

## 6. OEP gyakorlat

Témakör: szekvenciális inputfájl több szakaszból álló feldolgozása a felsoroló megállításával és folytatásával

1. Adott egy egész számokat tartalmazó gyűjtemény.

a) Hány páros szám előzi meg az első negatívát?

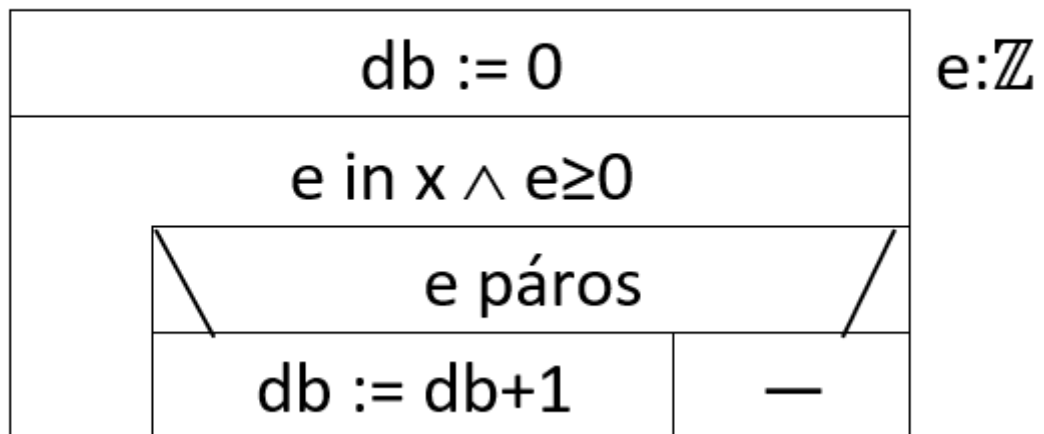
*Specifikáció:*

$A = (x: \text{enor}(\mathbb{Z}), db: \mathbb{N})$

$Ef = (x = x_0)$

$Uf = (db = \sum_{\substack{e \in x_0 \\ e \text{ páros}}}^{e \geq 0} 1)$

*Algoritmus:*



A számlálásnak három outputja van:  $(db, e', x')$ . Az  $e'$  a felsorolás első negatív eleme (ha van ilyen), azaz az  $e'$  az  $e$  változó értéke. Az  $x'$  pedig a hátralevő elemek felsorolója. (Az  $e'$  és  $x'$  most nem kell.)

*Számlálás, feltétel fennállásáig*

$t: \text{enor}(E) \sim x: \text{enor}(\mathbb{Z})$  amíg:  $e \geq 0$

$\text{felt}(e) \sim e \text{ páros}$

$c \sim db$

db := 0		e:ℤ				
x.First()						
¬x.End() ∧ x.Current() ≥ 0						
<table><tr><td colspan="2">x.Current() páros</td></tr><tr><td>db := db+1</td><td>—</td></tr></table>			x.Current() páros		db := db+1	—
x.Current() páros						
db := db+1	—					
x.Next()						

- ◊ Adott egy egészekből álló szekvenciális input fájl.
- ◊ (b) Hány páros szám követi az első negatív számot?

◊  $\langle 1\ 3\ 4\ \textcolor{red}{-1}\ 2\ -2\ 1\ 4\ 3 \rangle$

◊  $\langle 1\ 3\ 5\ \textcolor{red}{-1}\ 3\ 1\ 5 \rangle$

◊  $\langle \textcolor{red}{-1} \rangle$

◊  $\langle 1\ 3\ 4\ 2\ 7\ 2 \rangle$

◊  $\langle \textcolor{red}{-1} \rangle$

◊  $\langle \rangle$

- Meg kell találni a fájlban az első negatív számot, azt követően indulhat a számlálás, a fájl végéig.
- Lehet, hogy nem lesz negatív szám a fájlban.
- Milyen tétellel dolgozzuk fel a fájl első részét?
  - Lineáris keresés
  - Kiválasztás

## Megoldás lineáris keresést használva

### Specifikáció:

- $A = (x:\text{infile}(\mathbb{Z}), \text{db}:\mathbb{N})$
- $Ef = (x=x_0)$
- $Uf = ((l, \text{elem}, (st', e', x')) = \text{SEARCH}_{e \in x_0}(e < 0) \wedge \text{db} = \sum_{e \in (e', x')} 1)$

A tétel ciklusa eggyel tovább olvas!

