

OEP

5. táblás gyakorlat  
felsorolós programozási tételek I.

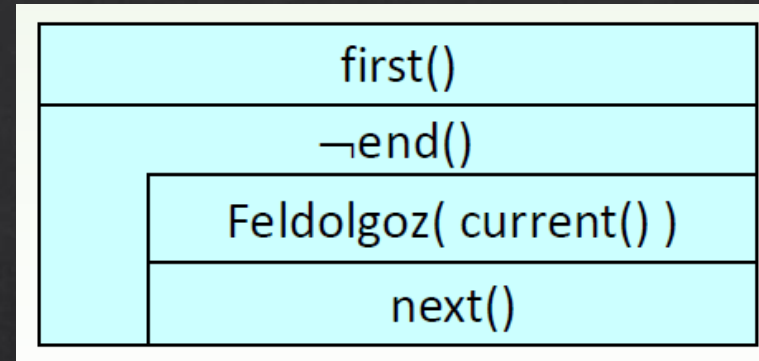
# Tartalom

Ismétlés (3. és 4.  
előadás)

Kaktuszos feladatok  
(minták az alap  
felsorolás tételek  
alkalmazására)

Feltétel fennállásáig  
tartó feldolgozás

## Felsoroló (3. előadás)



- ❑ Egy gyűjtemény feldolgozásához szükségünk van a gyűjtemény felsorolására (bejárására).
- ❑ Erre a felsorolásra úgy tekinthetünk, mint a gyűjtemény elemeiből képzett **véges sorozatra**, amely az alábbi **műveletekkel** rendelkezik:
  - *first()* : rááll a sorozat első elemére, azaz elkezdi a felsorolást
  - *next()* : rááll a sorozat soron következő elemére, azaz folytatja a felsorolást
  - $l := \text{end()} (l:\mathbb{L})$  : megmutatja, hogy a sorozat, azaz a felsorolás végére értünk-e
  - $e := \text{current()} (e:E)$ : visszaadja a sorozat, a felsorolás aktuális elemét

# Összegzés tétel felsorolóra

Összegezzük egy felsorolás elemeihez rendelt értékeket!

$$f : E \rightarrow H$$

$$+ : H \times H \rightarrow H$$

$$0 \in H \quad \text{baloldali neutrális elem}$$

$$A = ( t: \text{enor}(E), s: H )$$

$$Ef = ( t = t' )$$

$$Uf = ( s = \sum_{e \in t'} f(e) )$$

A felsorolás végén  $t = \langle \rangle$ , azaz nem marad  $t = t'$

$$s = \sum_{i=1..|t'|} f(t'_i)$$

*speciális eset: feltételes összegzés*

$$\sum_{\substack{e \in t' \\ \text{felt}(e)}} g(e) \text{ azaz } f(e) = \begin{cases} g(e) & \text{ha } \text{felt}(e) \\ 0 & \text{különben} \end{cases}$$

$s := 0$

$t.\text{first}()$

$\neg t.\text{end}()$

$s := s + f(t.\text{current}())$

$t.\text{next}()$

A felsorolás programozási tételket 3. előadás mutatja be!

# Szekvenciális input-, output fájl (4. előadás)

## Szekvenciális fájlok

- A **szekvenciális inputfájl** elnevezést olyan sorozatra értjük, amelynek mindig az első elemét lehet kiolvasni.
- Az olvasás egy  $st, e, x := \text{read}(x)$  értékadás, amelyet az  $st, e, x : \text{read}$  szimbólummal rövidítünk. Ebben
  - $x$  : **infile(E)** a szekvenciális inputfájltra hivatkozó változó
  - $e$  : **E** a fájl elejéről kiolvasott (leszakított) elemet tartalmazó változó
  - $st$  : **Status**={abnorm, norm} az olvasás sikerességét jelző változó

- A **szekvenciális outputfájl** elnevezést egy olyan sorozatra használjuk, amelynek végéhez lehet új elemet hozzáírni, és kezdetben az  $x := \langle \rangle$  értékadással üresre lehet állítani.
- Az írás egy  $x := \text{write}(x, e)$  értékadás, amelyet az  $x : \text{write}(e)$  jelöl majd:
  - $x$  : **outfile(E)** a szekvenciális outputfájltra hivatkozó változó
  - $e$  : **E** a fájl végéhez hozzáírt elem

