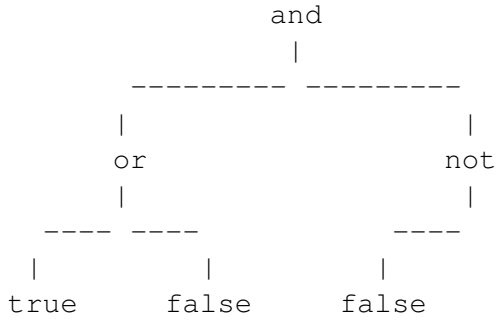
Arbre d'avaluacións d'expressió booleana — X34075_ca

INTRODUCCIÓ:

En aquest exercici considerarem arbres que representen expressions booleanes sobre valors **true, false** i els operadors booleans **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**.

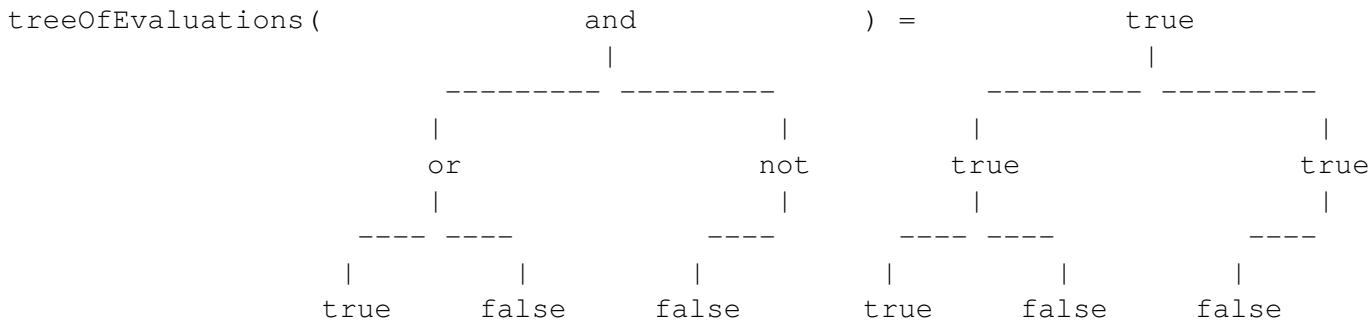


EXERCICI:

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.

