***5. táblás gyakorlat feladatai***

1. Egy szekvenciális inputfájlban egyes kaktuszfajtákról ismerünk néhány adatot: név, őshaza, virágszín, méret.
	1. Számoljuk meg a piros virágú kaktuszokat!

*Specifikáció*:

*A* = ( x:infile(Kaktusz), db:ℕ )
Kaktusz=rec(név:𝕊, szín:𝕊, ős:𝕊, méret:ℕ)
*Ef* = ( x=x0 )
*Uf* = ( db = $\sum\_{e\in x\_{0}}^{}1$)
 e.szín=”piros”

*Számlálás*

t:enor(E) ~ x:infile(Kaktusz) (st,e,x:read)

felt(e) ~ e.szín=”piros”

c ~ db

*Algoritmus:*

|  |  |
| --- | --- |
| db := 0 st, e, x:read | st:Statuse:Kaktusz |
| st = norm |  |
|  | e.szín=”piros”  |  |
| db := db + 1 | ― |  |
| st, e, x:read |  |

1. Igaz-e, hogy minden kaktusz virága piros?

*Specifikáció*:

*A* = ( x:infile(Kaktusz), l:𝕃 )

Kaktusz=rec(név:𝕊, szín:𝕊, ős:𝕊, méret:ℕ)
*Ef* = ( x=x0)
*Uf* = ( l,(st,e,x)= ∀**SEARCH**e∊x0 e.szín=”piros” )
Az utófeltétel azt fejezi ki, hogy az eldöntés
tétel nem biztos, hogy a fájlt végig olvassa,
hamarabb leállhat.

*Opt. linker*

t:enor(E)~ x: infile(Kaktusz) (st,e,x:read)

felt(e) ~ e.szín=”piros”

*Algoritmus*:

|  |  |
| --- | --- |
| l := igaz st, e, x:read | st:Statuse:Kaktusz |
| l ∧ st=norm |  |
|  | l := e.szín=”piros” |  |
| st, e, x:read |  |

1. Válogassuk ki egy szekvenciális outputfájlba a piros virágú kaktuszok neveit!

*Specifikáció*:

*A* = ( x:infile(Kaktusz), y:outfile(𝕊) )
 Kaktusz=rec(név:𝕊, szín:𝕊, ős:𝕊, méret:ℕ)
*Ef* = ( x=x0)
*Uf* = ( y = ⊕e∊x0 <e.név> )
 e.szín=”piros”

*Összegzés* (kiválogatás)

t:enor(E) ~ x:infile(Kaktusz) (st,e,x:read)

f(e) ~ <e.név> ha e.szín=”piros”

s ~ y

H, +, 0 ~ 𝕊\*, ⊕, <>

*Algoritmus:*

|  |  |
| --- | --- |
| y:= <> st, e, x:read | st:Statuse:Kaktusz |
| st = norm |  |
|  | e.szín=”piros” |  |
| y:write(e.név) | ― |  |
| st, e, x:read |  |

1. Válogassuk ki egy szekvenciális outputfájlba a piros virágú kaktuszok, egy másikba a mexikói őshazájú kaktuszok neveit!

*Specifikáció*:

*A* = ( x:infile(Kaktusz), y, z :outfile(𝕊) )
 Kaktusz=rec(név:𝕊, szín:𝕊, ős:𝕊, méret:ℕ)
*Ef* = ( x=x0)
*Uf* = ( y=⊕e∊x0<e.név> ∧ z=⊕e∊x0<e.név> )
 e.szín=”piros” e.ős=”Mexikó”

*2 sima- vagy 1 duplaösszegzés* (kiválogatás)

t:enor(E) ~ x:infile(Kaktusz) (st,e,x:read)

f1(e) ~ <e.név> ha e.szín=”piros”

f2(e) ~ <e.név> ha e.ős=”Mexikó”

s ~ y, z

H, +, 0 ~ 𝕊\*, ⊕, <>

Algoritmus:

|  |  |
| --- | --- |
| y, z := <>, <> st, e, x:read | st:Statuse:Kaktusz |
| st = norm |  |
|  | e.szín=”piros” |  |
| y:write(e.név) | ― |  |
| e.ős=”Mexikó” |  |
| z:write(e.név) | ― |  |
| st, e, x:read |  |

1. Melyik a legnagyobb piros virágú kaktusz neve?

*Specifikáció*:

*A* = ( x:infile(Kaktusz), l:𝕃, név:𝕊 ) Kaktusz=rec(név:𝕊, szín:𝕊, ős:𝕊, méret:ℕ)
*Ef* = ( x=x0)
*Uf* = ( (l, max, elem)= **MAX**e∊x0 e.méret ∧ l → név = elem.név ) max:ℤ, elem:Kaktusz
 e.szín=”piros”

*Felt. max. ker.*

t:enor(E) ~ x:infile(Kaktusz) (st,e,x:read)

f(e) ~ e.méret

felt(e) ~ e.szín=”piros”

H, > ~ ℕ, >

*Algoritmus*:

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

|  |
| --- |
| l := hamisst, e, x:read |
|  st=norm |
|  | e.szín≠”piros” | e.szín=”piros” ∧ l | e.szín=”piros”∧ ¬l | st:Status |
| ― |  e.méret > max | l, max, név := igaz, e.méret, e.név | e:Kaktusz |
| max, név := 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.

1. Adott egy egész számokat tartalmazó szekvenciális inputfájl.
2. Hány páros szám előzi meg benne az első negatív számot?

*Specifikáció*:

*A* = ( x:infile(ℤ), db:ℕ )
*Ef* = ( x=x0 )
*Uf* = ( db = $\sum\_{e\in x\_{0}}^{e\geq 0}1 $)
 e páros

*Számlálás, feltétel fennállásáig*

t:enor(E) ~ x:infile(ℤ) (st,e,x:read)

 amíg: e≥0

felt(e) ~ e páros

c ~ db*Algoritmus*:

|  |  |
| --- | --- |
| db := 0 st, e, x:read | e:ℤ st:Status |
| st=norm ∧ e≥0  |  |
|  | e páros |  |
|  | db := db+1 | ― |  |
|  | st, e, x:read |  |

*Megjegyzés*: A specifikációban a programozási tétel kulcsszava (Σ) feletti extra feltétel jelzi, hogy meddig tartson a felsorolás.

Az utófeltételben kifejezhetjük, hogy a feldolgozás nem biztos, hogy a fájl végéig tart:

*Uf* = ( db,(st,e,x) = $\sum\_{e\in x\_{0}}^{e\geq 0}1 $)
 e páros