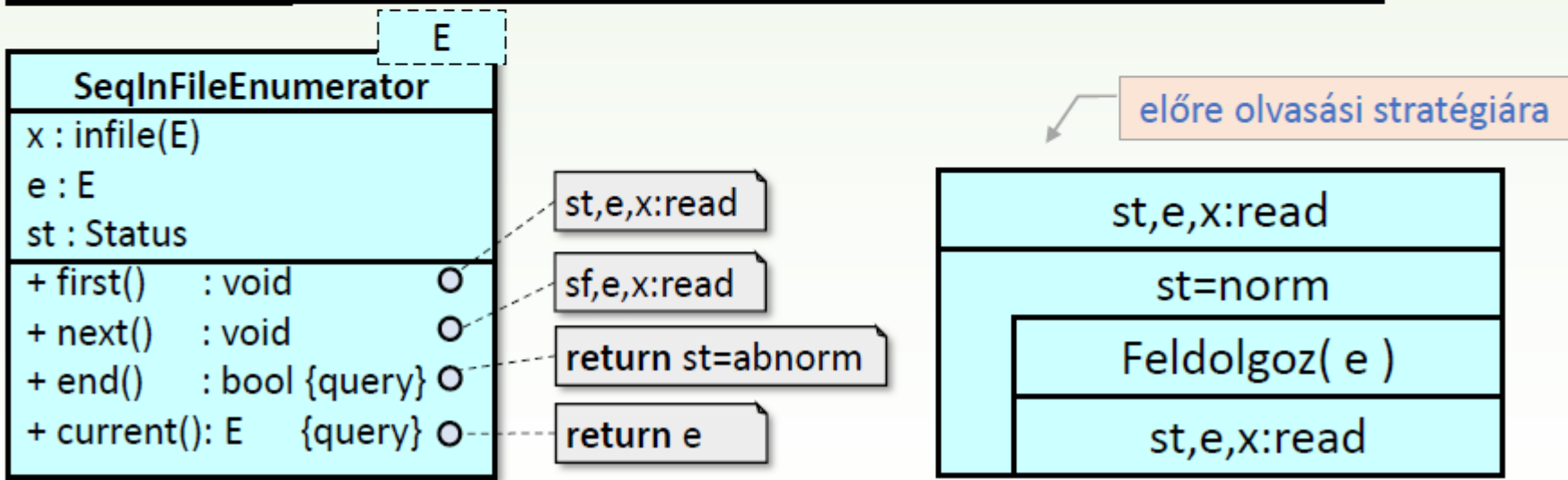
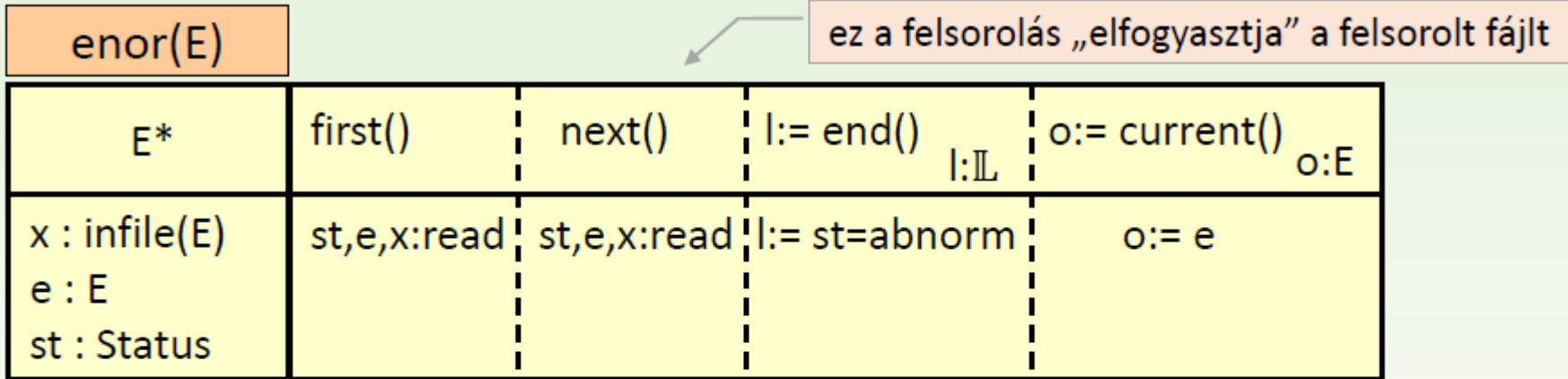
```
if x = <> then
    st := abnorm
else
    st := norm;
    e := x1;
    x := <x2, ..., x|x|>
endif
```

```
x := x ⊕ <e>
```



