|  |
| --- |
| **Beregi Bence Zsolt 2. beadandó/10. feladat** 2020. április 6.  DQK6TE  [dqk6te@inf.elte.hu](mailto:dqk6te@inf.elte.hu)  15. 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-súly párok (szóközök nélküli sztring és valósszám párok) formájában az adott versenyen elért fogásokat. Az adatok szóközökkel vagy tabulátorjelekkel vannak egy soron belül elválasztva. A szöveges állomány sorait a horgászok szerint rendezték. Feltehetjük, hogy a szöveges állomány helyesen van kitöltve. Példa az állomány egy sorára:  
JANIBÁ Kiliti0512 Ponty 4.5 Ponty 5.3 Harcsa 9.6

(1) Mennyi volt a legkisebb kifogott ponty súlya? Ki és melyik versenyen fogta? (Nem biztos, hogy fogtak pontyot a horgászok!)

(2) Igaz-e, hogy minden horgász legalább két versenyen fogott pontyot?

# **(1) részfeladat megoldása**

## Főprogram terve:

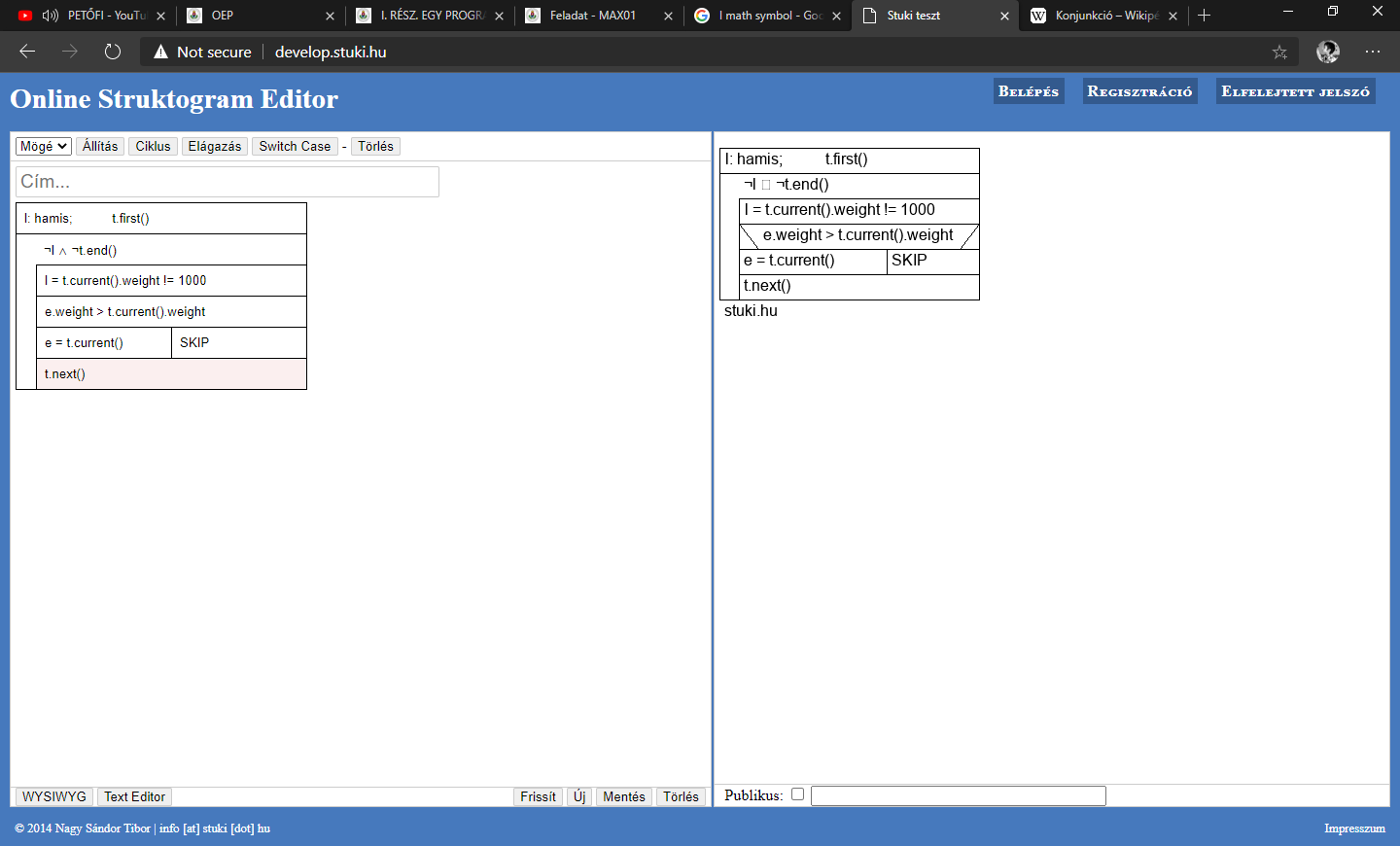
A = ( t : enor(Contest), l : 𝕃, e: Contest )