„elem”-ben lesz az első negatív, ha találtunk

A folytatásnak fel kell dolgoznia az előre olvasott elemet, és a fájl maradék részét!

$l := \text{hamis}; t.\text{first}()$
$\neg l \wedge \neg t.\text{end}()$
$\text{elem} := t.\text{current}()$
$l := \text{felt}(\text{elem})$
$t.\text{next}()$

$\langle 1 \ 3 \ 4 \ -1 \ 2 \ -2 \ 1 \ 4 \ 3 \rangle$

$l=\text{true}; \text{elem}=-1$

$e'=2$

$x'$  – a fájl hátralévő része

### Specifikáció:

- $A = (x:\text{infile}(\mathbb{Z}), \text{db}:\mathbb{N})$
- $Ef = (x=x_0)$
- $Uf = ((l, \text{elem}, (st', e', x')) = \text{SEARCH}_{e \in x_0}(e < 0) \wedge \text{db} = \sum_{e \in (e', x')} 1)$

### Visszavezetés:

### Lineáris keresés

- $t:\text{enor}(E) \sim x:\text{infile}(\mathbb{Z}) \ (st, e, x:\text{read})$
- $\text{felt}(e) \sim e < 0$
- Számlálás**
- $t:\text{enor}(E) \sim e, x:\text{infile}(\mathbb{Z})$  „folytatása”
- $\text{felt}(e) \sim e \text{ páros}$
- $c \sim \text{db}$

Fontos!  
Számlálásban nincs  
előreolvasás!  
first() elmarad

### Algoritmus:

$l := \text{hamis}$	$e, \text{elem}:\mathbb{Z}$
$st, e, x:\text{read}$	
$st = \text{norm} \wedge \neg l$	$st:\text{Status}$
$l, \text{elem} := e < 0, e$	
$st, e, x:\text{read}$	
$\text{db} := 0$	
$st = \text{norm}$	
$e \text{ páros}$	
$\text{db} := \text{db} + 1$	—
$st, e, x:\text{read}$	

## Megoldás kiválasztás tételt használva

◆ *Specifikáció:*

◆  $A = (x:\text{infile}(\mathbb{Z}), db:\mathbb{N})$

◆  $Ef = (x = x_0)$

◆  $Uf = ((e', (st', e', x')) =$

$\text{SELECT}_{st, e \in x_0} (st = \text{abnorm} \vee e < 0) \wedge$   
 $\wedge db = \sum_{\substack{e \in x' \\ e \text{ páros}}} 1)$

„e”-ben lesz az első negatív,  
ha találtunk

A tétel ciklusa leáll,  
amikor negatívot olvasott

$t.\text{first}()$
$\neg \text{felt}(t.\text{current}())$
$t.\text{next}()$
$\text{elem} := t.\text{current}()$

A folytatás feldolgozza a fájl  
maradék részét.

$\langle 1 \ 3 \ 4 \ -1 \ 2 \ -2 \ 1 \ 4 \ 3 \rangle$

$e = -1$

$x'$  – a fájl hátralévő  
része

◆ *Specifikáció:*

◆  $A = (x:\text{infile}(\mathbb{Z}), db:\mathbb{N})$

◆  $Ef = (x = x_0)$

◆  $Uf = ((e', (st', e', x')) =$

$\text{SELECT}_{st, e \in x_0} (st = \text{abnorm} \vee e < 0) \wedge$   
 $\wedge db = \sum_{\substack{e \in x' \\ e \text{ páros}}} 1)$