# Szekvenciális input fájl felsorolója

E-beli értékeket tartalmazó szekvenciális inputfájl elemeinek felsorolása



# Összegzés fájlkezeléshez

## Általános összegzés

$A = ( t:enor(E), s:H )$

$Ef = ( t = t_0 )$

$Uf = ( s = \sum_{e \in t_0} f(e) )$

$f : E \rightarrow H$   
 $+: H \times H \rightarrow H$   
 $0 \in H$

## Fájlfeldolgozás

$A = ( x:infile(E), y:outfile(F) )$

$Ef = ( x = x_0 )$

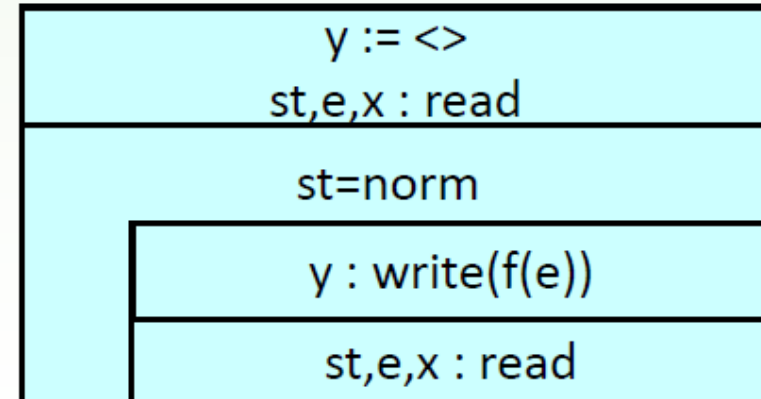
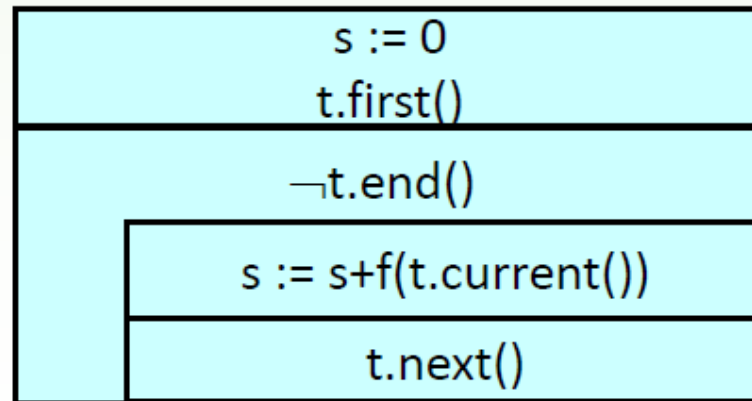
$Uf = ( y = \bigoplus_{e \in x_0} f(e) )$

$f : E \rightarrow F^*$   
 $\bigoplus : F^* \times F^* \rightarrow F^*$   
 $\langle \rangle \in F^*$  neutr. elem

Összegzés:

$t:enor(E) \sim x:infile(E)$   
 $H, +, 0 \sim F^*, \bigoplus, \langle \rangle$   
 $st, e, x : read$

előreolvasási technika



# Összegzés

- ◇ Egy szekvenciális inputfájlban egyes kaktuszfajtákról ismerünk néhány adatot: név, őshaza, virágszín, méret.
- ◇ (a) Válogassuk ki egy szekvenciális outputfájlba a piros virágú kaktuszok neveit!

## Összegzés

*Feladat:* Adott egy  $E$ -beli elemeket felsoroló  $t$  objektum és egy  $f:E \rightarrow H$  függvény. A  $H$  halmaz elemein értelmezett egy baloldali nulla elemmel rendelkező művelet (nevezzük ezt összeadásnak és jelölje a  $+$ ). Határozzuk meg a függvénynek a  $t$  elemeihez rendelt értékeinek összegét! (Üres felsorolás esetén az összeg értéke definíció szerint a nullelem:  $0$ ).

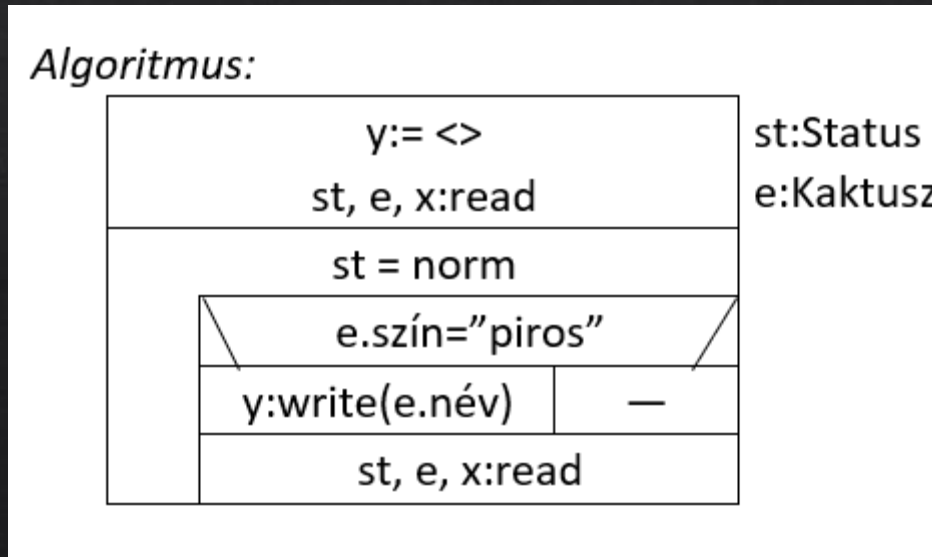
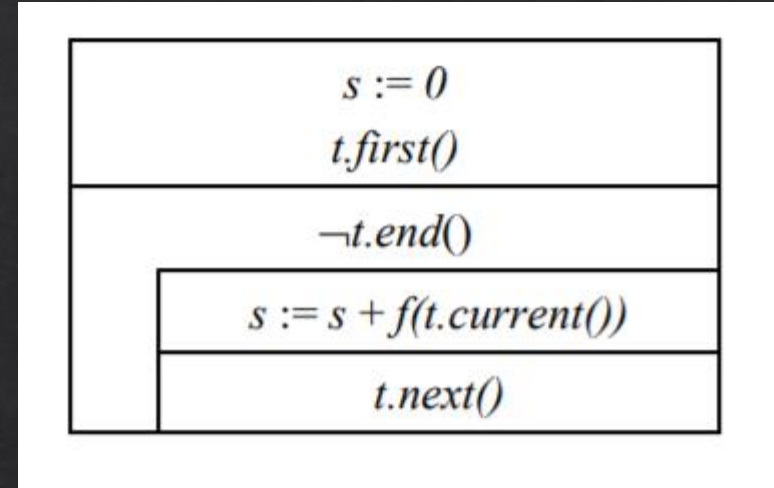
- ◇ Specifikáció:
- ◇  $A = ( x:\text{infile}(\text{Kaktusz}), y:\text{outfile}(\mathbb{S}) )$   
Kaktusz = rec(név:  $\mathbb{S}$ , szín:  $\mathbb{S}$ , ő:  $\mathbb{S}$ , méret:  $\mathbb{N}$ )
- ◇  $Ef = ( x = x_0 )$
- ◇  $Uf = ( y = \bigoplus_{e \in x_0} \langle e.\text{név} \rangle )$   
e.szín = "piros"