Contest = rec( angler: String, contest: String, wasCarp: 𝕃,  
weight: double )

Ef = ( t = t’ )

Uf = ( l, e = min( e.weight ))

e.wasCarp == true



ᴧ

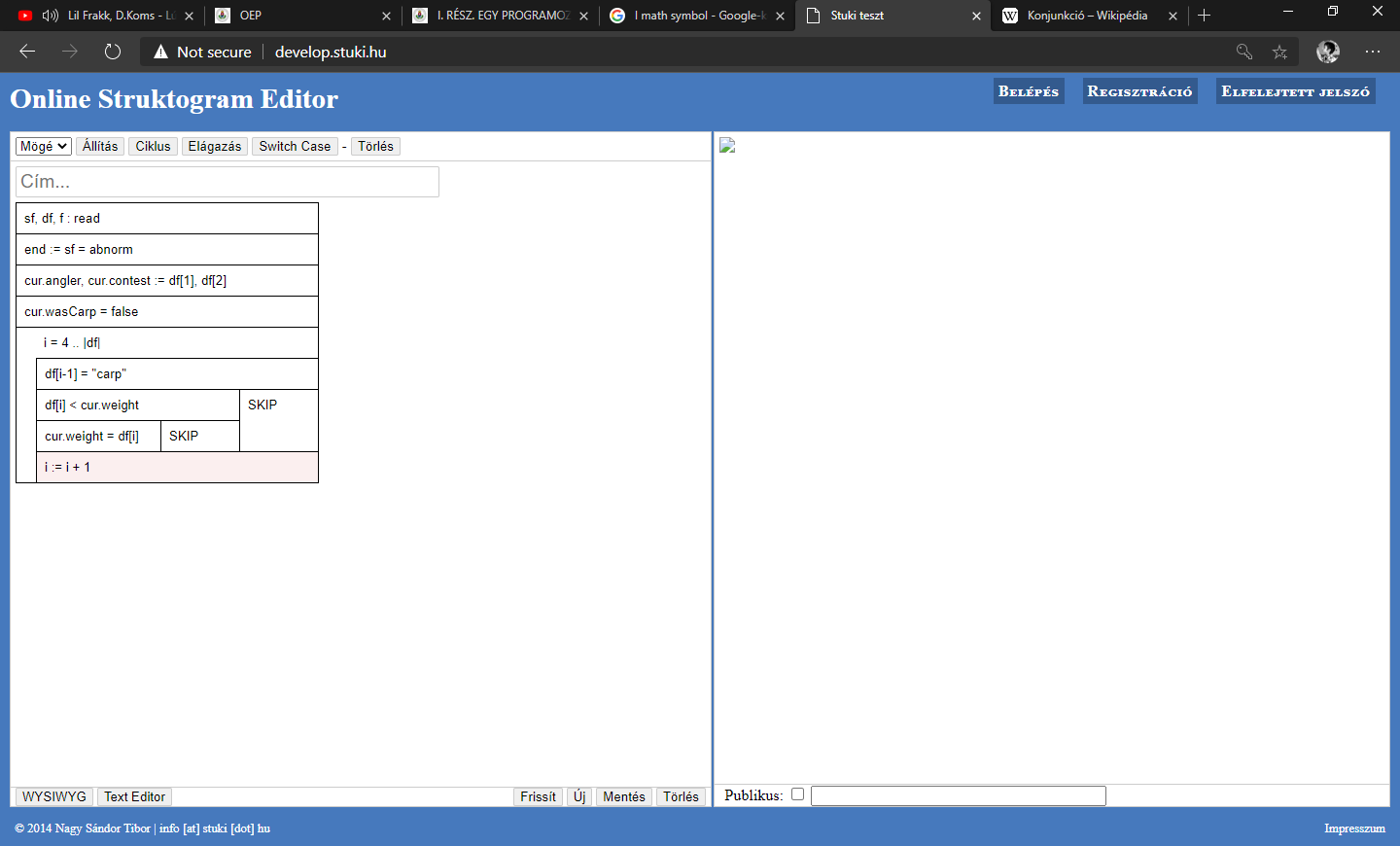
## Versenyek felsorolója

|  |  |
| --- | --- |
| enor(Contest) | first(), next(), current(), end() |
| f: infile(line)  cur: Contest 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 f 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 ezután, ha van ponty a sorban a wasCarp változót „true”-ra állítjuk (ez a (2) részfeladathoz szükséges )és megnézzük, hogy a ponty(ok) súlya kisebb-e a jelenlegi minimumnál.

Anext = ( f: infile(line), end: 𝕃, cur: Contest ) line = String

Efnext = ( f = f’ )

Ufnext = ( sf, df, f = read(f’) ᴧ end = (sf = abnorm) ᴧ ¬end 🡪 cur.angler = df1 ᴧ cur.contest = df2 ᴧ cur.weight = miniϵ{[4, |df|]} ᴧ i páros (df[i]))

df[i-1] == “carp”

