ∀i∈[1..n]:(-100<=hőm[i]<=100)
Uf: (,lnh)=
              MAX(i=1..n,hőm[i])
```

maxind, maxért	~	maxind, lnh
e..u	~	1..n
f(i)	~	hőm[i]





Példa

Egy hőmérsékletstatisztika alapján add meg a legnagyobb hőmérsékletet!

Legnagyobb hőmérséklet

```
Be:               n∈N, hőm∈R[1..n]
Ki:               lnh∈R
Ef: n>0 és
    ∀i∈[1..n]:(-100<=hőm[i]<=100)
Uf: ∃i∈[1..n]:(lnh=hőm[i]) és
    ∀i∈[1..n]:(lnh>=hőm[i])
```

Ez maximumkiválasztás!

Példa

Egy hőmérsékletstatisztika alapján add meg a legnagyobb hőmérsékletet!

Feladatsablon

```
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)
```

⬇

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

Legnagyobb hőmérséklet

```
Be: n∈N, hőm∈R[1..n]
Ki: lnh∈R
Ef: n>0 és
    ∀i∈[1..n]:(-100<=hőm[i]<=100)
Uf: ∃i∈[1..n]:(lnh=hőm[i]) és
    ∀i∈[1..n]:(lnh>=hőm[i])
```

De ha összehasonlítjuk, akkor maximumkiválasztást csak nyomokban tartalmaz!

Példa

Egy hőmérsékletstatisztika alapján add meg a legnagyobb hőmérsékletet!

Feladatsablon

```
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)

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

Legnagyobb hőmérséklet

```
Be:   n∈N, hőm∈R[1..n]
Ki:   lnh∈R
Ef:  n>0 és
     ∀i∈[1..n]:(-100<=hőm[i]<=100)
Uf:  ∃i∈[1..n]:(lnh=hőm[i]) és
     ∀i∈[1..n]:(lnh>=hőm[i])
```

1. Paraméteres visszavezetés

A tömb olyan értékét nem változtató bemenő adat, amelyet – értékét állandó paraméternek véve – elhagyhatónak tekintünk.

Példa

Egy hőmérsékletstatisztika alapján add meg a legnagyobb hőmérsékletet!

Feladatsablon

```
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)
```

⬇

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

Legnagyobb hőmérséklet

```
Be: e∈N, u∈N, n∈N, hőm∈R[1..n]
Ki: lnh∈R
Ef: n>0 és
    ∀i∈[1..n]:(-100<=hőm[i]<=100)
    és e=1 és u=n és e<=u
Uf: ∃i∈[1..n]:(lnh=hőm[i]) és
    ∀i∈[1..n]:(lnh>=hőm[i])
```

2. Alteres visszavezetés (kevesebb bemeneti adat)

A feladat implicit módon tartalmazza az intervallum határait. Ezek állandó értékek → beemelhetők az adatok közé → előfeltétel

Példa

Egy hőmérsékletstatisztika alapján add meg a legnagyobb hőmérsékletet!

Feladatsablon

```
Be: e∈Z, u∈Z
Sa: maxind∈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)

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

Legnagyobb hőmérséklet

```
Be: e∈N, u∈N, n∈N, hőm∈R[1..n]
Sa: maxind∈Z
Ki:             lnh∈R
Ef: n>0 és
    ∀i∈[1..n]:(-100<=hőm[i]<=100)
    és e=1 és u=n és e<=u
Uf: ∃i∈[1..n]:(lnh=hőm[i]) és
    ∀i∈[1..n]:(lnh>=hőm[i])
```

3. Alteres visszavezetés (kevesebb kimeneti adat)

A mintafeladat kimeneti adata segédadattá avanzsálható, a konkrét feladatban ugyanilyen néven megjelenik segédadat

Példa

Egy hőmérsékletstatisztika alapján add meg a legnagyobb hőmérsékletet!

Feladatsablon

```
Be: e∈Z, u∈Z
Sa: maxind∈Z
Ki:              maxért∈H
Ef: e<=u



Uf: maxind∈[e..u] és
    ∀i∈[e..u]:(f(maxind)>=f(i)) és
    maxért=f(maxind)

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

Legnagyobb hőmérséklet

```
Be: e∈N, u∈N, n∈N, hőm∈R[1..n]
Sa: maxind∈Z
Ki:           lnh∈R
Ef: n>0 és
    ∀i∈[1..n]:(-100<=hőm[i]<=100)
    és e=1 és u=n és e<=u
Uf: ∃i∈[1..n]:(lnh=hőm[i]) és
    ∀i∈[1..n]:(lnh>=hőm[i])
```

4. Általános visszavezetés (szigorúbb előfeltétel)

A konkrét feladat előfeltétele szigorúbb, így szűkíti a megoldás értelmezési tartományát, ami megtehető

Példa

Egy hőmérsékletstatisztika alapján add meg a legnagyobb hőmérsékletet!

