Geszti Bence
WXNDSZ

3. FELADAT:

Egy horgászversenyen valahányszor egy versenyző halat fog, feljegyzik egy szöveges állomány soron következő sorába a versenyző azonosítóját (négyjegyű szám), a hal fajtájának nevét (ponty, keszeg, süllő, stb.) és a hal méretét. Feltehetjük, hogy a szöveges állomány helyesen van kitöltve: minden sorában három adat van (azonosító, halfajta, méret) szóközökkel vagy tabulátorjelekkel elválasztva. A verseny végén a szöveges állomány sorait azonosító szerint sorba rendezik. Adjuk meg annak az azonosítóját, aki a legtöbb 30 cm-nél hosszabb pontyot fogta??

SPECIFIKÁCIÓ

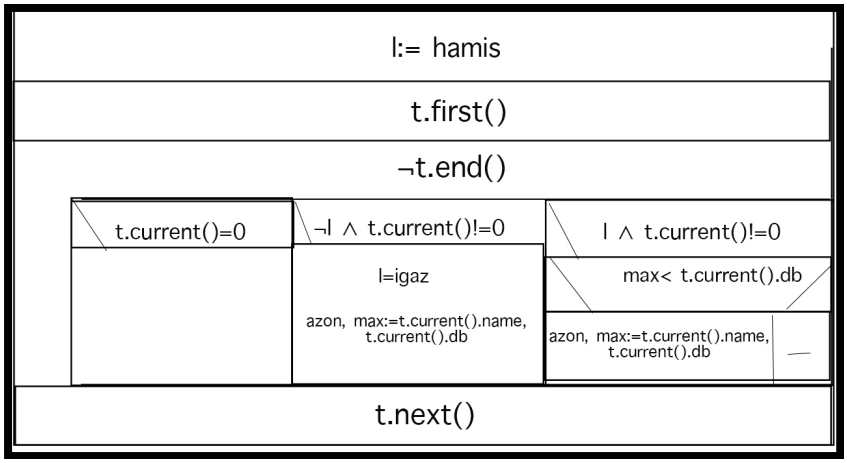
A = (t : Enor(peca), azon:String, max:Z, l:L)
peca = rec(name,faj:String, db:N)

Ef = (t=t' $\wedge$ peca.name szerint rendezett)

Uf = ((max,azon,l)= MAX akt.db)

$$akt \in t'$$

$$akt.db!=0$$



Max ker

t ~ x sorainak felsorolója

f(e) ~ akt.db

b(e) ~ akt.db != 0

enor(peca)	first(), next(), current(), end()
f: InFiles(String,) df, st:Státusz	first() ~ f>>df; next()
cur: peca **df:peca2**	next() ~ külön
vege= L	current() ~cur
	end() ~ verge

NEXT()

A = (f:infile(peca), df:peca2, fin:L, cur:peca , sf:Status)

peca2 = rec(name,faj:String, hossz:N)

Ef = (f=f' ∧ df=df' ∧ sf=sf')

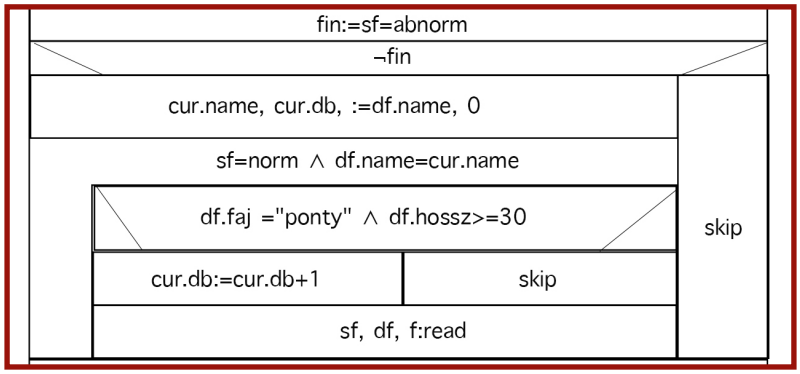
Uf= (fin=(sf'=abnorm) ∧ (¬fin → ∧ cur.name:=df.name ∧

df.name=cur.name ∧ sf=norm

cur.db,(sf, df, f)= ∑ 1))

df ∈ (df',f')

df.faj ="ponty" ∧ df.hossz>=30



IMPLEMENTÁCIÓ

Program váz

A program több állományból áll: main.cpp, enor.h, enor.cpp.

