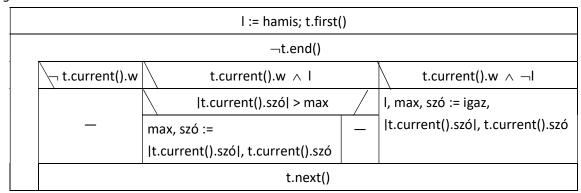
7. táblás gyakorlat feladatai

1. Egy karakterekből álló szekvenciális inputfájlban melyik az a leghosszabb szó, amelyik tartalmazza a 'w' betűt? (Egy szót szóközök vagy a fájl vége határol.)

```
Specifikáció:
                                                              Ötlet:
   A = (x:infile(\mathbb{K}), I:\mathbb{L}, szó:\mathbb{S})
                                                                  Soroljuk fel a szavakat, azzal az
                                                                  információval együtt, hogy van-e
    Ef = (x = x_0)
                                                                  bennük 'w' betű.
Új Specifikáció:
                                                              Feltételes maximum keresés
                                                                 f(e)
                                                                                ~ [e. szó]
   A = (t:enor(Szó), I:\mathbb{L}, szó:\mathbb{S})
                                                                 felt(e)
                                                                                 ~ e.w
                   Szó = rec(szó:S, w:L)
                                                                 H, >
                                                                                ~ ℕ,>
    Ef = (t = t_0)
   Uf = ( ((I, max, elem) = MAX_{e \in t_0} | e.szó|) \land (I \rightarrow szó=elem.szó))
```

Algoritmus:



Felsoroló:

t:enor(Szó)	Szó =	rec(szó:\$, w:L)	1	
Szó*	first()	next()	current() : Szó	end() : $\mathbb L$
x:infile(K) dx: K sx:Status akt:Szó vége: L	sx,dx,x:read next()	lásd külön	return akt	return vége

next() művelet

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

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

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

$$\land vége=(sx''=abnorm)$$

$$dx \neq''$$

$$\land (\neg vége \rightarrow (akt.szó, (sx,dx,x)) = \bigoplus_{dx\in(dx'',x'')} < dx > \land (akt.w, (sx,dx,x)) = \bigvee_{dx\in(dx'',x'')} dx='w'))$$

Megj: A "van 'w' karakter egy szóban" részfeladatot lineáris keresés helyett összevagyolással számoljuk ki, mert így is, úgy is végig kell menni a teljes aktuális szón (lásd a szó betűinek összefűzését), amellyel az összevagyolást össze lehet vonni közös felsorolásba (ciklusba).

Kiválasztás

 $t:enor(E) \; ^\sim \; \; x:infile(\mathbb{K}) \; (sx,dx,x:read)$

first() nélkül

felt(e) ~ sx=abnorm ∨ dx≠''

Két összegzés (összefűzés és összevagyolás)

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

f(e) \sim (<dx>, dx='w')

s ~ (akt.szó, akt.w)

 $H, +, 0 \sim (\mathbb{K}^*, \mathbb{L}), (\bigoplus, \vee), (<>, hamis)$

sx=norm ∧ dx=' '	
sx,dx,x:read	
vége := sx=abnorm	
¬vége	
akt.szó, akt.w := <>, hamis	
sx=norm ∧ dx≠' '	_
akt.szó, akt.w :=	
akt.szó \bigoplus <dx>, akt.w \vee (dx='w')</dx>	
sx,dx,x:read	

2. Egy szekvenciális inputfájlban a vadászok eredményeit (vadász neve, vadászat dátuma, zsákmány fajtája, zsákmány súlya) tároljuk a vadászok neve, azon belül a vadászat dátuma szerint rendezetten. Igaz-e, hogy minden vadász valamelyik vadászatán lőtt medvét?

Specifikáció:

```
A = (x:infile(Trófea), I:L)

Trófea =

rec(név:S, dátum:S, fajta:S, súly:N)

Ef = (x=x_0 \land x \land (név, dátum))
```

Optimista lineáris keresés

felt(e) ~ e

Új specifikáció:

 $A = (t:enor(\mathbb{L}), I:\mathbb{L})$ $Ef = (t=t_0)$

 $Uf = (I = \forall SEARCH_{e \in t_0} e)$

Ötlet:

Soroljunk fel annyi logikai értéket ahány vadász van: egy logikai érték akkor igaz, ha a vadász lőtt medvét.

Algoritmus:

_	
	l := igaz
	t.first()
	I ∧ ¬t.end()
	l := t.current()
	t.next()

Felsoroló:

$t:enor(\mathbb{L})$

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

Trófea = rec(név: \mathbb{S} , dátum: \mathbb{S} , fajta: \mathbb{S} , súly: \mathbb{N})

next() művelet

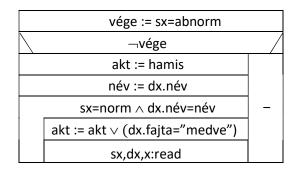
$$A = (x:infile(Trófea), dx:Trófea, sx:Status, akt: \mathbb{L}, vége: \mathbb{L})$$

$$Ef = (x = x' \land dx = dx' \land sx = sx') \qquad \qquad dx.név = dx'.név$$

$$Uf = (vége = (sx'=abnorm) \land (\neg vége \rightarrow (akt, (sx,dx,x)) = \bigvee_{dx \in (dx',x')} dx.fajta="medve"))$$

Összegzés (összevagyolás)

t:enor(E)
$$\sim$$
 x:infile(Trófea) (sx,dx,x:read) first() nélkül, felt: dx.név=dx'.név f(e) \sim dx.fajta="medve" s \sim akt H, +, 0 \sim L , \vee , $<>$



3. Egy szekvenciális inputfájlban a vadászok eredményeit (vadász neve, vadászat dátuma, zsákmány fajtája, zsákmány súlya) tároljuk a vadászok neve, azon belül a vadászat dátuma szerint rendezetten. Hány olyan vadász volt, aki minden vadászatán lőtt nyulat, és a zsákmányainak összsúlya meghaladta a 250 kilogrammot.

