

6. táblás gyakorlat feladatai

1. Adott egy egész számokat tartalmazó szekvenciális inputfájl.

a) Hány páros szám követi az első negatív számot?

Specifikáció:

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

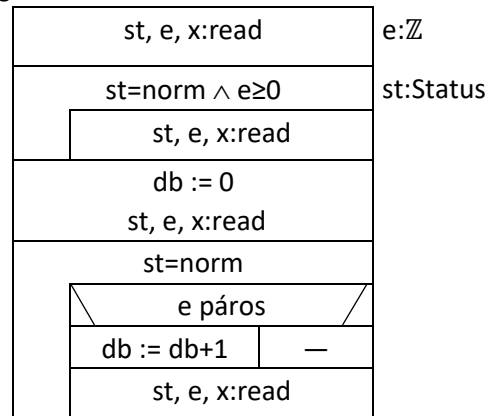
$Ef = (x=x_0)$

$Uf_1 = ((l, \text{elem}, (st', e', x')) = \text{SEARCH}_{e \in x_0} (e < 0) \wedge$
 $\wedge db = \sum_{\substack{e \in (e', x') \\ e \text{ páros}}} 1)$

// másképp:

$Uf_2 = ((e', (st', e', x')) =$
 $\text{SELECT}_{st, e \in x_0} (st = \text{abnorm} \vee e < 0) \wedge$
 $\wedge db = \sum_{\substack{e \in x'} \\ e \text{ páros}}} 1)$

Algoritmus:



Ez utóbbi Uf szerinti megoldás:

Kiválasztás

t:enor(E) ~ x:infile(ℤ) (st,e,x:read)

felt(e) ~ st=abnorm ∨ e < 0

c ~ db

Számlálás

t:enor(E) ~ x:infile(ℤ) „folytatása”

felt(e) ~ e páros

c ~ db

Megjegyzések:

- Az első Uf szerinti megoldás egy lineáris keresés, majd egy számlálás szekvenciája. Lineáris keresés esetén az „elem” változóban lesz az első negatív szám (ha van), e'-ben a ciklusban már előreolvasott következő szám, és x' a fájl maradék része. A számlálás tételnek ezt az előre olvasott e'-vel jelölt adatot, és a fájl maradék részét kell feldolgoznia. Ha nincs a fájlban negatív szám, akkor az olvasás eléri a fájl végét, st'=abnorm miatt a számlálás ciklusa nem hajtódik végre. Mivel a számlálás tételnek nem előfeltétele, hogy a felsoroló nem lehet üres, hibát nem okoz.
- A második megoldásban kiválasztást és számlálást alkalmazunk szekvenciában. Mivel nem garantálható, hogy a fájl tartalmaz negatív számot, a kiválasztás az st státusz változó értékét is figyeli. A kiválasztásnak két eredménye van: a keresett elem, amelyet a specifikáció most e'-vel jelöl (ennek tárolására nincs külön elem változó, mert erre az eredményre nincs szükség, és az e változó úgyis tárolja); valamint a szekvenciális inputfájl felsorolásának aktuális állapota, amelyet az st, e, és x változók pillanatnyi értékei (ezeket jelöli az st', e', x') együttesen jellemez.
- A specifikáció képes a majdani megoldó program három különböző állapotát is megmutatni: például az x kezdőértékét az indulásra kész x₀ jelzi; erre alkalmazzuk a kiválasztást (SELECT), amely következtében az x változó (közbeeső) értéke x' lesz; erre alkalmazzuk a számlálást, amely után az x végértéke az üres sorozat lesz (ezt explicit módon nem írja az utófeltétel).

b) Hány páros szám van az első negatív számot megelőzően, és hány azt követően?

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

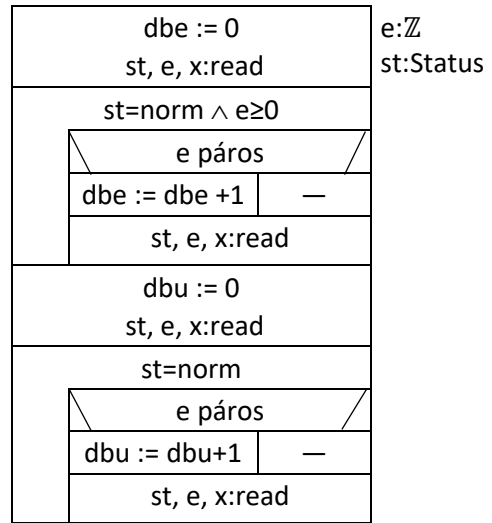
$$t:\text{enor}(E) \sim x:\text{infile}(\mathbb{Z}) (\text{st}, e, x:\text{read})$$

first() helyett next()

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

$$c \sim \text{dbu}$$

Algoritmus:



Megjegyzés: A második számlálás algoritmusának inicializáló részében szereplő read nem a first(), hanem a next() művelet (csak szekvenciális inputfájloknál ezek ugyanazok), hiszen itt folytatni kell a szekvenciális inputfájl már folyamatban levő felsorolását.

c) Hány páros szám van az első negatív számot megelőzően, és hány azt követően azzal együtt?