## Specifikáció:

$$\begin{aligned} A &= ( t:\text{enor}(E), s:H ) \\ Ef &= ( t=t' ) \\ Uf &= ( s = \sum_{e \in t'} f(e) ) \end{aligned}$$



- ◇ Visszavezetés:
- ◇ Összegzés (kiválogatás)
- ◇  $t:enor(E) \sim x:infile(Kaktusz) \quad (st, e, x:read)$
- ◇  $f(e) \sim \langle e.név \rangle$  ha  $e.szín = \text{"piros"}$
- ◇  $s \sim y$
- ◇  $H, +, 0 \sim S^*, \oplus, \langle \rangle$



# Tesztelési stratégiák

❑ **Fekete doboz:** a feladat (specifikációja) alapján felírt tesztesetek.

- az előfeltételt megszegő ún. érvénytelen tesztesetek
- az utófeltétel eseteinek vizsgálata
- ...

❑ **Fehér doboz:** a kód alapján felírt tesztesetek.

- algoritmus minden utasításának kipróbálása
- algoritmus minden vezérlési csomópontjának (elágazás, ciklus) kipróbálása
- ...

❑ **Szürke doboz:** végrehajtható specifikáció által előre vetített algoritmus működését ellenőrző tesztesetek.

- Ha ez a végrehajtható specifikáció algoritmus mintákra utal, akkor az **algoritmus minták szokásos teszteseteit** kell vizsgálni.

Tesztelés  
3. előadás

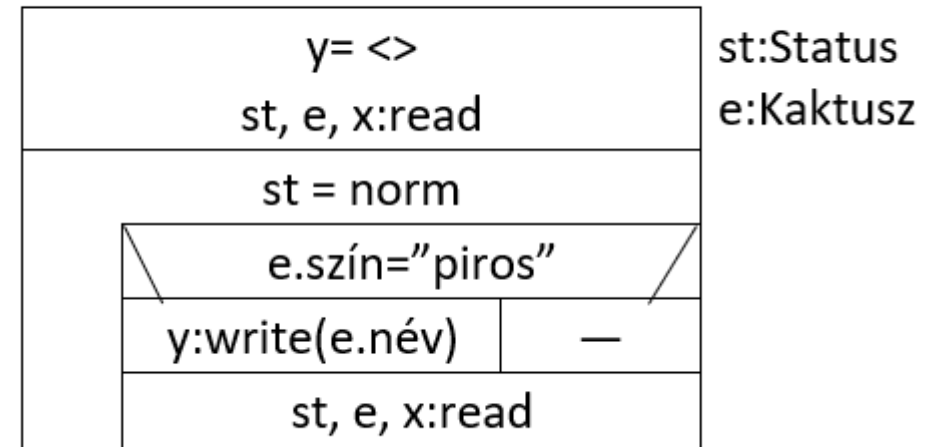
# Algoritmus minták tesztesei

- ❑ Felsoroló szerint (mindegyik algoritmus minta esetén)
  - eltérő *hosszúságú* felsorolások: nulla, egy illetve hosszabb felsorolásokra is kipróbájuk az algoritmust
  - Felsorolás *kezdete* és *vége*: feldolgozza-e az algoritmus a felsorolás *első* ill. *utolsó* elemét
- ❑ Funkció szerint (algoritmus mintánként eltérő)
  - összegzés: felsorolás hosszának *skálázása*
  - keresés, számlálás: *van vagy nincs* keresett tulajdonságú elem
  - max. kiv.: *egyetlen*, illetve *több* azonos maximális érték
  - felt. max. ker.: - *van vagy nincs* keresett tulajdonságú elem
    - feltételt kielégítő *egyetlen*, illetve *több* azonos maximális érték
    - a *legnagyobb értékű elem nem* elégíti ki a feltételt
- ❑ A  $felt(e)$  és  $f(e)$  kifejezések kiszámolásánál használt műveletek sajátosságai.

