6. táblás gyakorlat feladatai

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

Specifikáció:

$$A = (x:infile(\mathbb{Z}), y:outfile(Össz))$$

 $Össz = rec(szám:\mathbb{Z}, db:\mathbb{N})$
 $Ef = (x = x_0 \land x \nearrow)$
 $(x \nearrow azt jelzi, hogy az$
 $x növekedően rendezett)$

Új Specifikáció:

$$A = (t:enor(\ddot{O}ssz), y:outfile(\ddot{O}ssz))$$

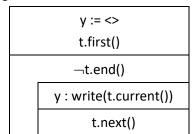
 $Ef = (t = t_0)$
 $Uf = (y = \bigoplus_{e \in t_0} < e >)$

Összegzés (másolás)

Ötlet:

Soroljuk fel az eredménybe szánt rekordokat, és másoljuk output fájlba őket.

Algoritmus:



Felsoroló:

t:enor(Össz)

 $\ddot{O}ssz = rec(num: \mathbb{Z}, count: \mathbb{N})$

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

next() művelet

$$A = (x:infile(\mathbb{Z}), dx:\mathbb{Z}, sx:Status, akt: \ddot{O}ssz, v\acute{e}ge:\mathbb{L})$$

$$Ef = (x = x' \land x \nearrow \land dx = dx' \land sx = sx')$$

$$dx = akt.num$$

$$Uf = (v\acute{e}ge = (sx'=abnorm) \land (\neg v\acute{e}ge \rightarrow akt.sz\acute{a}m=dx' \land (akt.db, (sx,dx,x)) = \sum_{dx \in (dx',x')} 1))$$

Megj: 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 (megszámolás)

t:enor(E)
$$\sim$$
 x:infile(\mathbb{Z}) (sx,dx,x:read)
first() nélkül, felt: dx=akt.szám
f(e) \sim 1
s \sim akt.db
H, +, 0 \sim \mathbb{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	

2. Egy szekvenciális inputfájlban egy banknál számlát nyitott ügyfelek e havi kivét/betét forgalmát (tranzakcióit) tároljuk. Egy tranzakció tartalmazza az ügyfél számlaszámát, a tranzakció dátumát, és az összegét (előjeles egész szám: negatív a kivét, pozitív a betét). A tranzakciók a szekvenciális fájlban számlaszám szerint rendezetten helyezkednek el. Gyűjtsük ki azon számlaszámokat, és az ahhoz tartozó tranzakcióknak az egyenlegét, ahol ez az egyenleg kisebb –100000 Ft-nál!

Specifikáció:

Új specifikáció:

```
A = (t:enor(Egyenleg), y:outfile(Egyenleg))
Ef = (t = t_0)
Uf = (y = \bigoplus_{e \in t_0} <e>)
e.egy<-100000
```

Összegzés (kiválogatás)

Ötlet:

Soroljuk fel számlaszámonként a transzakciók összesített egyenlegeit, hogy ezekből kiválogathassuk a -100000-nél kisebb egyenlegűeket.

Algoritmus:

y := <>				
t.first()				
¬t.end()				
t.current().egy < -100000/				
y : write(t.current()) –				
t.next()				

Felsoroló:

t:enor(Egyenleg) Egyenleg = $rec(száml: S, egy: \mathbb{Z})$

1 07 07	0,	,	, 0, ,	
Egyenleg*	first()	next()	current() : Egyenleg	end() : $\mathbb L$
x : infile(Ügyfél) dx : Ügyfél sx : Status akt : Egyenleg vége : L	sx,dx,x:read next()	lásd külön	return akt	return vége

 $Ugyfél = rec(száml: \mathbb{S}, dátum: \mathbb{S}, össz: \mathbb{Z})$

next() művelet

```
A = ( x:infile( \ddot{\cup} gyf\acute{e}l), dx: \ddot{\cup} gyf\acute{e}l, sx: Status, akt: Egyenleg, v\acute{e}ge: \mathbb{L} ) Ef = ( x = x' \land x \nearrow_{sz\acute{a}ml} \land dx = dx' \land sx = sx' ) Uf = (v\acute{e}ge=(sx'=abnorm) \land (\neg v\acute{e}ge \rightarrow akt.sz\acute{a}ml = dx'.sz\acute{a}ml = akt.sz\acute{a}ml \land (akt.egy, (sx,dx,x)) = \sum_{dx \in (dx',x')} dx.\ddot{o}ssz ) )
```

Összegzés

_		
vége := sx=abnorm		
	⊸vége	
akt	.száml, akt.egy := dx.száml, 0	
sx=norm ∧ dx.száml=akt.száml		
	akt.egy := akt.egy + dx.össz	
	sx,dx,x:read	

3. 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:

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

 $Ef = (x = x_0)$

Új specifikáció:

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

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

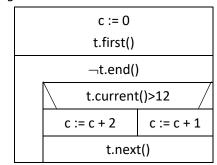
Összegzés

f(e)
$$\sim$$
 {2, ha e>12; 1, ha e≤12}
s \sim c
H, +, 0 \sim N, +, 0

Ötlet:

Soroljuk fel a szavak hosszait, és ilyen módon az eredményt már könnyen kiszámolhatjuk.

