Objektum elvű programozás 2. beadandó feladat/ 5. feladat

2020.04.12.

Madár Bálint

MRPBLG

balint.mb@gmail.com

18. csoport

5. 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?

# Részfeladat megoldása:

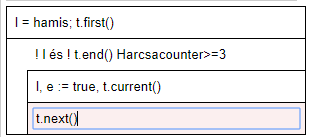
**Főprogram terve: (Lineáris keresés)**

A = (t : enor(Verseny), l : 𝕃, e: Verseny)

Verseny = rec(horgász:String, verseny:String, halfajta:String, db:ℕ)

Ef =( t = t’)

Uf =( 𝑙,𝑒 = 𝑺𝑬𝑨𝑹𝑪𝑯 𝑒∈𝑡 ′ (𝑒. 𝑑𝑏 ≥ 3 és csak harcsa))



Versenyek felsorolója

|  |  |
| --- | --- |
| enor(Verseny) | first(), next(), current(), end() |
| \_x : infile(Sor)  cur : FisherMan  end : L | first() ~ next()  next() ~ ld. külön  current() ~ cur  end() ~ end |

Az enor(Verseny) first() és next() műveletei megegyeznek, és az alábbi feladatot kell megoldaniuk: olvassuk be a szöveges állomány (az \_x szekvenciális inputfájl) soron következő sorát.

Ha ilyen nincs, akkor az end változó értéke legyen igaz.

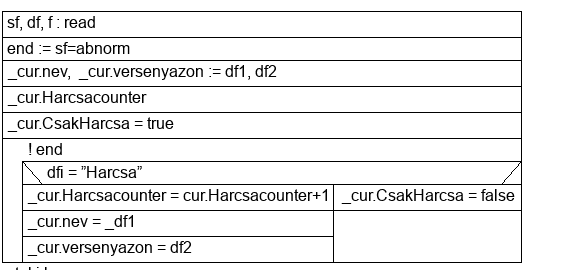
Ha van, akkor ezt a sort sztringek sorozatának tekintve kiszedhetjük belőle a horgász és a verseny azonosítóját, majd az ezeket követő részben megszámolhatjuk a „Harcsa” szavakat akkor ha az adott sorban csak harcsa található.

A next = (f: infile(Sor), end:𝕃, cur:Verseny) **Sor = String\***

Ef next = (x = x’)

Ufnext = ( sf, df, f = read(\_x’) ∧ end=(sf=abnorm) ∧ ¬end → cur.horgász = df1 ∧ cur.verseny = df2 ∧ cur.db = ∑𝒊∈[𝟑..∣𝒅𝒇∣] é𝒔 𝒊 𝒑𝒕𝒍 𝟏 𝒅𝒇[𝒊]="𝒉𝒂𝒓𝒄𝒔𝒂" és **l = true (l = csakharcsa)**)

**Megvalósítás:**



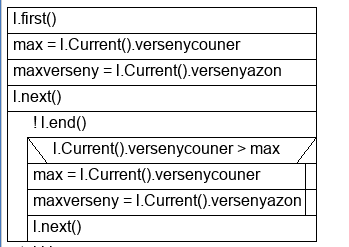
# Részfeladat megoldása

A = ( t:enor(Verseny), max:H, elem: Verseny )

Verseny = rec(horgász:String, verseny:String, halfajta:String, db:ℕ)

Ef = ( t=t’ ∧ |t|>0 )

Uf = ( max, elem =𝑀𝐴𝑋𝑒∈t'Verseny(𝑒))



Versenyek felsorolója2

|  |  |
| --- | --- |
| enor(Verseny) | first(), next(), current(), end() |
| \_x : infile(Sor)  cur : FisherMan  end : L | first() ~ next()  next() ~ ld. külön  current() ~ cur  end() ~ end |

Az enor(Verseny) next() műveletnek az alábbi feladatot kell megoldania:

Számoljuk meg a versenyek előfordulásának számát és válasszuk ki a legtöbbet szereplő versenyt, ennek adjuk meg a nevét.

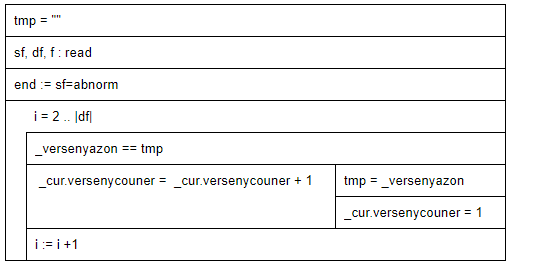
Ebben a részben a számlásás lesz

A next = (f: infile(Sor), end:𝕃, cur:Verseny) **Sor = String\***

Ef next = (x = x’)

Ufnext = ( sf, df, f = read(\_x’) ∧ end=(sf=abnorm) \_cur.versenycouner = ∑𝑒∈t' 1)

felt(df != df.next())



# Tesztelési terv

A megoldásban három programozási tételt alkalmaztunk: optimista lineáris keresés, számlálás, maximum kiválasztás.

1. **A. Első feladat lineáris keresésének tesztesetei:**

– Egy olyan horgász keresése, aki legalább két harcsát fogott.

– intervallum hossza szerint:

1. Üres állomány.

2. Egyetlen horgász.

3. Több horgász. intervallum eleje és vége szerint:

4. Első horgász fogott két harcsát.

5. Csak az utolsó horgász fogott három harcsát. tétel szerint:

1. Van megfelelő horgász.

2. Nincs megfelelő horgász.

3. Több megfelelő horgász is van. Ebben az esetben az elsőt adja vissza

B. Egy horgász egy versenyen fogott harcsáinak száma (számlálás) tesztesetei: intervallum hossza szerint:

1. Olyan sor, amely nem tartalmaz fogásokat.

2. Olyan sor, amely egy fogást tartalmaz.

3. Olyan sor, amely több fogást is tartalmaz. intervallum eleje és vége szerint:

4. Olyan sor, amely legelső fogása harcsa.

5. Olyan sor, amely utolsó fogása harcsa. tétel szerint:

7. Olyan sor, amely nem tartalmaz harcsa fogásokat.

8. Olyan sor, amely egy harcsa fogást tartalmaz.

9. Olyan sor, amely több harcsa fogást tartalmaz.

C Maximum Kiválasztás megkezdett felsorolóval:

• felsorolás hossza: nulla, egy,

• felsorolás eleje: az elején rögtön a maximum

• felsorolás vége: csak a végén van maximum

• A kiválasztás feltétele: nincs maximum (ekkor az első elemet választja ki)