Gregorics Tibor **1. beadandó/0.feladat** 2008. december 6.
EHACODE.ELTE
gt@inf.elte.hu
0.csoport

Feladat

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

Specifikáció

A neveket egy tömbben tárolom és a feladatot a lineáris keresés programozási tételének „optimista” változatára vezetem vissza:

$A = (t : String^n, l: \mathbb{L})$
$Ef = (t = t')$
$Uf = (Ef \wedge l = \forall i \in [1..n]: t[i] = \text{„Ibolya”}) = (Ef \wedge l = \overset{n}{\underset{i=1}{\forall \textbf{\textit{search}}}} t[i] = \text{"}Ibolya\text{"})$

Absztrakt program

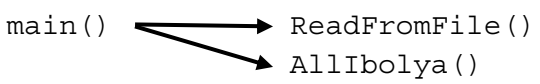
l,i:=igaz,1
l ∧ i ≤ n
l := t[i]=”Ibolya”
ind:=i
i := i+1

Mivel az index itt nem érdekes, ezért az *ind:=i* értékadás elhagyható.

Implementáció

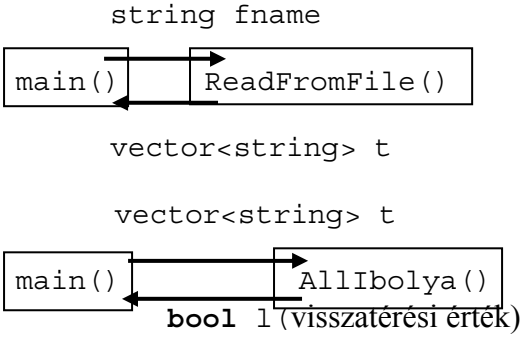
A kódban több függvényt is használok.

Függvények kapcsolódási szerkezete



- fájlból feltölti a tömböt
- kiértékeli a tömböt (lin. ker.)

Adatforgalom a függvények között



Főprogram

Feladat: A főprogram gondoskodik a beolvasás, a kiértékelés, kiíratás részek aktivizálásáról.

Bemenő adatok: `vector<string> t` - nevek tömbje, aminek forrása lehet fájl vagy a billentyűzet

Kimenő adatok: `bool l` - igen/nem válasz

Tevékenység: Az adatokat billentyűzetről vagy szöveges állományból lehet beolvasni. Ha a felhasználó szöveges állományt választja inputként, akkor meg kell adnia a már létező állomány nevét, amelyből az elválasztó jelekkel határolt sztringeket olvassa be a program. Ha a felhasználó a billentyűzetet választja, akkor be kell gépelnie a neveket, az adatbevitel végét egy „quit” szó megadásával jelzi. A szöveges állomány neve parancs sorban is megadható. Ilyenkor a program nem kérdezi meg az adatok forrását, hanem a szöveges állományt használja inputként. Az eredmény kiírása a konzol ablakban történik. A program újra és újra futtatható.

Definíció: helye `main.cpp`

```
int main(int argc, char *argv[])
```

A program függvényei

AllIbolya()

Feladat: Eldönti, hogy csupa "Ibolya" névből áll-e egy sztringeket tartalmazó tömb.

Bemenő adatok: `vector<string> t` - sztringek tömbje

Kimenő adatok: `bool l` - a válasz: visszatérési érték

Tevékenység: Optimista lineáris kereséssel megvizsgálja, hogy csupa "Ibolya" névből áll-e a megadott tömb (vector).

Definíció: helye `main.cpp`

```
bool AllIbolya(vector<string> t)
```

A lineáris keresés implementálásánál az *ind* eredményre most nincs szükség.

ReadFromFile()

Feladat: Sztringeket tartalmazó tömb (vector) feltöltése szöveges állományból.

Bemenő adatok: `string fname` - szöveges állomány neve

Kimenő adatok: `vector<string> t` - sztringek tömbje

`bool l` - sikerült-e a fájl nyitása: visszatérési érték

Tevékenység: Megnyitja a megadott szöveges állományt (sikertelen kísérlet esetén hibát jelez és hamis értéket ad vissza), majd a fájlból egymás után beolvassa az összes elválasztójelekkel határolt sztringet és elhelyezi azokat egy tömbben (vector-ban).

Definíció: helye `main.cpp`

```
bool ReadFromFile(const string &fname,
                  vector<string> &t)
```

Tesztelési terv

Tesztesetek a feladat alapján (fekete doboz tesztelés)

1. Egyetlen név sincs (üres fájl esete)
   (adat: [] – válasz: hamis)
2. Egyetlen Ibolya név van, semmi más.
   (adat: [Ibolya] – válasz: igaz)
3. Egyetlen nem Ibolya név van, semmi más.
   (adat: [Rózsa] – válasz: hamis)
4. Több Ibolya név van, semmi más.
   (adat: [Ibolya, Ibolya, Ibolya] – válasz: igaz)
5. Több nem Ibolya név van, semmi más.
   (adat: [Rózsa, Viola, Kata] – válasz: hamis)
6. Több nem Ibolya név, és a végén egyetlen Ibolya.
   (adat: [Rózsa, Viola, Ibolya] – válasz: hamis)
7. Több Ibolya név, és a végén egy nem Ibolya.
   (adat: [Ibolya, Rózsa, Ibolya, Viola] – válasz: hamis)
8. Több Ibolya név, és az elején egy nem Ibolya.
   (adat: [Rózsa, Ibolya, Viola, Ibolya] – válasz: hamis)
9. Több Ibolya név, és a közepén egy nem Ibolya.
   (adat: [Ibolya, Ibolya, Rózsa, Ibolya] – válasz: hamis)

Tesztesetek a kód alapján (fehér doboz tesztelés)

1. Adatbeviteli formák kipróbálása.
2. Hibás fájlnév megadása.
3. Főprogram ciklusának ellenőrzése: olyan bemenő adatokkal, amelyekre a ciklus egyszer sem fut le (Pl: üres fájl), pontosan egyszer fut le (Pl: nem Ibolya névvel kezdődő fájl), többször lefut, és a ciklus feltétel miatt lép ki (Pl: 4. fekete doboz teszt eset) , vagy a break miatt lép ki (Pl: 9. fekete doboz teszt eset).

.