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

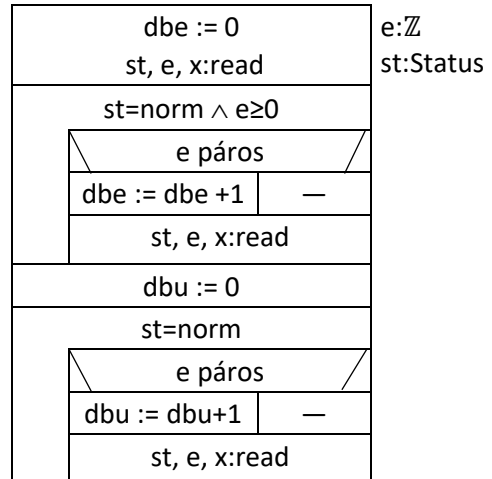
$$t:\text{enor}(E) \sim x:\text{infile}(\mathbb{Z}) (\text{st}, e, x:\text{read})$$

first() nélkül

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

$$c \sim \text{dbu}$$

Algoritmus:



Megjegyzés: A második számlálás algoritmusának inicializáló részében nem szerepel read (sem first(), sem next() nincsen), hiszen nincs szükség előre olvasására ahhoz, hogy az első negatív számot elérjük a fájlból, hiszen azt már előzőleg kiolvastuk.

2. Egy szekvenciális inputfájlban napi átlaghőmérsékleteket tárolunk. Mennyi az első fagypont alatti értéket megelőző napok hőmérsékleteinek átlaga, és mennyi a többi nap (első fagypont alatti értékkel együtt vett) átlaga?

Specifikáció:

$$\begin{aligned}
 A &= (x:\text{infile}(\mathbb{R}), a1, a2:\mathbb{R}) \\
 Ef &= (x=x_0 \wedge |x| \geq 2 \wedge x[1] \geq 0 \wedge \exists i \in [2..|x|]: \\
 &\quad x[i] < 0) \\
 Uf &= ((s1, (st', e', x')) = \sum_{e \in x_0}^{e \geq 0} e \wedge \\
 &\quad (db1, (st', e', x')) = \sum_{e \in x_0}^{e \geq 0} 1 \wedge \\
 &\quad \wedge s2 = \sum_{e \in (e', x')} e \wedge db2 = \sum_{e \in (e', x')} 1 \wedge \\
 &\quad \wedge a1 = s1/db1 \wedge a2 = s2/db2)
 \end{aligned}$$

Két összegzés, feltétel fennállásáig tartanak

$$\begin{aligned}
 t:\text{enor}(E) &\sim x:\text{infile}(\mathbb{R}) \text{ (st,e,x:read)} \\
 &\quad \text{amíg: } e \geq 0 \\
 f(e) &\sim e, 1 \\
 s &\sim s1, db1 \\
 H, +, 0 &\sim (\mathbb{R} +, 0.0), (\mathbb{N} +, 0)
 \end{aligned}$$

Két összegzés

$$\begin{aligned}
 t:\text{enor}(E) &\sim x:\text{infile}(\mathbb{R}) \text{ (st,e,x:read)} \\
 &\quad \text{first() nélkül} \\
 f(e) &\sim e, 1 \\
 s &\sim s2, db2 \\
 H, +, 0 &\sim (\mathbb{R} +, 0.0), (\mathbb{N} +, 0)
 \end{aligned}$$

Algoritmus:

$s1, db1 := 0.0, 0$ $st, e, x: \text{read}$	$e, s1:\mathbb{R}, c1:\mathbb{N}$ $st:\text{Status}$
$st = \text{norm} \wedge e \geq 0$	
$s1, db1 := s1 + e, db1 + 1$ $st, e, x: \text{read}$	
$s2, db2 := 0.0, 0$	$s2:\mathbb{R}, c2:\mathbb{N}$
$st = \text{norm}$	
$s2, db2 := s2 + e, db2 + 1$ $st, e, x: \text{read}$	
$a1, a2 := s1/db1, s2/db2$	

Megjegyzés: A specifikáció két utolsó összegzésének jelölésében az $e \in (e', x')$ felsorolás arra utal, hogy nincs szükség a first() műveletre, mert eleve rendelkezünk már az első elemmel.

