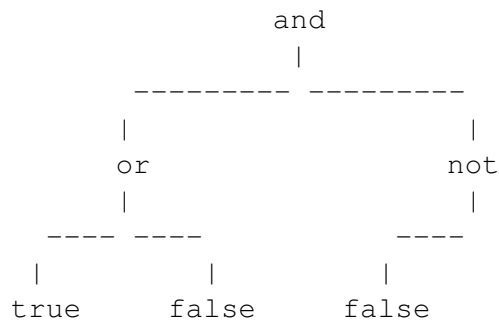


Arbre d'avaluacions d'expressió booleana

X34075_ca

INTRODUCCIÓ:

En aquest exercici considerarem arbres que representen expressions booleanes sobre valors **true**, **false** i els operadors booleanes **and**, **or**, **not**. En el cas de **not**, que és un operador amb un sol operand, considerarem que aquest operand és sempre el fill esquerre. Per exemple, el següent arbre representa l'expressió **(true or false) and (not(false))**. Aquesta expressió s'avalua a **true**.



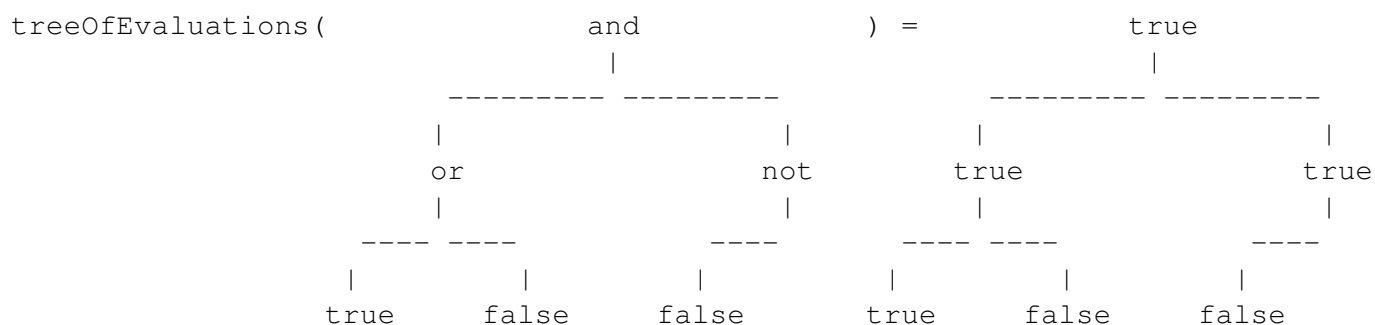
EXERCICI:

Implementeu una funció que, donat un arbre binari d'strings que representa una expressió booleana correcta sobre **true**,**false** i operadors **and**,**or**,**not**, retorna un nou arbre de booleans amb la mateixa estructura que l'inicial, i tal que, per a cada posició *p*, el nou arbre a posició *p* hi té l'avaluació de la expressió que hi ha a l'arbre original a posició *p*. Aquesta és la capcelera:

```

// Pre: t és un arbre no buit que representa una expressió booleana correcta
//       sobre els valors true,false i els operadors and,or,not.
// Post: Retorna un arbre binari de booleans
//       que té el mateix conjunt de posicions que t.
//       A més a més, per a cada posició p, el subarbre a posició p de t
//       representa una expressió que s'avalua a true si i només si
//       hi ha el valor true a la posició p de l'arbre retornat.
BinTree<bool> treeOfEvaluations (BinTree<string> t);
    
```

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



Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `main.cc`, `BinTree.hh`, `treeOfEvaluations.hh`. Us falta crear el fitxer `treeOfEvaluations.cc` amb els corresponents `includes` i implementar-hi la funció anterior. Només cal que pugeu `treeOfEvaluations.cc` al jutge.