## Tesztelési terv a feladathoz

- ◆ Felsorolás hossza szerint:
  - ◆ 0,1,2,több
- ◆ Feldolgozza-e az elsőt/utolsót:
  - ◆ Első kaktusz piros virágú /nem piros virágú
  - ◆ Utolsó kaktusz piros virágú / nem piros virágú
- ◆ Mind piros
- ◆ Egyik sem piros
- ◆ Vegyesen piros/nem piros

Algoritmus:





# Számlálás

- ◇ Egy szekvenciális inputfájlban egyes kaktuszfajtákról ismerünk néhány adatot: név, őshaza, virágszín, méret.
- ◇ (b) Számoljuk meg a piros virágú kaktuszokat!

## *Számlálás*

*Feladat:* Adott egy  $E$ -beli elemeket felsoroló  $t$  objektum és egy  $felt:E \rightarrow \mathbb{L}$  feltétel. A felsoroló objektum hány elemére teljesül a feltétel?

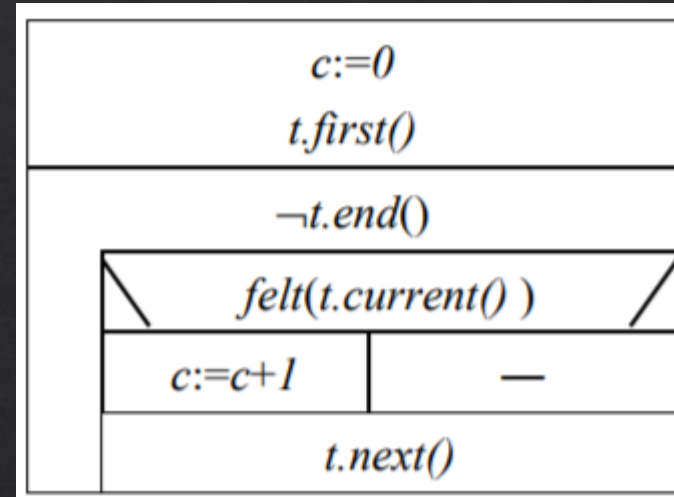
- ◇ Specifikáció:
- ◇  $A = ( x:\text{infile}(\text{Kaktusz}), \text{db}:\mathbb{N} )$   
 $\text{Kaktusz} = \text{rec}(\text{név}:\mathbb{S}, \text{szín}:\mathbb{S}, \text{ős}:\mathbb{S}, \text{méret}:\mathbb{N})$
- ◇  $Ef = ( x = x_0 )$
- ◇  $Uf = ( \text{db} = \sum_{e \in x_0} 1 )$   
e.szín="piros"

## *Specifikáció:*

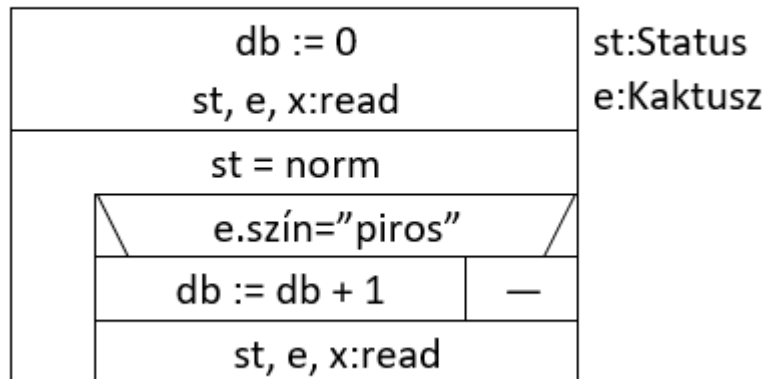
$$\begin{aligned} A &= ( t:\text{enor}(E), c:\mathbb{N} ) \\ Ef &= ( t = t' ) \\ Uf &= ( c = \sum_{e \in t'} 1 ) \\ &\quad \text{felt}(e) \end{aligned}$$



- ◇ Visszavezetés:
- ◇ Számlálás
- ◇  $t:enor(E) \sim x:infile(Kaktusz) (st,e,x:read)$
- ◇  $felt(e) \sim e.szín="piros"$
- ◇  $c \sim db$



**Algoritmus:**



- ◇ Tesztelés:
- ◇ Felsorolás hossza szerint:
  - ◇ 0,1,2,több
- ◇ Feldolgozza-e az elsőt/utolsót:
  - ◇ Első katusz piros virágú /nem piros virágú
  - ◇ Utolsó kaktusz piros virágú / nem piros virágú
- ◇ Mind piros
- ◇ Egyik sem piros
- ◇ Vegyesen piros/nem piros