◆ *Visszavezetés:*

◆ *Kiválasztás*

◆  $t:\text{enor}(E) \sim x:\text{infile}(\mathbb{Z}) (st, e, x:\text{read})$

◆  $\text{felt}(e) \sim st = \text{abnorm} \vee e < 0$

◆ *Számlálás*

◆  $t:\text{enor}(E) \sim x:\text{infile}(\mathbb{Z})$  „folytatása”  
 $\text{first}()$  helyett  $\text{next}()$

◆  $\text{felt}(e) \sim e \text{ páros}$

◆  $c \sim db$

*Algoritmus:*

$st, e, x:\text{read}$	$e:\mathbb{Z}$
$st = \text{norm} \wedge e \geq 0$	$st:\text{Status}$
$st, e, x:\text{read}$	
$db := 0$	
$st, e, x:\text{read}$	
$st = \text{norm}$	
$e \text{ páros}$	
$db := db + 1$	—
$st, e, x:\text{read}$	

Fontos!

Tovább kell olvasni, hogy az első  
negatív utáni számmal  
kezdhesen a számlálás.



# Azonos felsorolót használó tételek szekvenciába fűzése

- ◊ Adott egy egészekből álló szekvenciális input fájl.
- ◊ (c) Hány páros szám van az első negatív számot megelőzően, és hány azt követően?
- ◊  $\langle 1\ 3\ 4\ -1\ 2\ -2\ 1\ 4\ 3 \rangle$
- ◊  $\langle 1\ 3\ 5\ -1\ 3\ 1\ 5 \rangle$
- ◊  $\langle -1 \rangle$
- ◊  $\langle 1\ 3\ 4\ 2\ 7\ 2 \rangle$
- ◊  $\langle -1 \rangle$
- ◊  $\langle \rangle$

- A fájlban lévő első negatív elem a „vízválasztó”.
- Előtte lévő és utána következő fájl részeket kell tétellel feldolgozni.
- Első lépés: feltétel fennállásáig tartó számlálás.
- Negatív elemet „átlépjük”.
- Utána a fájl végéig egy újabb számlálás.

- ◊ *Specifikáció:*
- ◊  $A = (x:\text{infile}(\mathbb{Z}), \text{dbe}, \text{dbu}:\mathbb{N})$   
 $Ef = (x=x_0)$
- ◊  $Uf = ((\text{dbe}, (\text{st}', e', x')) = \sum_{\substack{e \in x_0 \\ e \text{ páros}}}^{e \geq 0} 1 \\ \wedge \text{dbu} = \sum_{\substack{e \in x' \\ e \text{ páros}}} 1)$
- ◊ Számlálás, feltétel fennállásáig
- ◊  $t:\text{enor}(E) \sim x:\text{infile}(\mathbb{Z}) (\text{st}, e, x:\text{read})$   
amíg:  $e \geq 0$
- ◊  $\text{felt}(e) \sim e \text{ páros}$
- ◊  $c \sim \text{dbe}$
- ◊ Számlálás
- ◊  $t:\text{enor}(E) \sim x:\text{infile}(\mathbb{Z}) (\text{st}, e, x:\text{read})$   
first() helyett next()
- ◊  $\text{felt}(e) \sim e \text{ páros}$
- ◊  $c \sim \text{dbu}$

## Algoritmus:

dbe := 0	e:ℤ
st, e, x:read	st:Status
st=norm ∧ e ≥ 0	
e páros	
dbe := dbe + 1	—
st, e, x:read	
dbu := 0	
st, e, x:read	
st=norm	
e páros	
dbu := dbu + 1	—
st, e, x:read	

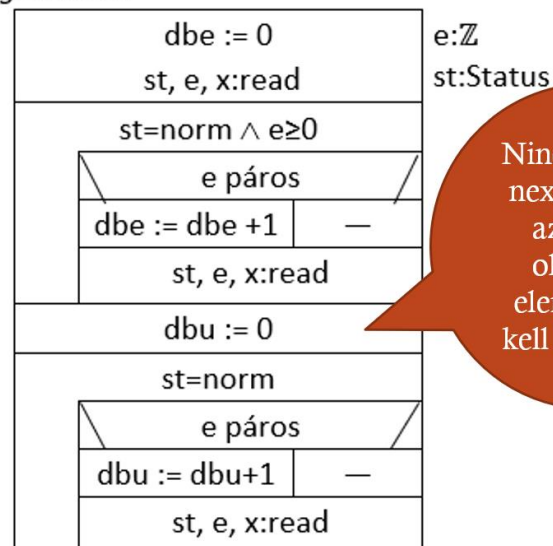
Tételek közötti átmenet: e-ben az első negatív van (ha van negatív szám a fájlban), így tovább kell olvasni.