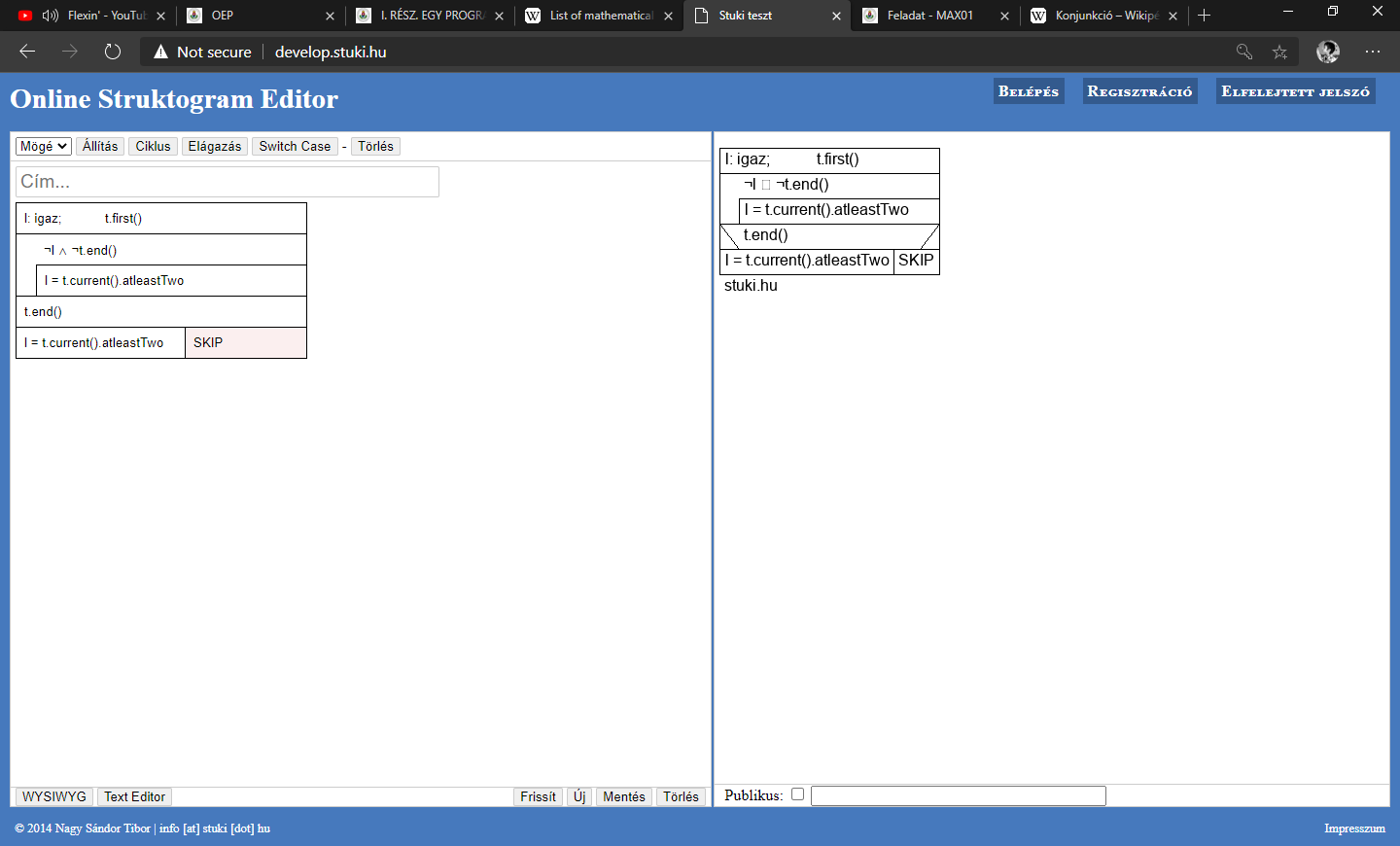
# **(2) részfeladat megoldása**

## Főprogram terve:

A = ( t : enor(Contest), l : 𝕃 )

Angler = rec( id: String, atleastTwo: 𝕃)

Ef = ( t = t’ )

Uf = ( l = l. Igaz, ha minden sorban igaz a Contest structaban a wasCarp  
ll. Hamis, ha akár egy sorban nem igaz a Contest structban a wasCarp)

