I. Tudjuk, hogy van olyan f(i) és g(j) érték, hogy f(i)+g(j)=e, ahol f, $g: \mathbb{N} \to \mathbb{R}$ és $e \in \mathbb{R}$. Adjunk meg ilyen i-t és j-t!

Gondolatban rendezzük el az f(i)+g(j) értékeket egy végtelen kiterjedésű 0-tól kezdődő indexelésű mátrixba úgy, hogy a mátrix i-dik sorának j-edik eleme az f(i)+g(j) érték legyen. A feladat megoldásához ezen mátrix elemeit kell bejárni, de úgy, hogy véges lépésen belül bármelyik eleméhez el lehessen jutni. Erre a klasszikus (sorfolytonos) bejárás nem alkalmas, mert az soha nem fog a mátrix első (0.) során kívül más elemhez eljutni, lévén a sor végtelen hosszú. Definiáljunk egy olyan egyedi felsorolást erre a mátrixra, amely a mellékátlók mentén járja be az elemeket. Egy t felsorolótól tehát index párokat várunk (i,j)et (i,j)e

A feladat specifikációja és absztrakt programja egyedi felsorolóval:

$$\begin{split} &A = (\text{ t:enor}(\mathbb{N} \times \mathbb{N}), \text{ e} : \mathbb{R}, \text{ i} : \mathbb{N}, \text{j} : \mathbb{N} \text{)} \\ &\text{Ef=} (\text{ e=e'}) \\ &\text{Uf=} (\text{ e=e'} \wedge (\text{i,j}) = \text{SELECT}_{(\text{i,j}) \in \text{t'}} (\text{f(i)+g(j)=e)}) \end{split}$$

t.First()			
f(t.	$f(t.Current().i)+g(t.Current().j)\neq e$		
	t.Next()		

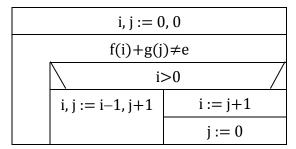
A felsoroló megvalósításához azt kell látni, hogy

- a reprezentálásához szükség lesz az i,j indexekre,
- a felsorolást a 0,0 indexekkel kell kezdeni (First()),
- az indexek felsorolásával a mellékátlók mentén haladjunk tovább (Next()),
- az aktuális indexpárt adja vissza a Current(),
- End() művelet most nem kell, a terminálást a kiválasztás tétele garantálja.

$enor(\mathbb{N} \times \mathbb{N})$

(N×N)*	First()	Next()	Current()
i, j :N	i, j := 0, 0	ha $i > 0$ akkor $i, j := i-1, j+1$	i, j
		ha $i=0$ akkor $i := j+1$; $j := 0$	

Összeolvasva



II. Ismerjük a Föld felszín tengerszint feletti magasságait egy adott szakaszon, amelyeket egy szekvenciális inputfájlban tároltunk el. Hány horpadás van a megadott felszínen?

$$A = (f: infile(\mathbb{R}), db: \mathbb{N})$$

 $Ef = (f = f')$

A szekvenciális fájl klasszikus felsorolása egyszerre csak egy elemét mutatja a fájlnak, nekünk azonban egyszerre három egymást követő elemre kell hivatkoznunk. Olyan felsorolás kell, amely ezt biztosítja.

$enor(\mathbb{R} \times \mathbb{R} \times \mathbb{R})$

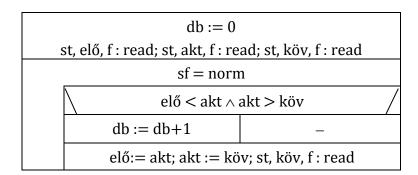
$(\mathbb{R} \times \mathbb{R} \times \mathbb{R})^*$	First()	Next()	End()	Current()
elő, akt, köv : ℝ	st, elő, f : read	elő:= akt	st= abnorm	elő, akt, köv
f: infile(\mathbb{R})	st, akt, f : read	akt := köv		
st : Status	st, köv, f : read	st, köv, f : read		