# Optimista eldöntés

- ◊ Egy szekvenciális inputfájlban egyes kaktuszfajtákról ismerünk néhány adatot: név, őshaza, virágszín, méret.
- ◊ (c) Igaz-e, hogy minden kaktusz virága piros?
- ◊ **Lineáris keresés:**  
Feladat: Adott egy  $E$ -beli elemeket felsoroló  $t$  objektum és egy  $\text{felt}:E \rightarrow \mathbb{L}$  feltétel. Keressük a  $t$  bejárása során az első olyan elemi értéket, amely kielégíti a  $\text{felt}:E \rightarrow \mathbb{L}$  feltételt
- ◊ **Optimista eldöntés:**  
Feladat: Adott egy  $E$ -beli elemeket felsoroló  $t$  objektum és egy  $\text{felt}:E \rightarrow \mathbb{L}$  feltétel. A felsoroló minden elemére igazat ad a feltétel?
- ◊ Specifikáció:
  - ◊  $A = ( x:\text{infile}(\text{Kaktusz}), l:\mathbb{L} )$   
 $\text{Kaktusz} = \text{rec}(\text{név}:\mathbb{S}, \text{szín}:\mathbb{S}, \text{ős}:\mathbb{S}, \text{méret}:\mathbb{N})$
  - ◊  $E_f = ( x = x_0 )$
  - ◊  $U_f = ( 1, (\text{st}, e, x) = \forall \text{SEARCH}_{e \in x_0} e.\text{szín} = \text{''piros''} )$

Specifikáció:

$A = ( t:\text{enor}(E), l:\mathbb{L}, \text{elem}:E )$

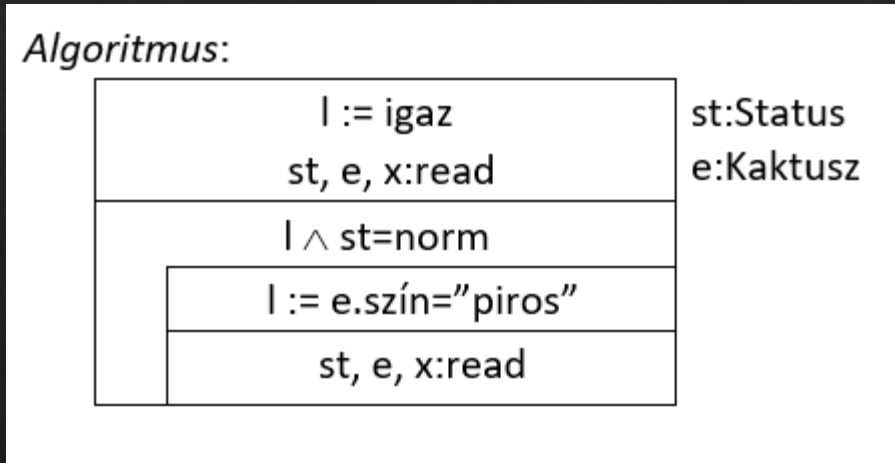
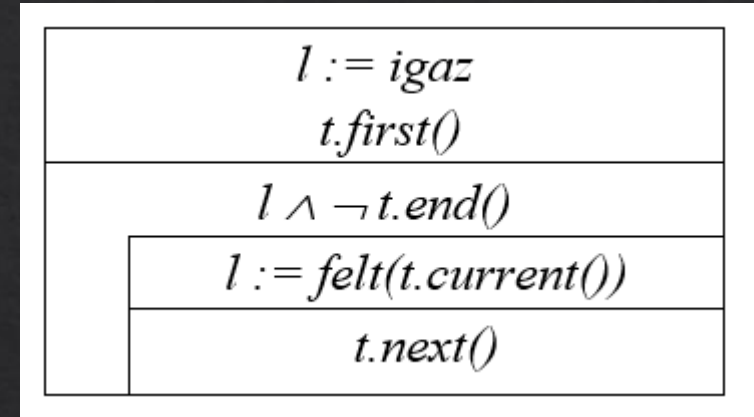
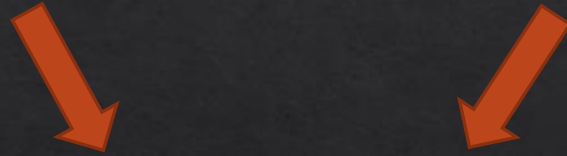
$E_f = ( t = t' )$

$U_f = ( l, \text{elem}, t =$

$\forall \text{SEARCH}_{e \in t} \text{felt}(e) )$

Eldöntés esetén „elem” elhagyható.

- ◇ Visszavezetés:
- ◇ Opt.lin.ker
- ◇  $t:enor(E) \sim x:infile(Kaktusz) (st,e,x:read)$
- ◇  $felt(e) \sim e.szín="piros"$



- ◇ Tesztelés:
- ◇ Felsorolás hossza szerint:
  - ◇ 0,1,2,több (mind piros)
- ◇ Feldolgozza-e az elsőt/utolsót:
  - ◇ Első nem piros virágú, többi piros
  - ◇ Utolsó nem piros virágú, többi piros
- ◇ Egyik sem piros
- ◇ Vegyesen piros/nem piros



