|  |
| --- |
| **Baranyai Dominik** 2**.beadandó/2.feladat**  2020.04.12  JH0M5P  [bdominik14@indamail.hu](mailto:bdominik14@indamail.hu)  11.csoport |

**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 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) Adjuk meg azt a horgászt, akinek a legtöbb halat sikerült kifognia valamely versenyen! Adjuk meg azt is, hogy melyik az a verseny, és hány halat fogott!

(2) Hány olyan verseny volt, ahol nem fogtak halat?

**(1) Részfeladat megoldása:**

Főprogram terve:

A = (t : enor(Line), l : 𝕃, e: Line) Line = rec(angler:String, contest:String, fish:ℕ)

Ef =( t = t’)

Uf =( e.fish, 𝑒.angler = **MAX** 𝑒∈𝑡 ′ (𝑒.fish))

|  |  |  |
| --- | --- | --- |
| t.first(); max := t.current() | | |
| !t.end() | | |
|  | t.current().fish > max.fish | |
| max := t.current() | SKIP |
| t.next() | |

Felsoroló:

t:enor(Food) Food = rec(name: 𝕊, ingredients: ℕ)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Line\* | first() | next() | current() : Item | end() :𝕃 |
| inp : infile(Line)  sx : Status  cur : Line  end : 𝕃 | sx,dx,inp:read  next() | lásd külön | **return** cur | **return** \_end |

next() művelet

A = (inp:infile(Line), sx:Status, cur : Line, \_end:𝕃)

Ef = ( inp = inp’ ∧ inp↗ ∧ dx = dx’ ∧ sx = sx’ )

Uf = ( end = (sx’=abnorm) ∧ ( ¬end → (cur.fish, (sx,dx,inp))=∑ dx∊(dx’,inp’) 1) )

|  |  |
| --- | --- |
| sx, dx, inp : read | |
| end := sx=abnorm | |
| cur.angler, cur.contest := dx1, dx2 | |
| cur.fish := 0 | |
| i = 3 .. ∣df ∣ | |
|  | cur.fish := cur.fish + 1 |
| i := i+1 |

**(2) Részfeladat megoldása:**

Főprogram terve:

A = (t : enor(Line), l : 𝕃, e: Line) Line = rec(angler:String, contest:String, fish:ℕ)

Ef =( t = t’)

Uf =( null\_fish = **∑** 𝑒∈𝑡 ′ e.fish == 0 (𝑒.fish))

|  |  |  |
| --- | --- | --- |
| t.first(); null\_fish = 0 | | |
| !t.end() | | |
|  | t.current().fish == 0 | |
| null\_fish := null\_fish + 1 | SKIP |
| t.next() | |

Felsoroló megegyezik az első feladatban használtal

**Tesztelési terv:**

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

1. Első feladat maximum kiválasztás tesztesetei:

*– Egy olyan horgász keresése, aki a legtöbb halat fogta.* ***–*****intervallum hossza** szerint:

* + 1. Üres állomány. (t0.txt)
    2. Egyetlen horgász. (t1.txt)
    3. Több horgász. (t3.txt)

**intervallum eleje és vége** szerint:

1. Első horgász fogta a legnagyobb harcsát. (t2.txt)
2. Az utolsó horgász fogta a legnagyobb harcsát. (t3.txt)

B. Második feladat feltételes számlálás tesztesetei:

*– Versenyek, ahol nem fogtak halat, számlálása.*

*–* **intervallum hossza** szerint:

* + 1. Üres állomány. (t0.txt)
    2. Egyetlen verseny. (t5.txt)
    3. Több verseny. (t6.txt)