

Gregorics Tibor
EHACODE.ELTE
gt@inf.elte.hu
0.csoport

0. beadandó/0.feladat

2011. december 28.

Feladat

Igaz-e, hogy a megadott keresztnemek között csupa „Ibolya” név található?

Specifikáció

A neveket egy tömbben tároljuk.

$$A = (t : \text{String}^n, l : \mathbb{L})$$

$$Ef = (t = t')$$

$$Uf = (Ef \wedge l = \forall i \in [1..n]: t[i] = \text{„Ibolya”}) = (Ef \wedge l = \forall_{i=1}^n \text{search } t[i] = \text{„Ibolya”})$$

Algoritmus

A feladatot a lineáris keresés programozási tételének „optimista” változatára vezetjük vissza. Mivel a tétel kimenetei között szereplő index a feladat szempontjából nem érdekes, ezért az $ind := i$ értékadás elhagyható az algoritmusból.

$$m..n \sim 1..n$$

$$\beta(i) \sim t[i] = \text{„Ibolya”}$$

$l, i := igaz, 1$
$l \wedge i \leq n$
$l := t[i] = \text{„Ibolya”}$
$i := i + 1$

Implementáció

Adattípusok megvalósítása

A kódoláskor a t tömböt `vector<string>`-ként deklaráljuk, amelynek mérete `t.size()` alakban érhető el. Mivel a vektor 0-tól indexelődik, azért a tervbeli ciklus nem az $1..n$, hanem a $0..n-1$ intervallumot, pontosabban a $0..t.size()-1$ intervallumot futja be, aminek következtében a struktogramm kódja az alábbi lesz:

```
bool l = true;
for(int i=0; l & i<(int)t.size(); ++i){
    l = t[i]=="Ibolya";
}
```

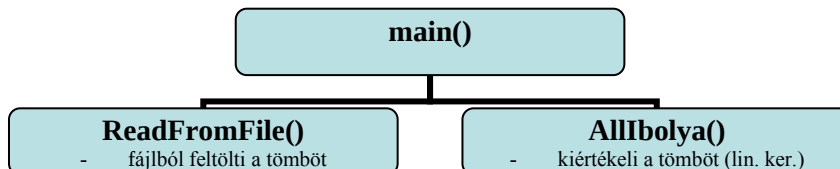
Bemenő adatok formája

A bemenő adatokat egy szöveges állományból kell a tömbbe bemásolni. Az állományban a megadott neveket szóközzel, tabulátor jelekkel vagy sorvége jelekkel elválasztva kell beírni. Az állomány minden sorát sorvége jel zárja le. Például:

Ibolya
Rózsa
Ibolya Ibolya
Kata Viola

Függvények kapcsolódási szerkezete

A kódban több függvényt is használunk. Az `AllIbolya()` tartalmazza a tervben leírt keresést, a `ReadFromFile()` tölti fel egy szöveges állományból a tömböt nevekkal, ezeket a függvényeket pedig a `main()` hívja, amelyik az eredmény kiírását is végzi.



Tesztelési terv

A feladat specifikációjára épülő (fekete doboz) tesztesetek:

Érvényes tesztesetek: (érvénytelen tesztesetek nincsenek)

A. Intervallum tesztesetei:

1. Egyetlen név sincs (üres vagy csak elválasztójeleket tartalmazó fájl esete)
(t1.txt: [] – válasz: hamis)
(t2.txt: [] – válasz: hamis)
2. Egy név szerepel csak
(t3.txt: [Ibolya] – válasz: igaz)
(t4.txt: [Rózsa] – válasz: hamis)
3. Több nem Ibolya név, és a végén egyetlen Ibolya.
(t5.txt: [Rózsa, Viola, Ibolya] – válasz: hamis)
4. Több Ibolya név, és a végén egy nem Ibolya.
(t6.txt: [Ibolya, Ibolya, Viola] – válasz: hamis)
5. Több nem Ibolya név, és az elején egyetlen Ibolya.
(t7.txt: [Ibolya, Rózsa, Viola] – válasz: hamis)
6. Több Ibolya név, és az elején egy nem Ibolya.
(t8.txt: [Viola, Ibolya, Ibolya,] – válasz: hamis)

B. Lineáris keresés tesztesetei:

1. Egyetlen Ibolya név szerepel csak
(t3.txt: [Ibolya] – válasz: igaz)
2. Egyetlen nem Ibolya név szerepel csak
(t4.txt: [Rózsa] – válasz: hamis)
3. Több Ibolya név van, semmi más.
(t9.txt: [Ibolya, Ibolya, Ibolya] – válasz: igaz)
4. Csak nem Ibolya név van.
(t10.txt: [Rózsa, Viola, Kata] – válasz: hamis)
5. Vannak nem Ibolyák is.
(t11.txt: [Rózsa, Ibolya, Viola, Ibolya] – válasz: hamis)
6. Több Ibolya név, és a közepén egy nem Ibolya. (t9.txt)
(t12.txt: [Ibolya, Ibolya, Rózsa, Ibolya] – válasz: hamis)

C. Különleges értékek esetei:

1. Üres sztring, mint név (A beolvasás nem veszi figyelembe)
2. Érdekes nevek: csak egy Ibola, vagy Iboly, vagy I.
3. Nemcsak betűket tartalmazó nevek esete (Ez egy nem Ibolya név)

A megoldó programra épülő (fehér doboz) tesztesetek:

1. Hibás vagy nem létező állománynév megadása.
2. Állomány nevének megadása parancssorból.
3. Ismételt futtatás kipróbálása
4. Olyan állomány olvasása, ahol egy sorban több név is található egyetlen illetve több szóközzel és/vagy tabulátor jellel elválasztva (t12.txt).
5. Olyan állomány olvasása, ahol minden név külön sorban van (t10.txt).
6. Olyan állomány olvasása, ahol az utolsó sort nem zárja sorvége jel, és éppen ennek a sornak a tartalma határozza meg az eredményt (adat: [Ibolya, Ibolya, Ibolya, Ibolya, Viola] – válasz: hamis) (t13.txt).
7. Főprogram ciklusának ellenőrzése: olyan bemenő adatokkal, amelyekre a ciklus egyszer sem fut le (Pl: t1.txt), pontosan egyszer fut le (Pl: t3.txt), többször lefut, igaz logikai értékkel lép ki (Pl: t9.txt), vagy hamis logikai értékkel lép ki (Pl: t6.txt).