3. Egy szekvenciális inputfájlban tároljuk a Föld felszínének egy vonalán – adott távolságokként – mért tengerszint feletti magasságértékeket. Milyen magas a legmagasabb horpadás?

Specifikáció:

$$\begin{aligned}
 A &= (x:\text{infile}(\mathbb{R}), l:\mathbb{L}, \text{max}:\mathbb{R}) \\
 Ef &= (x=x_0)
 \end{aligned}$$

Új specifikáció:

$$\begin{aligned}
 A &= (t:\text{enor}(\mathbb{R} \times \mathbb{R} \times \mathbb{R}), l:\mathbb{L}, \text{max}:\mathbb{R}) \\
 Ef &= (t=t_0) \\
 Uf &= ((l, \text{max}) = \mathbf{MAX}_{(e,a,k) \in t_0} a) \\
 &\quad e > a \wedge k > a
 \end{aligned}$$

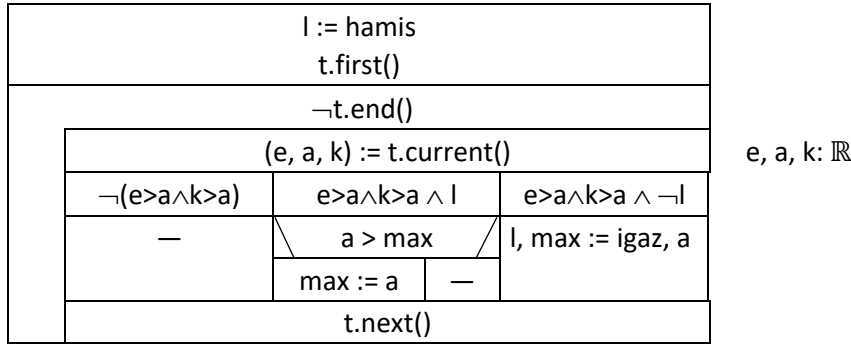
Ötlet:

Kellene egy olyan felsoroló, amelyik mindig egymás után mért három értéket mutatja.

Felt. max. ker.

$$\begin{aligned}
 t:\text{enor}(E) &\sim t:\text{enor}(\mathbb{R} \times \mathbb{R} \times \mathbb{R}), \\
 &\quad \text{ahol } (e, a, k) := t.\text{current}() \\
 f(e) &\sim a \\
 \text{felt}(e) &\sim e > a \wedge k > a \\
 H, > &\sim \mathbb{Z}, >
 \end{aligned}$$

Algoritmus:



Felsoroló

t:enor(ℝ × ℝ × ℝ)

(ℝ × ℝ × ℝ)*	first()	next()	current() : ℝ × ℝ × ℝ	end() : ℒ
x:infile(ℝ) e, a, k: ℝ st : Status	st, e, x:read st, a, x:read st, k, x:read	e := a a := k st, k, x:read	return (e, a, k)	return st=abnorm

4. Gyűjtsük ki egy szekvenciális inputfájlban rendezve tárolt egész számok közül azt, hogy melyik számból hány darab található!

Specifikáció:

A = (x:infile(ℤ), y:outfile(Össz))
Össz = rec(szám:ℤ, db:ℕ)

Ef = (x = x₀ ∧ x↗)
(x↗ azt jelzi, hogy az
x növekedően rendezett)

Új Specifikáció:

A = (t:enor(Össz), y:outfile(Össz))

Ef = (t = t₀)

Uf = (y = t₀) = (y = ⊕_{e∈t₀} <e>)

Összegzés (másolás)

f(e) ~ <e>

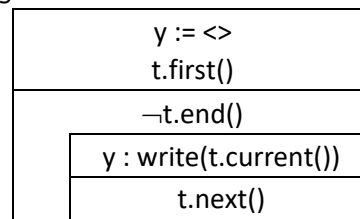
s ~ y

H, +, 0 ~ Össz*, ⊕, <>

Ötlet:

Ha lenne egy olyan felsorolónk, amelyik az eredményt, az összesítéseket tartalmazó rekordokat tudná felsorolni, akkor elég lenne ezeket az output fájlba átmásolni.