## Több összegzés

- ◊ Egy szekvenciális inputfájlban egyes kaktuszfajtákról ismerünk néhány adatot: név, őshaza, virágszín, méret.
- ◊ (d) Válogassuk ki egy szekvenciális outputfájlba a piros virágú kaktuszok, egy másikba a mexikói őshazájú kaktuszok neveit!
- ◊ Specifikáció:
- ◊  $A = ( x:\text{infile}(\text{Kaktusz}), y, z :\text{outfile}(\mathbb{S}) )$   
 $\text{Kaktusz} = \text{rec}(\text{név}:\mathbb{S}, \text{szín}:\mathbb{S}, \text{ős}:\mathbb{S}, \text{méret}:\mathbb{N})$
- ◊  $Ef = ( x = x_0 )$
- ◊  $Uf = ( y = \bigoplus_{e \in x_0} \langle e.\text{név} \rangle \wedge z = \bigoplus_{e \in x_0} \langle e.\text{név} \rangle )$   
 $e.\text{szín} = \text{''piros''} \quad e.\text{ős} = \text{''Mexikó''}$

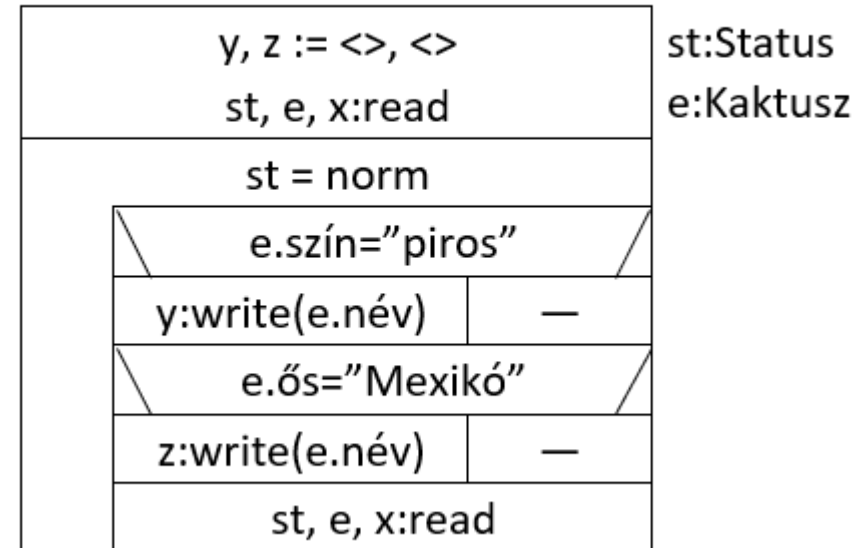
A felsorolót egyszer tudjuk használni. Mi lesz a visszavezetés?



- ◊ Egyszerűbb megoldás:  
két egyszerű összegzés (párhuzamosítjuk, azaz összevonjuk egy ciklusba)
- ◊ „Igazi” megoldás:  
speciális „dupla összegző” f() függvény
- ◊  $t:enor(E) \sim x:infile(Kaktusz) \quad (st,e,x:read)$
- ◊  $f_1(e) \sim \langle e.név \rangle$  ha  $e.szín="piros"$
- ◊  $f_2(e) \sim \langle e.név \rangle$  ha  $e.ős="Mexikó"$
- ◊  $s \sim y, z$
- ◊  $H, +, 0 \sim \mathbb{S}^*, \oplus, \langle \rangle$

Kvíz2

Algoritmus:





# Feltételes maximum keresés

- ◇ Egy szekvenciális inputfájlban egyes kaktuszfajtákról ismerünk néhány adatot: név, őshaza, virágszín, méret.
- ◇ (e) Melyik a legnagyobb piros virágú kaktusz neve?

## *Feltételes maximumkeresés*

*Feladat:* Adott egy  $E$ -beli elemeket felsoroló  $t$  objektum, egy  $felt:E \rightarrow \mathbb{L}$  feltétel és egy  $f:E \rightarrow H$  függvény. A  $H$  halmazon definiáltunk egy teljes rendezési relációt. Határozzuk meg  $t$  azon elemeihez rendelt  $f$  szerinti értékek között a legnagyobbat, amelyek kielégítik a  $felt$  feltételt.

*Specifikáció:*

$$A = ( t:enor(E), l:\mathbb{L}, max:H, elem:E )$$

$$Ef = ( t=t' )$$

$$Uf = ( (l, max, elem) = \underset{felt(e)}{MAX}_{e \in t} f(e) )$$

- ◇ Specifikáció:
- ◇  $A = ( x:infile(\text{Kaktusz}), l:\mathbb{L}, név:\mathbb{S} ) \quad \text{Kaktusz} = \text{rec}(név:\mathbb{S}, \text{szín}:\mathbb{S}, \text{ős}:\mathbb{S}, \text{méret}:\mathbb{N})$
- ◇  $Ef = ( x=x_0 )$
- ◇  $Uf = ( (l, max, elem) = \underset{e.szín="piros"}{MAX}_{e \in x_0} e.méret \wedge 1 \rightarrow név = elem.név ) \quad max:\mathbb{Z}, elem:\text{Kaktusz}$

- ◇ Visszavezetés:
- ◇ Felt.max.ker
- ◇  $t:enor(E) \sim x:infile(Kaktusz) \quad (st,e,x:read)$
- ◇  $f(e) \sim e.méret$
- ◇  $felt(e) \sim e.szín="piros"$
- ◇  $H, > \sim \mathbb{N}, >$

$l := hamis; t.first()$		
$\neg t.end()$		
$\neg felt(t.current())$	$felt(t.current()) \wedge l$	$felt(t.current()) \wedge \neg l$
SKIP	$f(t.current()) > max$	$l, max, elem :=$
	$max, elem :=$ $f(t.current()), t.current()$	igaz, $f(t.current()), t.current()$
$t.next()$		



**Algoritmus:**

Elég a legkisebb kaktusz helyett (elem) annak nevét (név) nyilvántartani.