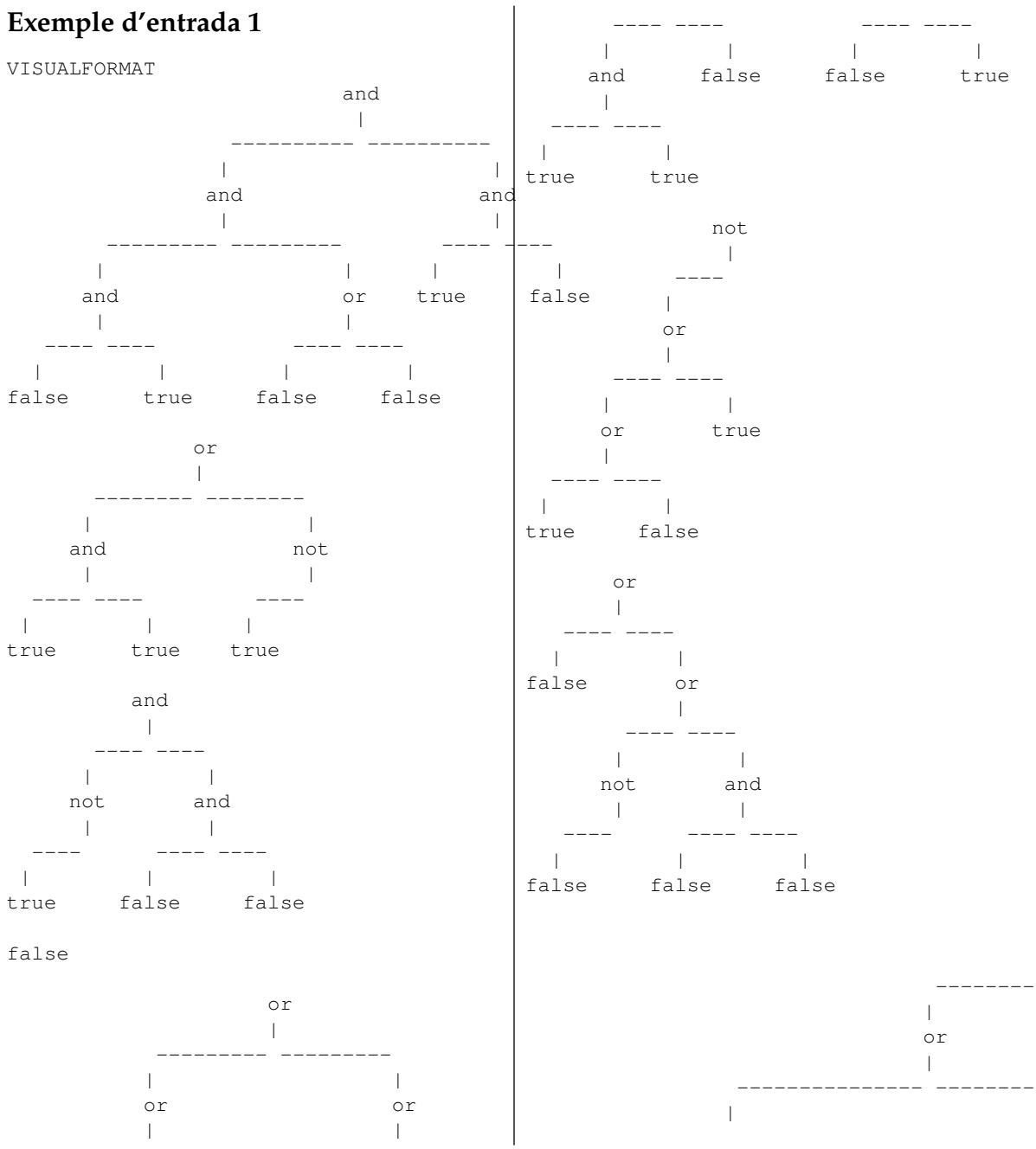
Entrada

La primera linia de l'entrada descriu el format en el que es descriuen els arbres, o bé INLINEFORMAT 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

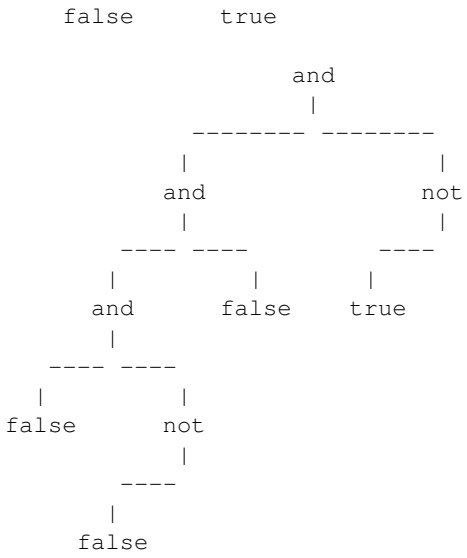
Per a cada cas, la sortida conté el corresponent arbre d'avaluacions. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquest resultat. Només cal que implementeu la funció abans esmentada.