A feladat specifikációja és absztrakt megoldása az egyedi felsorolóval:

A: (t:enor(
$$\mathbb{R} \times \mathbb{R} \times \mathbb{R}$$
), db: \mathbb{N})
Ef: (t = t' \lambda e=e')
Uf: (db = $\sum_{(e|\delta, akt, ut\delta) \in t'} 1$)
elő>akt^akt

db := 0			
t.First()			
¬t.End()			
t.Current().elő > t.Current().akt \ t.Current().akt < t.Current().köv /			
db := db+1	_		
t.Next()			

Összeolvasva



III. Adottak a szekvenciális inputfájlban a hallgatók évközben szerzett jegyei, a hallgatók neve szerint rendezett formában. Számoljuk ki az egyes hallgatók átlagait!

A: (x:infile(Hallg), y:Outfile(Hallg)) Hallg=rec(azon: String, jegy:
$$\mathbb{R}$$
)
Ef: (x = x₀ \land x \nearrow _{azon})

A feladat megoldásához nem az x szekvenciális inputfájl felsorolását kell közvetlenül felhasználni: olyan egyedi felsoroló kell, amely képes hallgatónként felsorolni az átlagokat, azaz lényegégében előállítja a listázandó adatokat.

A feladat specifikációja és absztrakt megoldása tehát egyszerű: ez egy másolás.

$$\begin{split} A &= (\ t : enor(Hallg), \ y : Outfile(Hallg)\) \\ Ef &= (\ t = t_0) \\ Uf &= (\ y = t_0\) = (\ y = \oplus_{h \in t_0} < h >) \end{split}$$

	y:=<>	
t.First()		
	¬t.End()	
	y:write(t.Current())	
	t.Next()	

enor(Hallgató)

Hallgató*	First()	Next()	End()	Current()
h: Hallg	"első hallgató"	"következő hallgató"	vége	h
vége :L	sx, dx, x : read	lásd lenn		
x: infile(Hallg)	Next()			
dx : Hallg				
sx : Status				

Next()

```
Ef:( sx=sx' \land dx=dx' \land x=x')

Uf:( vége=(sx'=abnorm)

\land (sx'=norm \longrightarrow h.azon = dx'.azon

\land (össz,db), (sx, dx ,x) = \sum_{(sx,dx,x)\in(dx',x')}^{sx=norm \land dx.azon=h.azon}(dx. jegy, 1)

\land h.jegy = össz/db )
```

Megj.: Látjuk, hogy az egyedi felsoroló megvalósítása az x szekvenciális inputfájl szokásos felsorolására támaszkodik. Az x felsorolásának Next() művelete az sx,dx,x:read, a Current() művelete a dx-et adja vissza. Az összegzés esetében (itt számpárokat összegzünk, és a neutrális elem a (0,0) számpár) azonban kicsit módosul a szekvenciális inputfájl standard felsorolása. Egyrészt a már korábban kiolvasott dx' elemet is fel kell dolgozni az x' elemeivel együtt, tehát itt a dx', majd az x' elemeinek felsorolásáról van szó. Erre utal a $(sx, dx, x) \in (dx', x')$ jelölésben szereplő (dx', x'), amely mintha újra egybe forrasztaná egy pillanatra a korábban már kiolvasott dx' elemet az x' sorozattal (fájllal). Ennek az a következménye, hogy nem kell (nem szabad) előre olvasással kezdeni a (dx', x') felsorolását, hiszen a felsorolás első eleme, a dx' már a kezünkben van. A felsorolás First() művelete tehát az üres utasítás lesz. Másrészt a rekordok olvasása nem a fájl végéig tart, hanem csak addig, amíg sx=norm \wedge dx.azon=h.azon. Ennek alapján tehát módosul a felsorolás End() művelete.

t.Ne	xt()	
	vége := sx = abnorm	
	−vége	
	h.azon := dx.azon	
	össz, db := 0, 0	
	$sx=norm \land dx.azon = h.azon$	_
	össz, db := össz+dx.jegy, db+1	
	sx, dx, x : read	
	h.jegy := össz/db	