Entrada

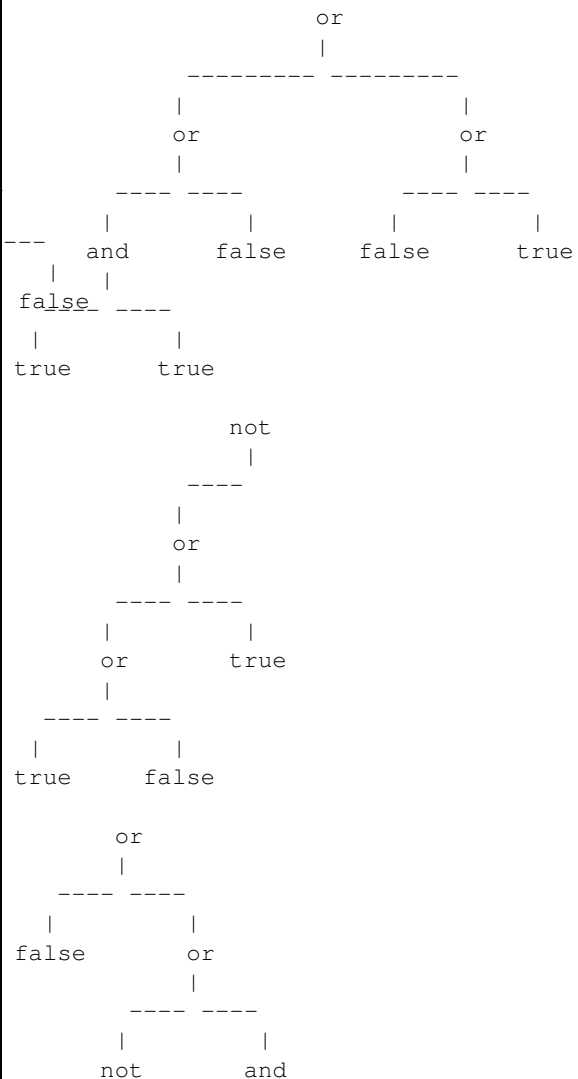
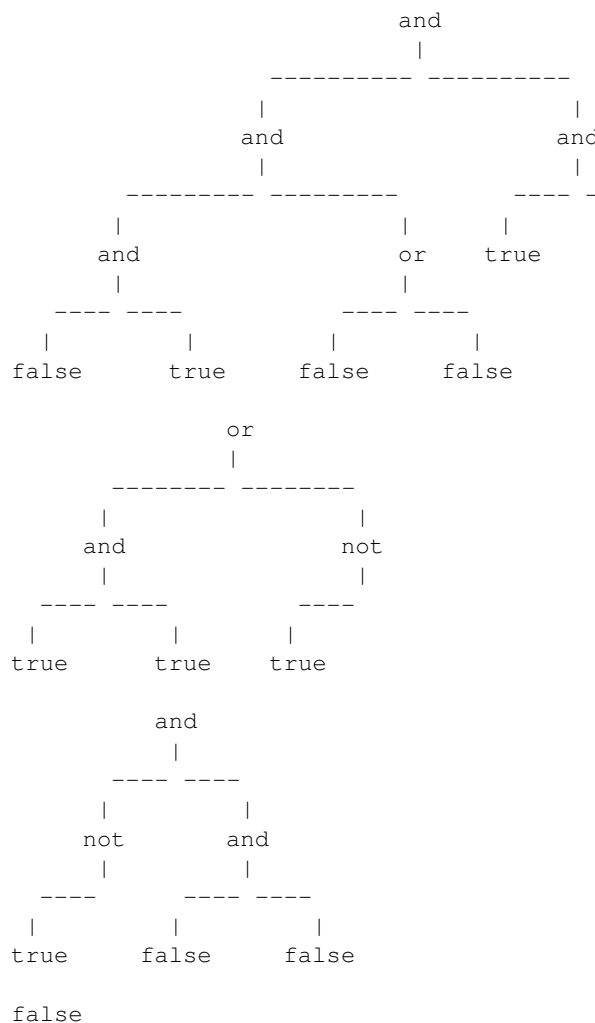
La primera línia de l'entrada descriu el format en el que es descriuen els arbres, o bé `IN-LINEFORMAT` o bé `VISUALFORMAT`. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre un arbre binari que representa una expressió booleana correcta. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que implementeu la funció abans esmentada.