main.cpp	enor.cpp	enor.h
int main()	**void next()**	struct peca, peca2
	enor()	void first()
		class enor()
		peca current()
		bool end()

felsoroló osztálya:

```
class Enor
{
    public:
        struct peca{
            std::string name,faj;
            int db;
        };
         struct peca2{
            std::string name,faj;
            int hossz;
        };

        Enor(const std::string &str);
        void first() { f >> df.name >> df.faj>>df.hossz; next();}
        void next();
        bool end() const { return fin;}
        peca current() const { return cur;}
    private:
        std::ifstream f;
        peca2 df;
        peca cur;
        bool fin;

};
```

Tesztelés

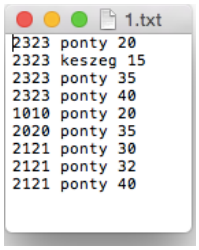
Megszámlálás

1. üres **5.txt** [ ] **c=üres**

   csak szóközök **6.txt** [ ] **c=üres**

2. egyetlen keresett tulajdonsagu van “2323 ponty 35” **7.txt c=1**

3. több keresett tulajdonsagu van: **1.txt c={2, 0, 1, 3}**



Maximum kiválasztás

1. üres **5.txt** [ ] **max=üres**

   csak szóközök **6.txt** [ ] **max=üres**

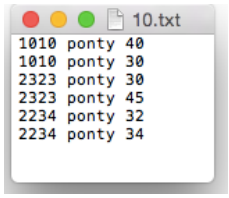
2. egyetlen keresett tulajdonsagu van “2323 ponty 35” **7.txt max=1**

3. több keresett tulajdonsagu van: **1.txt max=3**

4. Első a maximum “2323 ponty 40 1212 ponty 10” **4.txt max=1**

5. Utolsó a maximum “2323 keszeg 10 1010 ponty 30 “ **3.txt max=1**

6. tobb ugyan olyan értékú keresett tulajdonságu **10.txt max=2**



A megoldó programra épülő (fehér doboz) tesztesetek:

1.Hibás vagy nem létező állománynév megadása.

2.Minden érték külön sorban **2.txt**

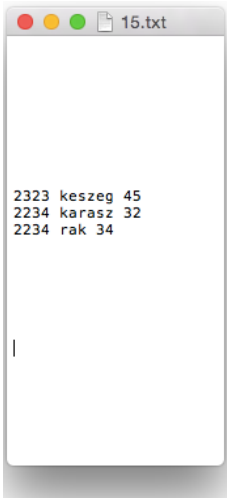
3.Minden érték elsősorban van szóközzel elválsztva **12.txt**

4.Főprogram ciklusának ellenőrzése: olyan bemenő adatokkal, amelyekre a ciklus egyszer sem fut le (Pl: **5.txt**), pontosan egyszer fut le (Pl: **8.txt**), többször lefut (Pl: **1.txt**).

5.üres sorok után tulajdonságoknak megfelelő van benne **11.txt**

6. üres sorok után tulajdonságoknak nemmegfelelő van benne **13.txt**

7. üres sorok után tulajdonságoknak megfelelő van benne, majd megint üres sorok: **14.txt**



8. üres sorok után tulajdonságoknak nemmegfelelő van benne, majd megint üres sorok: **15.txt**

9. üres sorok után tulajdonságoknak megfelelő adatok vannak benne közöttük üres sorok **16.txt**