

## OEP 5. gyakorlat

*Visszavezetés szekvenciális inputfájl felsorolásával*

Témakör: szekvenciális inputfájl feldolgozása programozási tételek összetételével

1. Keressük meg egy pozitív egész számokat tartalmazó nem üres szekvenciális inputfájlban a legnagyobb számot, és döntsük el azt is, hogy van-e páros szám. (Három megoldást is mutatunk.)

*Első változat:*

*Specifikáció:*

$A = (x:\text{infile}(\mathbb{N}^+), l:\mathbb{L}, \text{nagy}:\mathbb{N})$

$Ef = (x=x' \wedge |x|\geq 1)$

$Uf = (\text{nagy} = \mathbf{MAX}_{e\in x'} e \wedge l = \mathbf{SEARCH}_{e\in x'} (e \text{ páros}))$

*Maxkiv és linker összevonva*

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

$f(e) \sim e, e \text{ páros}$

$s \sim \text{nagy}, l$

### Algoritmus:

st,e,x : read		e:ℕ st:Status
nagy, l := e, e páros		
st,e,x : read		
st=norm		
nagy < e		
nagy := e	—	
e páros		
l := igaz		—
st,e,x : read		

### ÉS táblázata

IGAZ	és	IGAZ	⇒	<b>IGAZ</b>
IGAZ	és	HAMIS	⇒	<b>HAMIS</b>
HAMIS	és	IGAZ	⇒	<b>HAMIS</b>
HAMIS	és	HAMIS	⇒	<b>HAMIS</b>

### VAGY táblázata

IGAZ	vagy	IGAZ	⇒	<b>IGAZ</b>
IGAZ	vagy	HAMIS	⇒	<b>IGAZ</b>
HAMIS	vagy	IGAZ	⇒	<b>IGAZ</b>
HAMIS	vagy	HAMIS	⇒	<b>HAMIS</b>

Második változat:

$Uf = ( \text{nagy} = \bigtriangleup_{e \in x'} e \wedge l = \bigvee_{e \in x'} (e \text{ páros}) )$

ahol:

$\bigtriangleup: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}, \bigtriangleup(a,b) := \max(a,b), 0$

$\bigvee: \mathbb{L} \times \mathbb{L} \rightarrow \mathbb{L}, \bigvee(a,b) := a \vee b, \text{ hamis}$

Két összegzés összevonva

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

$f(e) \sim e, e \text{ páros}$

$s \sim \text{nagy}, l$

$H, +, 0 \sim (\mathbb{N}, \bigtriangleup, 0), (\mathbb{L}, \bigvee, \text{hamis})$

Algoritmus:

nagy, l := 0, hamis	
st,e,x : read	
st=norm	
	nagy := max(nagy, e)
	l := l ∨ e páros
	st,e,x : read

e:ℕ  
st:Status

Harmadik változat:

$$Uf = ( (nagy, l) = (\bigtriangleup V)_{e \in x'} (e, e \text{ páros}) )$$

ahol

$$(\bigtriangleup, V): (\mathbb{N}, \mathbb{L}) \times (\mathbb{N}, \mathbb{L}) \rightarrow (\mathbb{N}, \mathbb{L}), (0, \text{hamis})$$

$$(\bigtriangleup, V)((a, p), (b, q)) := (\max(a, b), (a \vee b))$$

Egy (dupla) összegzés

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

$f(e) \sim e, e \text{ páros}$

$s \sim \text{nagy}, l$

$H, +, 0 \sim ((\mathbb{N}, \mathbb{L}), (\bigtriangleup, V), (0, \text{hamis}))$

*Algoritmus:*  $e: \mathbb{N} \text{ st: Status}$

$\text{nagy}, l := 0, \text{hamis}$

$\text{st}, e, x : \text{read}$

$\text{st} = \text{norm}$

$\text{nagy}, l := \max(\text{nagy}, e), l \vee e \text{ páros}$

$\text{st}, e, x : \text{read}$

2. Egy horgászversenyen a horgászok eredményét egy szekvenciális inputfájlban rögzítették. A fájl egy eleme egy horgász nevét és a halfogásainak sorozatát tartalmazza. Egy fogás egy időpontból, a kifogott hal fajtájának nevéből, a hal súlyából (kg) és hosszából (cm) áll. Keressünk olyan horgászt, aki az 50 cm-esnél hosszabb pontyokból legalább 10 kilogramnyit fogott.