Exemple d'entrada 1

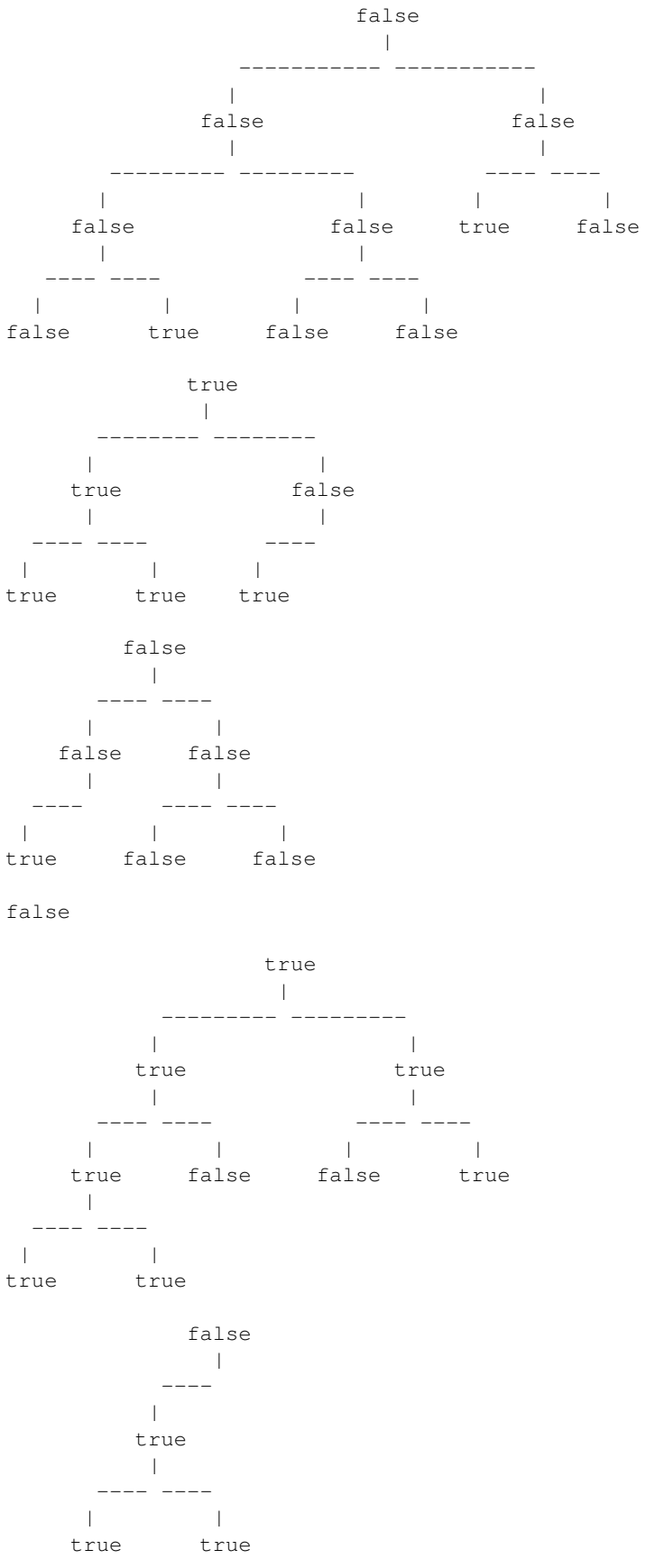


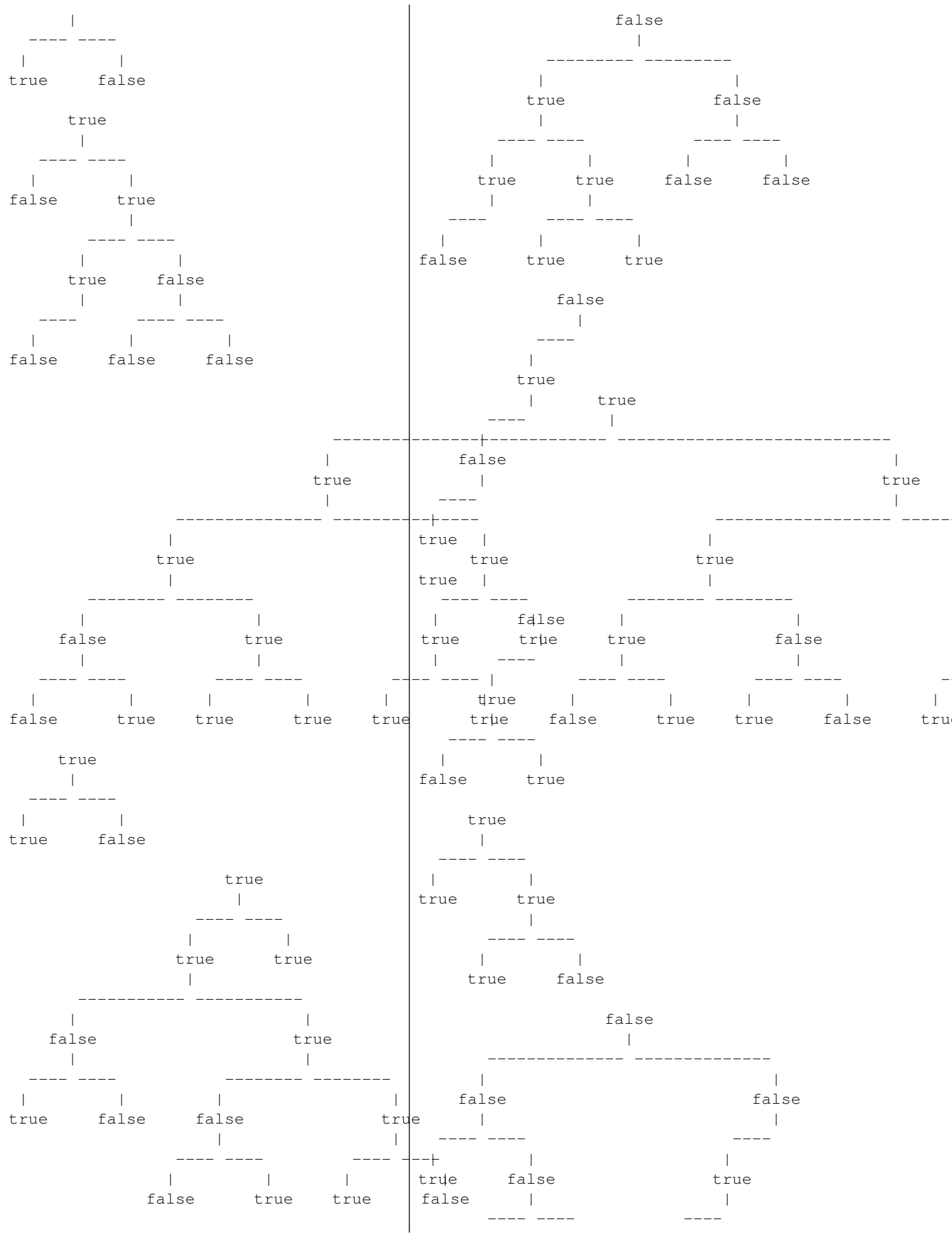
```
              or
              |
     -------- --------
     |               |
    and             and
     |               |
  ---- ----       ---- ----
  |        |      |        |      |
false     true   true     true   true

    or
    |
 ---- ----
 |        |
true    false

                     or
                     |
                 ---- ----
                 |        |
                 or      true
                 |
      ---------- ----------
      |                   |
     and                  or
      |                   |
  ---- ----         -------- --------
  |        |        |               |
true    false      and              or
                    |               |
               ---- ----        ---- ---- |
               |        |       |        |
             false     true    true    false

                    and
                     |
          -------- --------
          |               |
          or              or
          |               |
     ---- ----       ---- ----
     |        |      |        |
    not      and    false    false
     |        |
  ----     ---- ----
  |        |        |
false     true     true

                not
                 |
              ----
              |
             not
              |
          ----
          |
         not