- Adott egy egészekből álló szekvenciális input fájl.
- (d) Hány páros szám van az első negatív számot megelőzően, és hány azt követően, azzal együtt?
- $\langle 1\ 3\ 4\ -2\ 2\ -2\ 1\ 4\ 3 \rangle$
- $\langle 1\ 3\ 5\ -2\ 3\ 1\ 5 \rangle$
- $\langle -2 \rangle$
- $\langle 1\ 3\ 4\ 2\ 7\ 2 \rangle$
- $\langle -2 \rangle$
- $\langle \rangle$

- A fájlban lévő első negatív elem előttiiek tartoznak az első számláláshoz.
- Első lépés: feltétel fennállásáig tartó számlálás.
- Utána a fájl végéig egy újabb számlálás.
- Negatív elemet a második számlálás fel kell dolgozza!

- Specifikáció:
- $A = (x:\text{infile}(\mathbb{Z}), \text{dbe}, \text{dbu}:\mathbb{N})$   
 $Ef = (x = x_0)$
- $Uf = ((\text{dbe}, (\text{st}', e', x')) = \sum_{\substack{e \in x_0 \\ e \text{ páros}}}^{e \geq 0} 1$   
 $\wedge \text{dbu} = \sum_{\substack{e \in (e', x')}}^{e \text{ páros}} 1)$
- Számlálás, feltétel fennállásáig
- $t:\text{enor}(E) \sim x:\text{infile}(\mathbb{Z}) (\text{st}, e, x:\text{read})$   
amíg:  $e \geq 0$
- $\text{felt}(e) \sim e \text{ páros}$
- $c \sim \text{dbe}$
- Számlálás
- $t:\text{enor}(E) \sim x:\text{infile}(\mathbb{Z}) (\text{st}, e, x:\text{read})$   
first() nélkül
- $\text{felt}(e) \sim e \text{ páros}$
- $c \sim \text{dbu}$

#### Algoritmus:



Nincs first() / next(), mivel az „előre olvasott” elemet is fel kell dolgozni!

- Egy más utáni napok átlaghőmérsékleteit egy szekvenciális inputfájl tartalmazza. Mennyi az első fagypont alatti értéket megelőző napok (ilyenek biztosan vannak) hőmérsékleteinek átlaga, továbbá az első fagypont alatti értéktől kezdődően (az első fagypont alatti napot is beleértve) vajon minden nap fagypont alatt maradt-e a hőmérséklet, és mi volt a legalacsonyabb hőmérséklet?

#### Specifikáció:

$A = (x:\text{infile}(\mathbb{R}), a:\mathbb{R}, l:\mathbb{L}, \text{kicsi}:\mathbb{R})$

$Ef = (x = x_0 \wedge |x| \geq 2 \wedge x[1] \geq 0 \wedge \exists i \in [2..|x|]: x[i] < 0)$

$Uf = ((s, e', x') = \sum_{e \in x_0}^{e \geq 0} (e) \wedge (db, e', x') = \sum_{e \in x_0}^{e \geq 0} 1 \wedge a = s/db \wedge$

$\wedge l = \forall \text{SEARCH}_{e \in \langle e', x' \rangle} (e < 0) \wedge \text{kicsi} = \text{MIN}_{e \in \langle e', x' \rangle} e)$

Algoritmus:

