

Tizedik gyakorlat feladatainak megoldása

1. Adott egy egész számokat felsoroló felsoroló. Hány darab páros számot tartalmaz?

$$A = (t : \text{enor}(\mathbb{Z}), c : \mathbb{N}_0)$$

$$Ef = (t = t')$$

$$Uf = (c = \sum_{e \in t'} \chi(2|e))$$

c := 0	
t.First()	
¬ t.End()	
2 t.Current()	
c := c+1	SKIP
t.Next()	

2. Adott egy egész számokat tartalmazó fájl. Hány páros szám található benne?

$$A = (f : \text{infile}(\mathbb{Z}), c : \mathbb{N}_0)$$

$$Ef = (f = f')$$

$$Uf = (c = \sum_{e \in f'} \chi(2|e))$$

c := 0	
sf, df, f : read	
sf	
2 df	
c := c+1	SKIP
sf, df, f : read	

3. Adott egy fájl, ami 1-eseket és 0-kat tartalmaz. Milyen hosszú a leghosszabb 1-es sorozat?

Pl.: 01100101111001101 esetén 4.

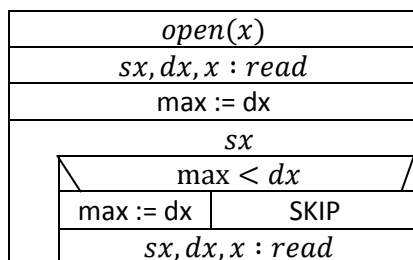
$$A = (f : \text{infile}(\{0, 1\}), \text{max} : \mathbb{N}_0)$$

Ezen a ponton gondoljunk bele, hogy legalább háromszoros visszavezetésre lenne szükségünk, hogy megoldjuk a feladatot. Ehelyett találjuk ki, hogy milyen bemenet esetén lenne ez a feladat sokkal egyszerűbb. Tegyük fel, hogy a fájlunk nem is sorozatokat tartalmaz, hanem az egyes sorozatok hosszait. Ekkor ugyebár az egész feladat egy maximumkiválasztásra redukálódik. Specifikáljuk ezt a feladatot ezzel az új, absztrakt x fájljal.

$$A' = (x : \text{infile}(\mathbb{N}_0), \text{max} : \mathbb{N}_0)$$

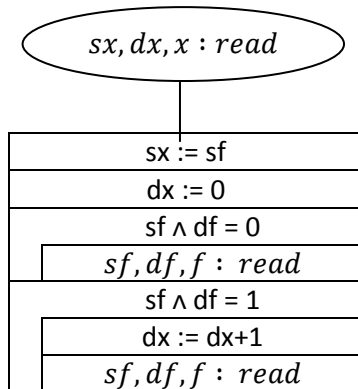
$$Ef = (x = x' \wedge |x| \geq 1)$$

$$Uf = (\text{max} = \mathbf{MAX}(e \in x')(e))$$

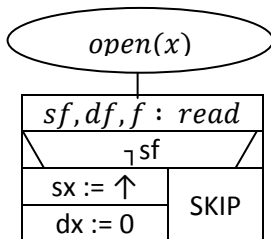


A feladat ezzel még nincs megoldva, mivel ez az x fájl nem létezik, következésképp az $sx, dx, x : read$, valamint az $open(x)$ parancsok sem –ez utóbbiról még beszélek-. Így hát meg kell valósítatnunk ezeket a műveleteket, az $sx, dx, x : read$ parancs mindig a soron következő 1-es sorozat hosszát kell, hogy megadja. Az $open(x)$ parancs feladata az, hogy ilyen absztrakt fájlok esetében az első $read$ parancs működjön, míg a $read$ -nek be kell állítani az sx és a dx változókat az igazi fájl, tehát az f alapján.

Mondjuk azt, hogy az x fájlból való $read$ eljárásához érve a fizikai fájlban mindig egy 0-s sorozat elején állunk –ezt invariáns tulajdonságnak is nevezzük-. Ekkor először egy lineáris kereséssel átlépkedünk a 0-kon. A ciklus végén vagy vége van a fájlnak, vagy 1-es sorozat kezdődik. Az utóbbi esetén átlépkedünk az 1-eseken, és folyamatosan növeljük a dx -et, ami így az 1-esek számát fogja megadni. Ha átléptünk az 1-eseken, akkor vagy vége van a fájlnak, vagy megint egy 0-s sorozat elején állunk, azaz teljesül az előbb leírt invariáns tulajdonság. Az sx -et nem a $read$ végén állítjuk be az sf -re, hiszen úgy elveszne a legutoljára mért 1-es sorozat. Ehelyett a $read$ eljárás elején végezzük el –ily módon, amikor egy egyes sorozat mérésekor elérünk a fizikai fájl végére, először visszaadjuk a dx -et, és csak a következő $read$ -nél szólunk, hogy vége van a kitalált fájlunknak.



Az *open(x)* feladata, hogy amennyiben a valódi fájl üres, mi azért szolgáltatassunk egy 0 értéket, azaz hogy a leghosszabb 1-es sorozat 0 hosszú, illetve azért is, mert az új fájlt tartalmazó specifikációban kikötöttük, hogy a fájl hossza legalább 1. Arra is oda kell figyelni, hogy az *open* után következő *read* megfelelően működjön.



Végül vegyük észre, hogy ez a megoldás visszavezetés egy felsorolás tételre:

x.First()	<i>open(x)</i>
	<i>sx, dx, x : read</i>
x.End()	¬sx
x.Current()	dx
x.Next()	<i>sx, dx, x : read</i>