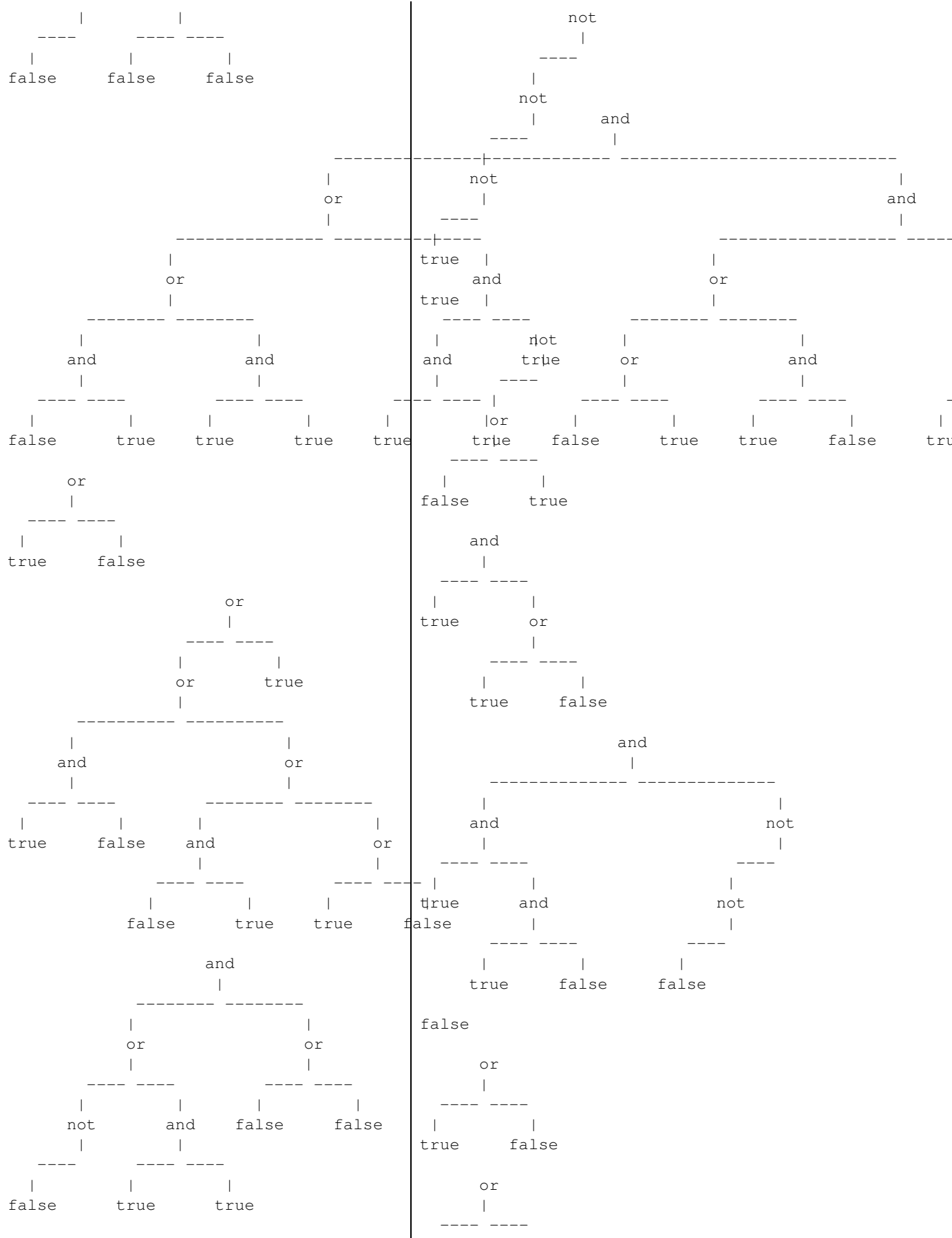
Sortida

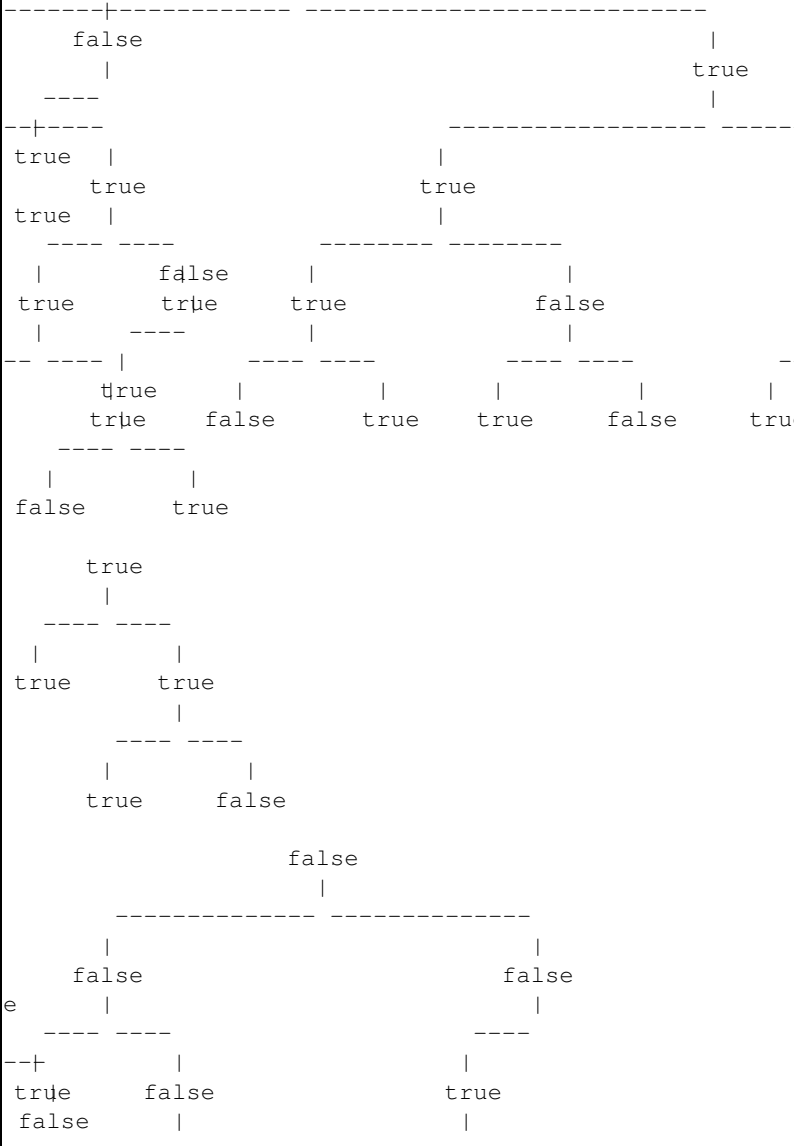
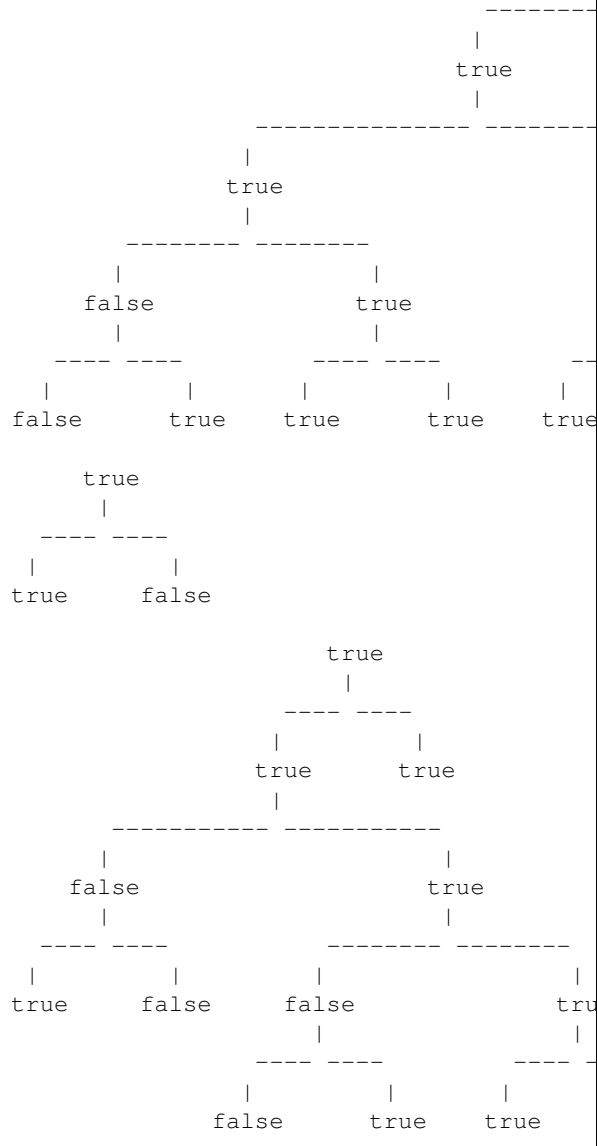
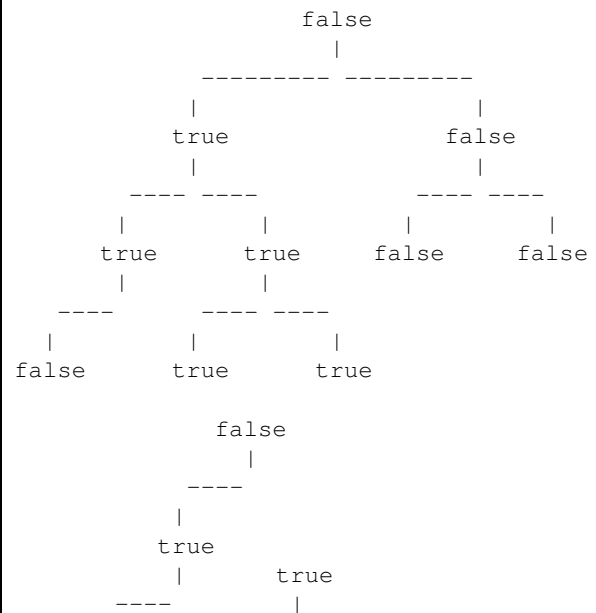
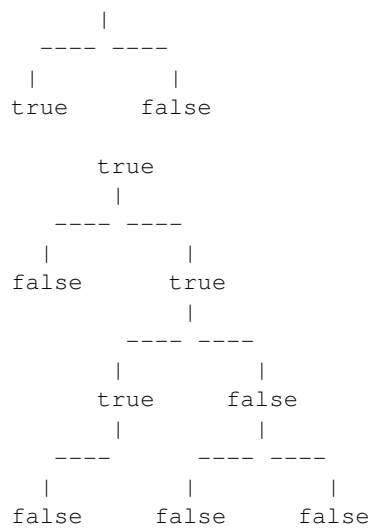
Per a cada cas, la sortida conté el corresponent arbre d'avaluacions. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquest resultat. Només cal que implementeu la funció abans esmentada.

Exemple d'entrada 1

VISUALFORMAT







Entenem com a solució ràpida una que és correcta, de cost lineal i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

Informació del problema

Autor : PRO2

Generació : 2023-10-23 23:22:06

© *Jutge.org*, 2006–2023.

<https://jutge.org>