$l := hamis$			$st: Status$
$st, e, x: read$			
$st = norm$			
$e.szín \neq "piros"$	$e.szín = "piros" \wedge l$	$e.szín = "piros" \wedge \neg l$	$e: Kaktusz$
—	$e.méret > max$	$l, max, név :=$	
	$max, név :=$ $e.méret, e.név$	igaz, $e.méret, e.név$	
$st, e, x: read$			

*Megjegyzés:* A specifikáció szerint egy extra elágazás is kell a név változó értékének megadásához, de itt ezt beépült a feltételes maximum keresés ciklusába.

# Szürke doboz tesztelés terve

- ◇ Felsoroló hossza szerint:
  - ◇ 0 hosszú → nincs
  - ◇ 1 hosszú → van / nincs
  - ◇ több hosszú:
    - ◇ Elején van egy piros
    - ◇ Végén van egy piros
- ◇ Tétel esetei (több hosszú):
  - ◇ Nincs piros
  - ◇ Több piros, első a legnagyobb
  - ◇ Több piros, utolsó a legnagyobb
  - ◇ Több piros, belső a legnagyobb
  - ◇ Több piros mind egyforma nagy
  - ◇ Abszolút legnagyobb piros/nem piros



Kvíz3

## Feltétel fennállásáig tartó felsorolás

- ◊ Adott egy egész számokat tartalmazó szekvenciális inputfájl.
- ◊ (a) Hány páros szám előzi meg benne az első negatív számot?
- ◊  $\langle 1\ 3\ 4\ 2\ 7\ -1\ 2\ -2\ \dots \rangle$
- ◊  $\langle 1\ 3\ 5\ -1\ 2\ -2\ \dots \rangle$
- ◊  $\langle -1\ 2\ -2\ \dots \rangle$
- ◊  $\langle 1\ 3\ 4\ 2\ 7\ 2 \rangle$
- ◊  $\langle -1 \rangle$
- ◊  $\langle \rangle$

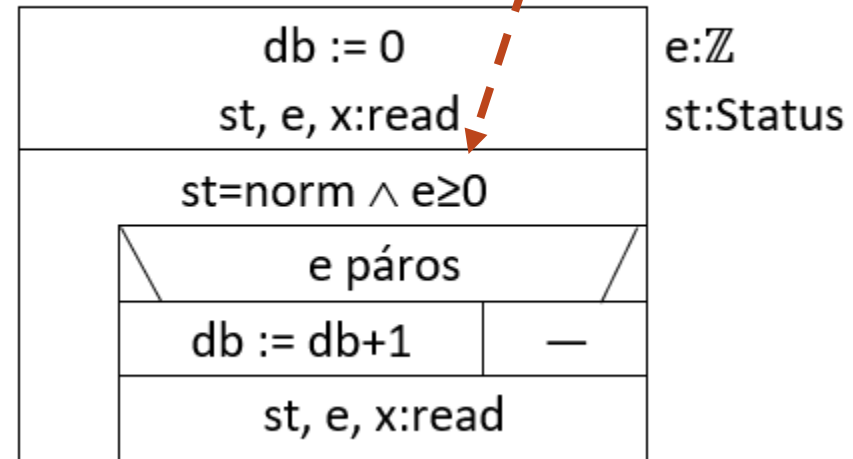
- Nem akarjuk a fájlt végig olvasni! Az első negatív szám megjelenésénél leállhatunk.
- A felsorolót csak egy adott feltétel fennállásáig (amíg szám nem negatív) használjuk!



- ◇ *Specifikáció:*
- ◇  $A = ( x:\text{infile}(\mathbb{Z}), \text{db}:\mathbb{N} )$
- ◇  $Ef = ( x=x_0 )$
- ◇  $Uf = ( \text{db} = \sum_{\substack{e \in X_0 \\ e \text{ páros}}}^{e \geq 0} 1 )$
- ◇ *Számlálás, feltétel fennállásáig*
- ◇  $t:\text{enor}(E) \sim x:\text{infile}(\mathbb{Z}) \quad (\text{st}, e, x:\text{read})$   
amíg:  $e \geq 0$
- ◇  $\text{felt}(e) \sim e \text{ páros}$
- ◇  $c \sim \text{db}$

A specifikációban a programozási tétel kulcsszava ( $\Sigma$ ) feletti extra feltétel jelzi, hogy meddig tartson a felsorolás.

Algoritmus:



Más tételek esetén is alkalmazható, amelyek a felsoroló végéig dolgozzák fel az elemeket:

- Összegzés
- Feltételes maximum keresés
- Maximum kiválasztás (nem lehet üres a feltételig felsorolt elemek halmaza)