          |
      ----
      |
     true
```

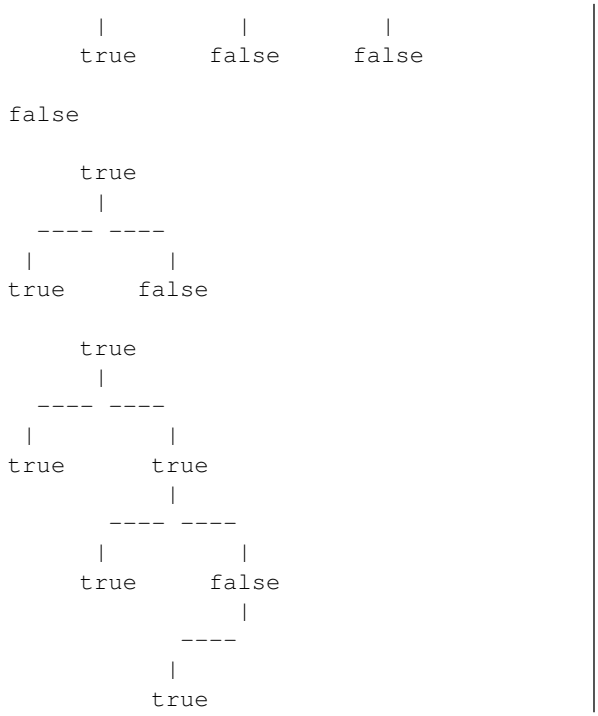
```
        and                  or
true     |                    |
    ---- ----        -------- --------
    |        not     |               |
   and       true    or             and
    |      ----      |               |
---- ---- |       ---- ----       ---- ----
   |or     |        |      |        |
   true  false     true   true    false    true
 ---- ----
 |        |
false    true

       and
        |
    ---- ----
    |        |
   true      or
             |
         ---- ----
         |        |
        true    false

                          and
                           |
         -------------- --------------
         |                           |
        and                         not
         |                           |
     ---- ----                     ----
     |        |                    |
    true     and                  not
              |                    |
          ---- ----            ----
          |        |           |
         true    false       false

false

       or
       |
   ---- ----
   |        |
  true    false

       or
       |
   ---- ----
   |        |
  true      or
            |
        ---- ----
        |        |
       true     not
                 |
              ----
              |
              or
              |
          ---- ----
          |        |
```

