

INTRODUCCIÓ:

En aquest exercici avaluarem arbres que representen expressions sobre valors de tipus string (de lletres minúscules) i els operadors de concatenació de dos strings i revessat de un string **Concat**, **Reverse**. En el cas de **Reverse**, que és un operador amb un sol operand, considerarem que aquest operand és sempre el fill esquerre. Per exemple, l'arbre **Reverse(Concat(Concat(a,ac),Reverse(b,cb)))** s'avalua a **abcaa**.

EXERCICI:

Implementeu una funció que, donat un arbre binari d'strings que representa una expressió correcta sobre strings de lletres minúscules i operadors **Concat**, **Reverse**, retorna la seva avaluació. Aquesta és la capcelera:

```
// Pre: t és un arbre no buit que representa una expressió correcta
//      sobre strings de lletres minúscules i els operadors Concat, Reverse.
// Post: Retorna l'avaluació de l'expressió representada per t.
string evaluate(const BinaryTree<string> &t);
```

Aquí tenim un exemple de paràmetre d'entrada de la funció i la corresponent sortida:

```
evaluate(Reverse(Concat(Concat(a,ac),Reverse(ab,)),)) = abcaa
```

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `Makefile`, `program.cpp`, `BinaryTree.hpp`, `evaluate.hpp`. Us falta crear el fitxer `evaluate.cpp` amb els corresponents `includes` i implementar-hi la funció anterior. Quan pugeu la vostra solució al jutge, només cal que pugeu un `tar` construït així:

```
tar cf solution.tar evaluate.cpp
```

Entrada

L'entrada té un nombre arbitrari de casos. Cada cas consisteix en una línia amb un string describint un arbre binari d'strings que representa una expressió correcta. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que implementeu la funció abans esmentada.

Sortida

Per a cada cas, la sortida conté la corresponent avaluació de l'arbre. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquesta avaluació. Només cal que implementeu la funció abans esmentada.

Exemple d'entrada

| | |
|---|---|
| Concat(Concat(Concat(bcb,baa),Reverse(caa,cb)),Reverse(bbb,cb)) | Reverse(abc,) |
| Reverse(Concat(Reverse(Reverse(a,)),c),) | Concat(Concat(cb,Reverse(ac,)),Concat(a,Concat(c,b))) |
| Concat(c,Reverse(acb,)) | Concat(Concat(Reverse(bcb,c),aca)) |
| Reverse(Reverse(Concat(aa,abb),),) | Concat(Reverse(cbc,),Concat(Reverse(cbc,),Reverse(bab,))) |
| | Concat(Reverse(c,cb),bab) |
| | Concat(Concat(bc,bcc),b) |
| | Concat(Concat(Concat(a,bbb),Reverse(bbb,)),Reverse(a,)) |

| | |
|---|---------------------------|
| Reverse (Reverse (ccb,),) | Exemple de sortida |
| ac | bcbbaaaacbbb |
| Reverse (Reverse (Reverse (Concat (cc, aca),),),) | ca |
| Reverse (cb,) | babccca |
| ba | cbca |
| Reverse (Reverse (Reverse (cba,),),) | aaabb |
| Concat (Reverse (Reverse (abc,),), Concat (Reverse (cb,), Concat (baa, ac))) | cha |
| cca | cbcaacb |
| aaa | ccbca |
| | cbccbcbab |
| | bcbccb |
| | abbbbbba |
| | ccb |
| | ac |
| | acacc |
| | bc |
| | ba |
| | abc |
| | abcbcbaaac |
| | cca |
| | aaa |

Observació

La vostra funció i subfuncions que creu han de treballar només amb arbres. Heu de trobar una solució **RECURSIVA** del problema. En les crides recursives, incloeu la hipòtesi d'inducció, és a dir una explicació del que es compleix després de la crida, i també la funció de fita/decreixement o una justificació de perquè la funció recursiva acaba.

En aquest exercici necessitareu un càlcul auxiliar que revessi un string. Podeu revessar string iterativament, però si ho feu recursivament obtindreu una mica més de nota.

Sobre 10 punts, la justificació val 1 punt, la solució ràpida val 6 punts mentre que la lenta val 4 punts, i revessar string recursiu dona 3 punts extra. Aquesta és la casuística completa de l'avaluació sobre 10 punts:

- Solució lenta: 4 punts.
- Solució lenta + justificació: 5 punts.
- Solució lenta + revessar string recursiu: 7 punts.
- Solució lenta + justificació + revessar string recursiu: 8 punts.
- Solució ràpida: 6 punts.
- Solució ràpida + justificació: 7 punts.
- Solució ràpida + revessar string recursiu: 9 punts.
- Solució ràpida + justificació + revessar string recursiu: 10 punts.

Informació del problema

Autor : PRO1

Generació : 2023-01-02 21:44:25

© Jutge.org, 2006–2023.

<https://jutge.org>