Algoritmus:



Felsoroló:

t:enor(ℕ)

N*	first()	next()	current() : N	end() $: \mathbb{L}$
x:infile(K) dx:K sx:Status akt:N vége:L	sx,dx,x:read next()	lásd külön	return akt	return vége

next() művelet

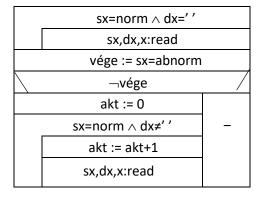
$$\begin{split} A &= (\text{ x:infile}(\mathbb{K}), \, \mathsf{dx} : \mathbb{K}, \, \mathsf{sx} : \mathsf{Status}, \, \mathsf{akt} : \mathbb{N}, \, \mathsf{v\'ege} : \mathbb{L} \,) \\ Ef &= (\, \mathsf{x} = \mathsf{x'} \, \land \, \mathsf{dx} = \mathsf{dx'} \, \land \, \mathsf{sx} = \mathsf{sx'} \,) \\ Uf &= (\, (\mathsf{dx''}, (\mathsf{sx''}, \mathsf{dx''}, \mathsf{x''})) = \mathsf{SELECT}_{\mathsf{dx} \in (\mathsf{dx'}, \mathsf{x'})} \, (\, \mathsf{sx} = \mathsf{abnorm} \, \lor \, \mathsf{dx} \neq' \, ') \, \land \\ & \mathsf{v\'ege} = (\mathsf{sx''} = \mathsf{abnorm}) \, \, \land \, \, (\, \neg \mathsf{v\'ege} \, \to \, (\mathsf{akt}, \, (\mathsf{sx}, \mathsf{dx}, \mathsf{x})) = \sum_{\mathsf{dx} \in (\mathsf{dx''}, \mathsf{x''})} 1 \, \,) \, \,) \end{split}$$

Megj: A kiválasztásnak két eredménye van: a keresett szóköz (dx"), amelyhez nem vezetünk be külön output változót; és a felsoroló aktuális állapota, amelyet az (sx",dx",x") hármas ír le. (Nem okozna félreértést, ha csak az (sx",dx",x")-t tüntetnénk fel a kiválasztás eredményeként.)

Kiválasztás

t:enor(E)
$$\sim$$
 x:infile(\mathbb{K}) (sx,dx,x:read)
first() nélkül
felt(e) \sim sx=abnorm \vee dx \neq ''

Összegzés (megszámolás)



4. Másoljuk át karakterenként egy szekvenciális inputfájl szövegét egy szekvenciális outputfájlba úgy, hogy a szavak között csak egyetlen szóközt tartunk meg!

Specifikáció:

$$A = (x:infile(\mathbb{K}), y:infile(\mathbb{K}))$$

 $Ef = (x = x_0)$

Új specifikáció:

$$A = (t:enor(S), y:infile(K))$$

$$Ef = (t = t_0)$$

$$Uf = (y = \bigoplus_{e \in t_0} (e \bigoplus <' '>)$$

Összegzés (összefűzés)

$$f(e) \qquad \begin{array}{ccc} \sim & e \oplus <' \ '> \\ s & \sim & y \\ H, +, 0 & \sim & \mathbb{K}^*, \oplus, <> \\ \end{array}$$

Ötlet:

Soroljuk fel a szavakat, és írjuk ki úgy őket, hogy egyetlen szóközt teszünk közéjük.

Algoritmus:

y := <>				
t.first()				
¬t.end()				
y : write(t.current())				
y : write(' ')				
t.next()				

Felsoroló

t:enor(S)

S*	first()	next()	current() : $\mathbb S$	end() $: \mathbb{L}$
x:infile(K) dx:K sx:Status akt:S vége:L	sx,dx,x:read szóközt_átlép() next()	lásd külön	return akt	return vége

szóközt_átlép() művelet

$$A = (x:infile(\mathbb{K}), dx:\mathbb{K}, sx:Status)$$

$$Ef = (x = x' \land dx = dx' \land sx = sx')$$

$$Uf = ((sx,dx,x) = SELECT_{dx \in (dx',x')}(sx=abnorm \lor dx \neq''))$$

Kiválasztás

t:enor(E)
$$\sim$$
 x:infile(\mathbb{K}) (sx, dx, x:read)
first nélkül
felt(e) \sim sx=abnorm \vee dx \neq ''

sx=norm \wedge dx=' '
sx,dx,x:read

next() művelet

$$A = (x:infile(\mathbb{K}), dx:\mathbb{K}, sx:Status, akt:\mathbb{S}, vége:\mathbb{L})$$

$$Ef = (x = x' \land dx = dx' \land sx = sx' \land (sx=norm \rightarrow dx \neq' ')) \qquad \qquad dx \neq' '$$

$$Uf = (vége = (sx'=abnorm) \land (\neg vége \rightarrow (akt, (sx'', dx'', x'')) = \bigoplus_{dx \in (dx', x')} < dx >$$

$$\land (sx, dx, x) = SELECT_{dx \in (dx', x')} (sx=abnorm \lor dx \neq' ')))$$

Összegzés (összefűzés)

t:enor(E)
$$\sim$$
 x:infile(\mathbb{K}) (sx,dx,x:read)
first() nélkül, felt: dx \neq ' 'f(e) \sim
s \sim akt
H, +, 0 \sim \mathbb{K}^* , \oplus , <>

vége := sx=abnorm		
⊸vége		
akt := <>		
sx=norm ∧ dx≠′′		
akt := akt ⊕ <dx></dx>		
sx,dx,x:read		
szóközt_átlép()		

Házi feladat:

Egy étteremben a pincérek által felvett rendeléseket egy szöveges állományban tartják nyilván az ételek neve, azon belül a rendelések időpontja szerint rendezett formában. Feltehetjük, hogy a fájl nem üres. Egy rendelés az asztal sorszámából, a rendelt étel nevéből (sztring), a rendelés időpontjából (sztring), rendelt adagok számából (természetes szám), egy adag árából (természetes szám) áll. Melyik étel hozta az étteremnek a legtöbb bevételt (összesített darab*egységár)?