Algoritmus:



Felsoroló:

t:enor(Össz) Össz = rec(szám:ℤ, db:ℕ)

Össz*	first()	next()	current() : Össz	end() :ℒ
x : infile(ℤ) dx : ℤ sx : Status akt : Össz vége : ℒ	sx,dx,x:read next()	lásd külön	return akt	return vége

next() művelet

A = (x:infile(ℤ), dx:ℤ, sx:Status, akt:Össz, vége:ℒ)

Ef = (x = x' ∧ x ↗ ∧ dx = dx' ∧ sx = sx')

dx = akt.szám

Uf = (vége = (sx'=abnorm) ∧ (¬vége → akt.szám=dx' ∧ (akt.db, (sx,dx,x)) = ∑_{dx∈(dx',x')} 1))

Megjegyzés: Az összegzésnek két eredménye van: a darabszám (akt.db); és a felsoroló aktuális állapota, amelyet az sx,dx,x változók értékei írnak le a next() művelet végén.

Összegzés

t:enor(E) ~ x:infile(ℤ) (sx,dx,x:read)
first() nélkül, amíg: dx=akt.szám

f(e) ~ 1

s ~ akt.db

H, +, 0 ~ ℕ, +, 0

vége := sx=abnorm	
¬vége	
akt.szám, akt.db := dx, 0	-
sx=norm ∧ dx=akt.szám	
akt.db := akt.db + 1	
sx,dx,x:read	

5. Számoljuk meg egy karakterekből álló szekvenciális inputfájlban a szavakat úgy, hogy a 12 betűnél hosszabb szavakat duplán vesszük figyelembe! (Egy szót szóközök vagy a fájl vége határol.)

Specifikáció:

$$A = (f:\text{infile}(\mathbb{K}), c:\mathbb{N})$$

$$Ef = (f = f_0)$$

Új specifikáció:

$$A = (t:\text{enor}(\mathbb{N}), c:\mathbb{N})$$

$$Ef = (t = t_0)$$

$$Uf = (c = \sum_{e \in t_0} \{2, \text{ ha } e > 12; 1, \text{ ha } e \leq 12\})$$

Összegzés

$$f(e) \sim \{2, \text{ ha } e > 12; 1, \text{ ha } e \leq 12\}$$

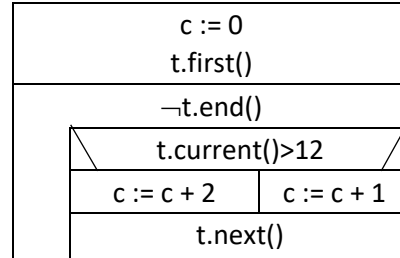
$$s \sim c$$

$$H, +, 0 \sim \mathbb{N}, +, 0$$

Ötlet:

Soroljuk fel csak a szavak hosszait, és ebből az eredményt már könnyen kiszámolhatjuk.

Algoritmus:



Felsoroló:

t:enor(N)

N*	first()	next()	current() : N	end() : L
f : infile(K) ch : K st : Status akt : N vége : L	st,ch,f:read next()	lásd külön	return akt	return vége

next() művelet

$$A = (f:\text{infile}(\mathbb{K}), ch:\mathbb{K}, st:\text{Status}, akt:\mathbb{N}, vége:\mathbb{L})$$

$$Ef = (f = f' \wedge ch = ch' \wedge st = st')$$

$$Uf = ((ch'', (st'', ch'', f'')) = \text{SELECT}_{ch \in (ch'', f'')} (st = \text{abnorm} \vee ch \neq '') \wedge ch \neq '')$$

$$vége = (st'' = \text{abnorm}) \wedge (\neg vége \rightarrow (akt, (st, ch, f)) = \sum_{ch \in (ch'', f'')} 1))$$

Kiválasztás

$$t:\text{enor}(E) \sim f:\text{infile}(\mathbb{K}) (st, ch, f:\text{read})$$

first() nélkül

$$\text{felt}(e) \sim st = \text{abnorm} \vee ch \neq ''$$

Összegzés (megszámolás)

$$t:\text{enor}(E) \sim f:\text{infile}(\mathbb{K}) (st, ch, f:\text{read})$$

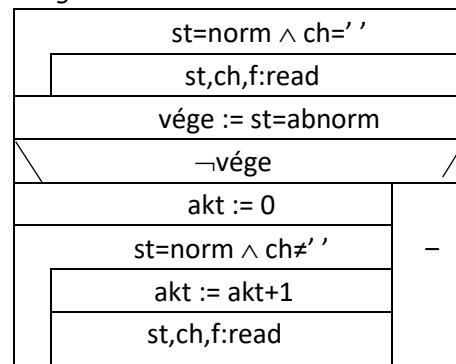
first() nélkül, amíg: ch ≠ ''

$$f(e) \sim 1$$

$$s \sim akt$$

$$H, +, 0 \sim \mathbb{N}, +, 0$$

Algoritmus:



Megjegyzés: Konkrét megvalósításban nemcsak szóközök, hanem mindenféle szeparátor (sorvége-jel, tabulátor-jel, stb.) is lehet a szavak között.