Feladatsablon

```
Be: e∈Z, u∈Z
Sa: maxind∈Z
Ki:            maxért∈H
Ef: e<=u



Uf: maxind∈[e..u] és
    ∀i∈[e..u]:(f(maxind)>=f(i)) és
     maxért=f(maxind)

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

Legnagyobb hőmérséklet

```
Be: e∈N, u∈N, n∈N, hőm∈R[1..n]
Sa: maxind∈Z
Ki:             lnh∈R
Ef: n>0 és
    ∀i∈[1..n]:(-100<=hőm[i]<=100)
    és e=1 és u=n és e<=u
Uf: maxind∈[1..n] és
    ∀i∈[1..n]:(hőm[maxind]>=hőm[i])
 és lnh=hőm[maxind]
```

5. Általános visszavezetés (gyengébb utófeltétel)

Utófeltétel átalakítása: a bevezetett segédadattal átfogalmazható, mit tudunk lnh és e,u,hőm kapcsolatáról → szigorítjuk az utófeltételt

Példa

Egy hőmérsékletstatisztika alapján add meg a legnagyobb hőmérsékletet!

Feladatsablon

```
Be: e∈Z, u∈Z
Sa: maxind∈Z
Ki:             maxért∈H
Ef: e<=u



Uf: maxind∈[e..u] és
    ∀i∈[e..u]:(f(maxind  )>=f  (i))
 és maxért=f  (maxind)
```

Legnagyobb hőmérséklet

```
Be: e∈N, u∈N, n∈N, hőm∈R[1..n]
Sa: maxind∈Z
Ki:           lnh   ∈R
Ef: e<=u és e=1 és u=n és
    n>0 és
    ∀i∈[1..n]:(-100<=hőm[i]<=100)
Uf: maxind∈[1..n] és
    ∀i∈[1..n]:(hőm[maxind]>=hőm[i])
 és lnh   =hőm[maxind]
```

Kis átrendezéssel a konkrét feladat megfeleltethető a mintafeladatnak.

Példa

Egy hőmérsékletstatisztika alapján add meg a legnagyobb hőmérsékletet!

Feladatsablon

```
Be: e∈Z, u∈Z
Sa: maxind∈Z
Ki:            maxért∈H
Ef: e<=u



Uf: maxind∈[e..u] és
    ∀i∈[e..u]:(f(maxind  )>=f  (i))
 és maxért=f  (maxind)

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

Legnagyobb hőmérséklet

```
Be: e∈N, u∈N, n∈N, hőm∈R[1..n]
Sa: maxind∈Z
Ki:             lnh   ∈R
Ef: e<=u és e=1 és u=n és
    n>0 és
    ∀i∈[1..n]:(-100<=hőm[i]<=100)
Uf: maxind∈[1..n] és
    ∀i∈[1..n]:(hőm[maxind]>=hőm[i])
 és lnh   =hőm[maxind]

Uf: (maxind,lnh)=
                  MAX(i=1..n,hőm[i])
```

maxind, maxért	~	maxind, lnh
e..u	~	1..n
f(i)	~	hőm[i]

A visszavezetés megtehető

Példa

Egy hőmérsékletstatisztika alapján add meg a legnagyobb hőmérsékletet!

Feladatsablon

```
Be: e∈Z, u∈Z
Ki: maxind∈Z, maxért∈H
Ef: e<=u

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

Legnagyobb hőmérséklet

```
Be: n∈N, hőm∈R[1..n]
Ki: lnh∈R
Ef: n>0 és
    ∀i∈[1..n]:(-100<=hőm[i]<=100)
Uf: (,lnh)=
                MAX(i=1..n,hőm[i])
```

```
maxind, maxért  ~  maxind, lnh
e..u            ~  1..n
f(i)            ~  hőm[i]
```

```
maxért:=f(e); maxind:=e
i=e+1..u
    f(i)>maxért  (T / F)
    T: maxért:=f(i)
       maxind:=i
    F: -
```

```
Változó
  i:Egész
```

```
lnh:=hőm[1]; maxind:=1
i=1+1..n
    hőm[i]>lnh  (T / F)
    T: lnh:=hőm[i]
       maxind:=i
    F: -
```

```
Változó
  i:Egész,
  maxind:Egész
```