Specifikáció:

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

 $Trófea =$
 $rec(név:\mathbb{S}, dátum:\mathbb{S}, fajta:\mathbb{S}, súly:\mathbb{N})$
 $Ef = (x=x_0 \land x \nearrow (név, dátum))$

Új specifikáció:

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

$$Vadász = rec(név: \mathbb{S}, nyúl: \mathbb{L}, összsúly: \mathbb{N})$$

$$Ef = (t=t_0)$$

$$Uf = (c = \sum_{e \in t_0} 1)$$

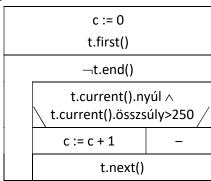
$$e.nyúl \wedge e. összsúly>250$$

Számlálás

Ötlet:

Soroljuk fel a vadászok neveit azzal az információval együtt, hogy lőttek-e minden vadászatukon nyulat, és mennyi volt a zsákmányaiknak az összsúlya.

Algoritmus:



Felsoroló:

Ötlet: A t:enor(Vadász) felsoroló megvalósításához tegyük fel, hogy rendelkezünk olyan y:enor(Összesítés) felsorolóval, amely a vadászoknak az egyes vadászatokon elért összesített eredményeit sorolja fel a vadász neve és a vadászat dátuma szerint növekedően rendezetten: lőtt-e egy adott vadászatán nyulat, és mennyi volt ott a zsákmányai súlyának összege.

t:enor(Vadász) Vadász = rec(név: S, nyúl: L, összsúly: N)

Vadász*	first()	next()	current() : Vadász	end() : $\mathbb L$
y : enor(Összesítés) akt : Vadász vége : L	y.first() next()	lásd külön	return akt	return vége

Összesítés = rec(név: S, dátum: S, lőtt: L, össz: N)

next() művelet

$$A = (y:enor(\ddot{O}sszesítés), akt:Vadász, vége: \mathbb{L})$$

$$Ef = (y = y' \land y \nearrow (név, dátum))$$

$$Uf = (vége = y'.end() \land (\neg vége \rightarrow akt.név = y'.current().név$$

$$e.név = akt.név$$

$$\land (akt.nyúl, y) = \bigwedge_{e \in (y'.current(),y')} e.lőtt$$

$$e.név = akt.név$$

$$\land (akt.\ddot{O}sszsúly, y) = \sum_{e \in (y'.current(),y')} e.\ddot{O}ssz$$
))

Két összegzés (éselés és összeadás)

közös felsorolóval

t:enor(E) ~ y:enor(Összesítés)

first() nélkül

felt: y.current().név=akt.név

f(e) ~ y.current().lőtt

s ~ akt.nyúl

H, +, 0 \sim L, \wedge , igaz

f(e) ~ y.current().össz s ~ akt.összsúly

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

	vége := y.end()	
	¬vége	
·	akt := (y.current().név, igaz, 0)	
	¬y.end() ∧ y.current().név=akt.név	_
	akt.nyúl := akt.nyúl ∧ y.current().lőtt	
	akt.összsúly :=	
	akt.összsúly + y.current().össz	
	y.next()	

Felsoroló:

y:enor(Összesítés) Összesítés = rec(név: S, dátum:S, lőtt:L, össz:N)

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

Trófea = rec(név:S, dátum:S, fajta:S, súly:N)

next() művelet

A = (x:infile(Trófea), dx:Trófea, sx:Status, akt:Összesítés, vége:L)

$$Ef = (x = x' \land x \nearrow (\text{n\'ev,d\'atum}) \land dx = dx' \land sx = sx')$$

 $Uf = (vége = (sx'=abnorm) \land (\neg vége \rightarrow akt.név=dx'.név \land akt.dátum=dx'.dátum \land$

dx,név=akt.név ∧ dx,dátum=akt.dátum

$$\land$$
 (akt.lőtt, (sx,dx,x)) = $\bigvee_{dx \in (dx',x')} dx.$ fajta="nyúl" \land

dx,név=akt.név \(\times \, dx, dátum=akt. dátum

$$\wedge$$
 (akt.össz, (sx,dx,x)) = $\sum_{dx \in (dx',x')} dx.súly$)

Két összegzés (vagyolás és összeadás)

közös felsorolóval

t:enor(E) ~ x:infile(Eredmény)

sx,dx,x:read, first() nélkül,

felt: dx.név=akt.név

∧ dx.dátum=akt.dátum

f(e) ~ dx.fajta="nyúl"

s ~ akt.lőtt

H, +, 0 \sim L, \vee , hamis

f(e) ~ dx.súly

s ~ akt.össz

 $H, +, 0 \sim N, +, 0$

vége := sx=abnorm	
¬vége	
akt := (dx.név, dx.dátum, hamis, 0)	
sx=norm ∧ dx.név=akt.név	_
∧ dx.dátum=akt.dátum	
akt.lőtt := akt.lőtt v (dx.fajta="nyúl")	
akt.össz := akt.össz + dx.súly	
sx,dx,x:read	