|  |  |  |
| --- | --- | --- |
| **Lőrincz Ákos** ky90bp  akoslorincz00@gmail.com  12.csoport | **2. beadandó/5.feladat** | 2021. 04. 02. |

# Feladat

Az Országos Horgász Bajnokság mindegyik versenyén feljegyezték a versenyzők eredményeit, és egy szöveges állományban rögzítették az adatokat. Ennek minden sora tartalmazza a versenyző horgász és a verseny azonosítóját (szóközök nélküli sztring), majd halfajta-méret párok (szóközök nélküli sztring és természetes szám párok) formájában az adott versenyen elért fogásokat. Az adatok szóközökkel vagy tabulátorjelekkel vannak egy soron belül elválasztva. A szöveges állomány sorait a versenyek szerint rendezték. Feltehetjük, hogy a szöveges állomány helyesen van kitöltve.

Példa az állomány egy sorára:

JANIBÁ Kiliti0512 Ponty 45 Ponty 53 Harcsa 96

(1) Volt-e olyan, horgász, aki valamelyik versenyen csak harcsát fogott, de abból legalább hármat? Ha igen, adjuk meg az első ilyen horgászt, és azt is, melyik versenyen volt!

(2) Melyik versenyen indult a legtöbb horgász?

***(1) Részfeladat megoldása:***

***Főprogram terve:***

*A = (t : enor(HarcsaHorgasz), l :* 𝕃*, horgasz\_nev:String)*

*HarcsaHorgasz =* ***rec****(nev:String, haromHarcsa:* 𝕃*)*

*Ef =( t = t’)*

*Uf =(* 𝑙, 𝑒lem = 𝑺𝑬𝑨𝑹𝑪𝑯′(elem.haromHarcsa) and *horgasz\_nev* = elem.name*)*

𝑒∈𝑡

|  |  |
| --- | --- |
| *l:=hamis; t.first()* | |
| !*l* ∧ !*t.end()* | |
|  | *l horgasz\_nev := t.current().haromHarcsa,*  *t.current().name* |
| *t.next()* |

# Harcsahorgászok felsorolója[[1]](#footnote-1)

|  |  |
| --- | --- |
| enor(HarcsaHorgász) | first(), next(), current(),  end() |
| x: infile(Horgász)  sx: Status | first() { read(); next() } |
| elem: HarcsaHorgász dx: Horgász | next() ~ ld. külön |
| end\_: L | current() ~ elem  end() ~ end\_ |

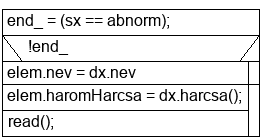
Felsoroló programterve:

*Anext = (f: infile(Horgász), end\_:*𝕃*, elem:HarcsaHorgász) Horgász = rec(nev:string, versenyid:string, halak:Haladat[], bool harcsa();*

*Efnext =* ( *f = f’ )*

*Ufnext =* ( *sx, dx, x = read(f’)* ∧ *end=(sx=abnorm)*

∧ ¬ *end* → *elem.nev* = *dx.nev* ∧ elem.haromHarcsa= dx.harcsa()



harcsa függvény specifikációja:

Aharcsa=(t: enor(Horgász), h: Horgász, harcsa: 𝕃, count: ℕ, csakharcsa: 𝕃)

*Efharcsat =* ( *e = t0)*

Ufharcsa=(e.csakharcsa ,

e.count

harcsa = e.csakharcsa ∧ e.count > 2)

***(2) Részfeladat megoldása***

# Főprogram terve

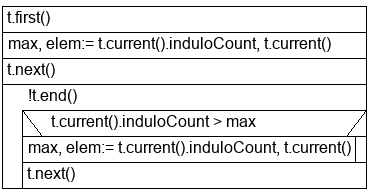
*A = (g : enor(Verseny), id : String)*

*Verseny =**rec(id:String, induloCount*: ℕ*)*

*Ef =( g = g’, |g| > 0)*

*Uf =(* max, 𝑒lem =MAX elem.induloCount*)*

𝑒∈g



# Versenyek felsorolója[[2]](#footnote-2)

|  |  |
| --- | --- |
| *enor(Verseny)* | *first(), next(), current(), end()* |
| x : filestream sx : Status  dx: Horgász | *first() read, next()* |
| *elem*: Verseny | *next() ~ ld. külön* |
| end\_ : L | *current()* ~ *elem*  *end()* ~ *end*\_ |

*Anext = (g:enor(Horgász), elem:Verseny end\_:*𝕃*)*

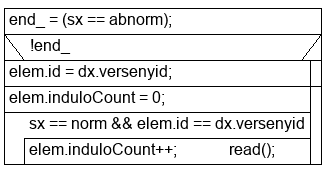
*Efnext =* ( *g =gt’*)

*Ufnext =* ( *sx, dx, x = read(gt’)* ∧ *end=(sx=abnorm)*

∧ ¬ *end* → = ∑ 1)

𝑒∈g𝑡′

*elem.id* = *dx.versenyid*



# Tesztelési terv

A megoldásban kettő programozási tételt alkalmaztunk: számlálás, maximum kiválasztás.

1. Egy olyan horgász keresése, aki csak harcsát fogott, de minimum 3-mat tesztesetei:
   1. Nincs megfelelő horgász.
   2. Első horgász ilyen.
   3. Utolsó horgász ilyen.
2. Legtöbb résztvevőjű verseny keresése tesztesetei:
   1. Első verseny ilyen.
   2. Utolsó verseny ilyen.
   3. Két versenynek is ugyanannyi résztvevője van.

1. A felsorolót külön fordítási egységben helyezzük el. [↑](#footnote-ref-1)
2. A versenyek felsorolója is külön fordítási egységbe kerül. [↑](#footnote-ref-2)