Specifikáció:

$A = ( f:\text{infile}(\text{Horgász}), l:\mathbb{L}, \text{név}:\mathbb{S} )$

$\text{Horgász} = \text{rec}( \text{név}:\mathbb{S}, \text{fogás}:\text{Fogás}^* )$   
 $\text{Fogás} = \text{rec}( \text{idő}:\mathbb{S} , \text{hal}:\mathbb{S}, \text{súly}:\mathbb{R}, \text{hossz}:\mathbb{R} )$

$Ef = ( f=f_0 )$

$Uf = ( (l, \text{elem}) = \text{SEARCH}_{h \in f_0} \text{pontysúly}(h.\text{fogás}) \geq 10.0 \wedge l \rightarrow (\text{név} = \text{elem.név}) )$

ahol  $\text{pontysúly}(h.\text{fogás}) = \sum_{\substack{e \in h.\text{fogás} \\ e.\text{hal} = \text{"ponty"} \wedge e.\text{hossz} \geq 50.0}} e.\text{súly}$

Lineáris keresés

$t:\text{enor}(E) \quad \sim f:\text{infile}(\text{Horgász}) \quad (st, h, f:\text{read})$   
 $\text{felt}(e) \quad \sim \text{pontysúly}(h.\text{fogás}) \geq 10.0$

ahol  $\text{pontysúly} : \text{Fogás}^* \rightarrow \mathbb{R}$

Algoritmus:

$l := \text{hamis}$ $st, h, f : \text{read}$		$st:\text{Status}$ $h:\text{Horgász}$
$\neg l \wedge st = \text{norm}$		
<div><div><math>\text{pontysúly}(h.\text{fogás}) \geq 10.0</math></div></div>		
$l, \text{név} := \text{igaz}, h.\text{név}$	$st, h, f : \text{read}$	

Részfeladat:  $s := \text{pontysúly}(x)$

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

$Ef = (x=x_0)$

$Uf = (x=x_0 \wedge s = \sum_{\substack{e \in x_0 \\ e.\text{hal}=\text{"ponty"} \wedge e.\text{hossz} \geq 50.0}} e.\text{súly})$

Összegzés (feltételes összegzés)

$t:\text{enor}(E) \sim e \text{ in } x$

$f(e) \sim e.\text{súly}$

ha  $e.\text{hal}=\text{"ponty"}$

$\wedge e.\text{hossz} \geq 50.0$

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

Algoritmus:

s := 0.0		e:Fogás
e in x		
e.hal="ponty" ∧ e.hossz≥50.0		
s := s+ e.súly	—	

Megjegyzés:

A feltételes összegzés itt is lehetne a fájlból történő olvasás része, azaz az olvasás a horgász nevén kívül csak ezt az összeget adná meg, a teljes fogás listát nem.

3. Számoljuk ki egy számítástechnikai szaküzlet napi bevételét az aznapi forgalom alapján. A forgalmat a kiadott számlák mutatják, amelyeket egy szöveges állományban (szekvenciális inputfájl) rögzítettek. Az állomány minden sora egy-egy számla adatait tartalmazza: a vásárló nevét és az általa vásárolt termékek (cikkszám és ár párok) sorozatát.

### 1. megoldás

#### Specifikáció:

$A = ( f:\text{infile}(\text{Szám}la), \text{bevét}:\mathbb{N} )$

$\text{Szám}la = \text{rec}(\text{név}:\mathbb{S}, \text{lista}:\text{Áru}^*) \quad \text{Áru} = \text{rec}(\text{cikkszám}:\mathbb{S}, \text{ár}:\mathbb{N})$

$Ef = ( f=f_0 )$

$Uf = ( \text{bevét} = \sum_{sz \in f_0} \text{össz}(sz.\text{lista}) ) \quad \text{ahol } \text{össz}(sz.\text{lista}) = \sum_{e \in sz.\text{lista}} e.\text{ár}$

#### Összegzés

$t:\text{enor}(E) \quad \sim f:\text{infile}(\text{Szám}la) \quad (st, sz, f:\text{read})$

$f(e) \quad \sim \text{össz}(sz.\text{lista})$   
 ahol  $\text{össz} : \text{Áru}^* \rightarrow \mathbb{N}$

$s \quad \sim \text{bevét}$

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

Algoritmus:

bevét := 0		st:Status sz:Számla
st, sz, f : read		
st = norm		
bevét := bevét + össz(sz.lista)		
st, sz, f : read		

Részfeladat:

sum := össz(x)

$A = (x : \text{Áru}^*, \text{sum} : \mathbb{N})$

$Ef = (x = x_0)$

$Uf = (\text{sum} = \sum_{p \in x_0} p.\text{ár})$

Összegzés

$t:\text{enor}(E) \sim p \text{ in } x$

$f(e) \sim p.\text{ár}$

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

Algoritmus:

sum := 0		p : Árú
p in x		
	sum := sum+ p.ár	



## 2. megoldás (kihagyható)

### Specifikáció:

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

$Ef = ( f=f_0 )$

$Uf = ( \text{bevét} = \sum_{\text{sum} \in f_0} \text{sum} )$

Mintha az eredeti szekvenciális inputfájl helyett egy olyan fájlunk lenne, amely csak a számlák végösszegeit tartalmazná: úgy kell olvasni az eredeti állományt, hogy egy olvasás a soron következő sor számlájában szereplő árucikkei árának összegét adja meg.

### Összegzés

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

$f(e) \sim \text{sum}$

$s \sim \text{bevét}$

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

### Algoritmus:

bevét := 0		st : Status sum : ℕ
st, sum, f : read		
st = norm		
bevét := bevét + sum		
st, sum, f : read		