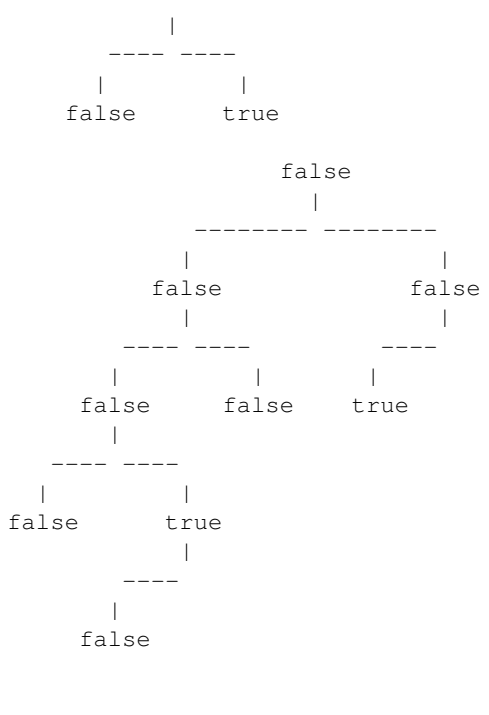


Exemple de sortida 1









Exemple d'entrada 2

```
INLINEFORMAT
and(and(and(false,true),or(false,false)),and(true,false))
or(and(true,true),not(true,))
and(not(true,),and(false,false))
false
or(or(and(true,true),false),or(false,true))
not(or(or(true,false),true),)
or(false,or(not(false,),and(false,false)))
and(or(or(and(false,true),and(true,true)),and(true,false)),true)
or(true,false)
or(or(and(true,false),or(and(false,true),or(true,false))),false)
and(or(not(false,),and(true,true)),or(false,false))
not(not(not(true,),),)
true
not(or(false,true),)
and(true,or(true,false))
and(and(true,and(true,false)),not(not(false,)))
false
or(true,false)
or(true,or(true,not(or(false,true),)))
and(and(and(false,not(false,)),false),not(true,))
```

Exemple de sortida 2

```
false(false(false(false,true),false(false,false)),false
true(true(false,true),false(true,))
false(false(true,),false(false,false))
false
true(true(true(true,true),false),true(false,true))
false(true(true(true,false),true),)
true(false,true(true(false,),false(false,false)))
true(true(true(false(false,true),true(true,true)),true
true(true(false(true,false),true(false(false,true),true
false(true(false(true,),),true(true,true)),false(false,fa
false(true(false(true,),),)
true
false(true(false,true),)
true(true,true(true,false))
false(false(true,false(true,false)),false(true(false,),
false)
true(true,false)
true(true,true(true,false(true(false,true),)))
false(false(false(false,true(false,)),false),false(true
```

Observació

La vostra funció i subfuncions que creeu han de treballar només amb arbres. Heu de trobar una solució **RECURSIVA** del problema.
Avaluació sobre 10 punts:

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

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

Autoria: PRO2

Generació: 2026-01-25T15:17:56.087Z