ᴧ

## Horgászok felsorolója

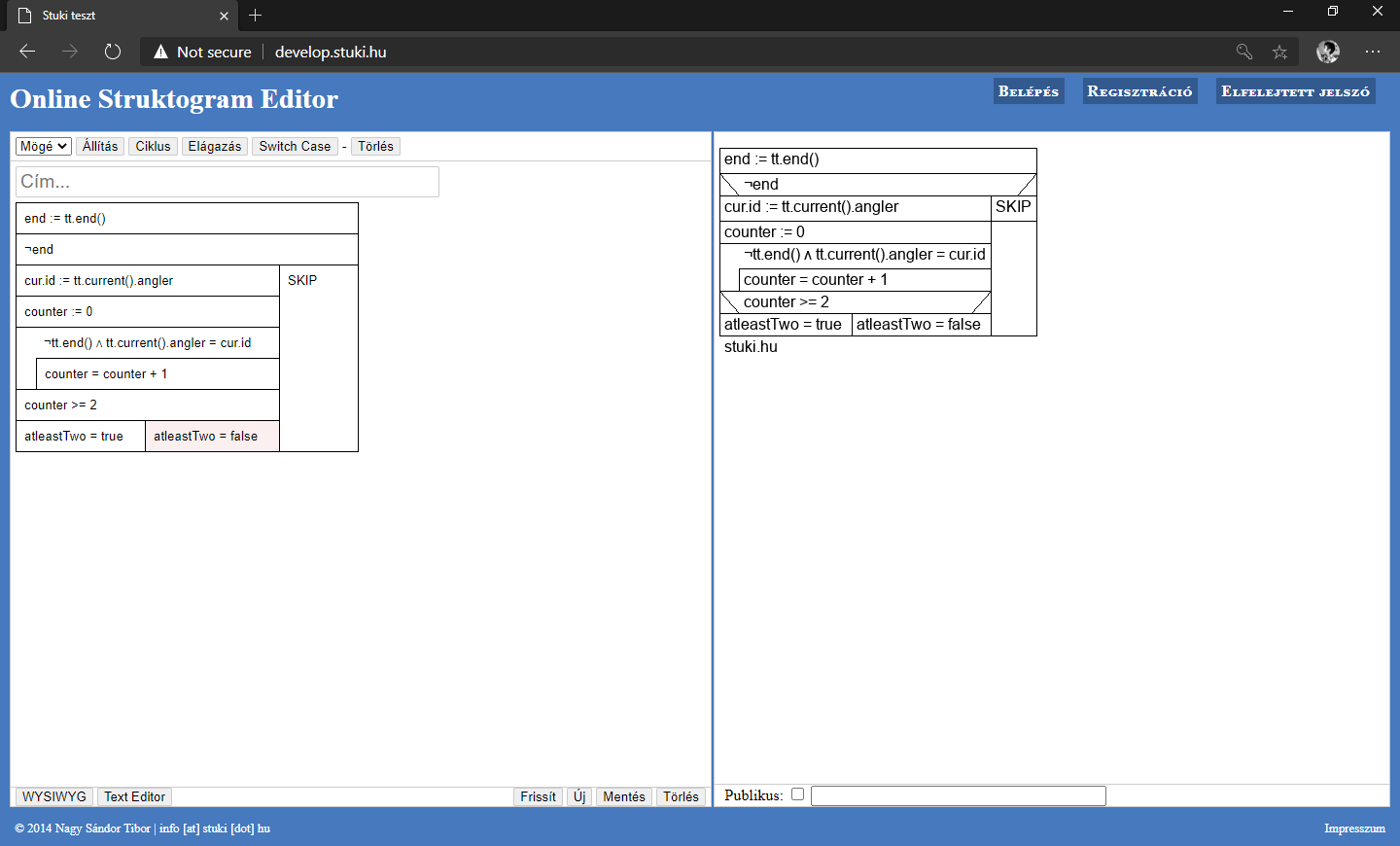
|  |  |
| --- | --- |
| enor(Angler) | first(), next(), current(), end() |
| tt: enor(Contest)  cur: Angler end: 𝕃 | first() ~next()  next() ~ld. külön  current() ~cur  end() ~end |

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

Állítsuk elő a soron következő horgászt, akiről el kell dönteni, hogy vajon legalább két versenyen fogott-e pontyot. Ehhez fel tudjuk használni az első részfeladat megoldásához készített verseny felsorolót (enor(Contest)), mely az input fájl egy sorát értelmezi, és megadja, hogy az adott versenyen az adott horgász fogott-e pontyot. A next() művelet végrehajtásakor a Contest felsoroló (ha még nem ért a felsorolás végére) már tartalmazza azt a horgászt, és annak első versenyét, amelyet a next() feldolgoz majd, így, a tt.current() értékét már ismerjük, ehhez nem kell sem a tt.first(), sem a tt.next(). A felsorolás csak addig tart, amíg ugyanazon horgász versenyeit „olvassuk” a tt.next() művelettel.

Anext = ( tt : enor(Contest), end : 𝕃, cur : Angler)

Efnext = ( tt = tt’ )

Ufnext = ( end = tt’.end() ᴧ ¬end 🡪 (cur.id = tt’.current().angler ᴧ  
cur.atleastTwo = (2 <= )))

# **Tesztelési terv**

A teszteseteket úgy állítottam össze, hogy lehető legtöbb módon tesztelejem a programot. Van benne, hogy a helyét változtatja a keresendő értek, a fájlban lévő adatok hiányosságával tesztelem. Külön tesztelem a felsorolót és a főprogramot, hogy a lehető legtöbb módon próbáljam a hibát kiszűrni. A tesztelés folyamán sor került a felmerült hibák kijavítására is, így sikerült kiszűrni a hibákat.