$e, s: \mathbb{R}, db: \mathbb{N} \text{ st: Status}$

feltétel fennállásáig tartó két összegzés  
közös ciklusban

átlagszámítás

lin. ker. és min. kiv. közös ciklusban  
az inicializálásnál figyelembe vesszük,  
hogy az első megvizsgálandó elemet  
már korábban beolvastuk,  
ami biztosan negatív szám

s, db := 0.0, 0 st, e, x : read	
st=norm $\wedge$ $e \geq 0$	
s, db := s+e, db +1	
st, e, x : read	
a := s / db	
l, kicsi := igaz, e st, e, x : read	
st=norm	
l, kicsi := l $\wedge$ $e < 0$ , min(kicsi,e)	
st, e, x : read	

2. Egy horgászversenyen a horgászok eredményét egy szekvenciális inputfájlban rögzítették. A fájl egy eleme egy horgász nevét és a halfogásainak sorozatát tartalmazza (lásd előző gyakorlat). Gyűjtsük ki azon horgászok neveit, akik legalább négy 1 méternél (100 cm) hosszabb harcsát fogtak az első legalább 1 kilós ponty-fogásukat követően.

Specifikáció:

$A = ( f: \text{infile}(\text{Horgász}), \text{lista}: \mathbb{S}^* ) \quad \text{Horgász} = \text{rec}(\text{név}: \mathbb{S}, \text{eredm}: \text{Fogás}^* )$

$\text{Fogás} = \text{rec}(\text{idő}: \mathbb{S}, \text{hal}: \mathbb{S}, \text{súly}: \mathbb{R}, \text{hossz}: \mathbb{R})$

$Ef = ( f=f_0 )$

$Uf = ( \text{lista} = \bigoplus_{e \in f_0} \langle e.\text{név} \rangle \mid \text{jó}(e.\text{eredm}) )$

Összegzés (kiválogatás)

$t: \text{enor}(E) \sim f: \text{infile}(\text{Horgász}) \quad (st, e, f: \text{read})$   
 $f(e) \sim \langle e.\text{név} \rangle \text{ ha } \text{jó}(e.\text{eredm})$   
 $s \sim \text{lista}$   
 $H, +, 0 \sim \mathbb{S}^*, \bigoplus, \langle \rangle$

Algoritmus:

lista := < > st, e, f : read		st:Status e:Horgász
st = norm		
jó(e.eredm)		
lista := lista $\oplus$ <e.név>	—	
st, e, f : read		

Részfeladat: **ok := jó(e.fogás)** ahol  $\text{jó} : \text{Fogás}^* \rightarrow \mathbb{L}$

$A = (x : \text{Fogás}^*, \text{ok} : \mathbb{L}) \quad \text{Fogás} = \text{rec}(\text{idő} : \mathbb{S}, \text{hal} : \mathbb{S}, \text{súly} : \mathbb{R}, \text{hossz} : \mathbb{R})$

$Ef = (x = x_0)$

$Uf = (x = x_0 \wedge (e', x') = \text{SELECT}_{e \in x_0} (e.\text{hal} = \text{"ponty"} \wedge e.\text{súly} \geq 1.0) \wedge$   
 $\wedge db = \sum_{\substack{e \in \langle e', x' \rangle \\ e.\text{hal} = \text{"harcsa"} \wedge e.\text{hossz} \geq 1.0}} \mathbf{1} \wedge \text{ok} = db \geq 4)$

Algoritmus: **ok := jó(x)** ahol  $\text{jó} : \text{Fogás}^* \rightarrow \mathbb{L}$

Kiválasztás

$t:\text{enor}(E) \sim x : \text{Fogás}^*$

$\text{felt}(e) \sim (e.\text{hal} = \text{"ponty"} \wedge e.\text{súly} \geq 1.0)$

Számlálás

$t:\text{enor}(E) \sim x : \text{Fogás}^*$

$\text{felt}(e) \sim e.\text{hal} = \text{"harcsa"} \wedge e.\text{hossz} \geq 1.0$

$c \sim db$

e in x $\wedge \neg(e.\text{hal} = \text{"ponty"} \wedge e.\text{súly} \geq 1.0)$		db:ℕ
—		
db := 0		
e in x		
$e.\text{hal} = \text{"harcsa"} \wedge e.\text{hossz} \geq 1.0$		
db := db+1	—	
ok := db≥4		