

INTRODUCCIÓ

El tipus genèric `BinaryTree` permet llegir i escriure arbres amb diferents formats. Per exemple, el format `VISUALFORMAT` és bastant intuïtiu i genera sortides d'aquesta mena:

```
      hello
      |
  ---- ----
 |           |
how         are
           |
         ---- ----
         |         |
        you       doing
```

Observeu que l'arbre representat és d'strings i té arrel "hello", i un fill esquerra amb arrel "how", que alhora té dos fills buits (els arbres buits no s'escriuen).

El format `INLINEFORMAT` és menys intuïtiu:

```
hello(how,are(you,doing))
```

Tot i així és més compacte i eficient.

El següent programa és un exemple de com indicar quin format volem utilitzar. Llegeix un arbre d'strings en format `INLINEFORMAT` i l'escriu en format `VISUALFORMAT`.

```
#include "BinaryTree.hpp"

int main() {
    BinaryTree<string> t;
    t.setInputFormat(BT::INLINEFORMAT);
    cin >> t;
    t.setOutputFormat(BT::VISUALFORMAT);
    cout << t << endl;
}
```

EXERCICI

Escriviu un programa que llegeixi una seqüència d'arbres d'entrada en un format (`INLINEFORMAT` o `VISUALFORMAT`) i l'escriu en l'altre format. La primera línia de l'entrada indica quants arbres hi ha i en quin format s'han de llegir.

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `Makefile`, `BinaryTree.hpp`. Us falta crear el fitxer `program.cpp` amb el programa esmentat. Quan pugueu la vostra solució al jutge, només cal que pugueu un tar construït així:

```
tar cf solution.tar program.cpp
```

Entrada

La primera línia de l'entrada té un natural n i un mot, o bé `INLINEFORMAT` o bé `VISUALFORMAT`, que indica el format en que s'han de llegir els arbres. Després venen n arbres d'strings sobre lletres minúscules en el format indicat.

Sortida

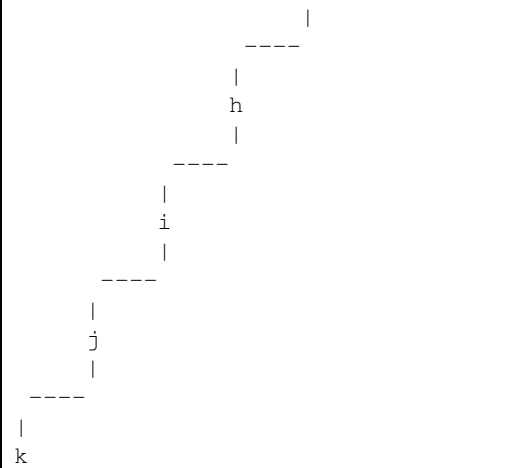
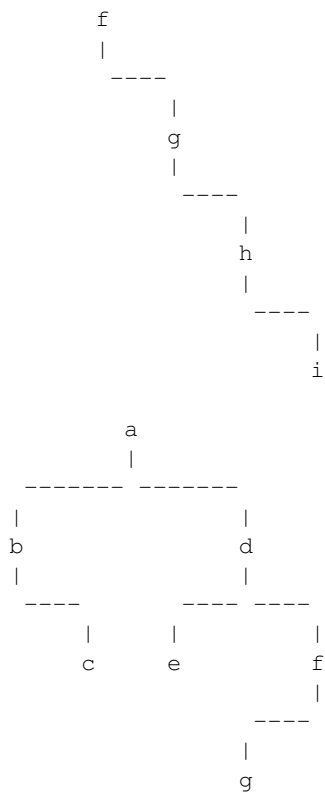
Per a cada arbre, s'ha d'escriure per la sortida l'arbre un altre cop, però en l'altre format.

Exemple d'entrada 1

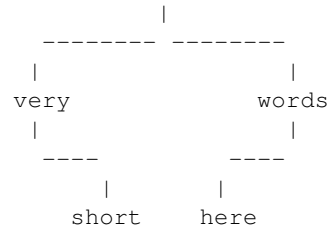
```
5 INLINEFORMAT
hello(how,are(you,doing))
a(b,c(e(f(g(h(i),),),),))
a(c,(e,(f,(g,(h,(i))))),b)
a(b,(c),d(e,f(g(h(i(j(k),),),),)))
Supercalifragilisticexpialidocious(very(short),words(here,))
```

Exemple de sortida 1

```
hello
  |
  -----
 |           |
how         are
  |           |
  you       doing
  |           |
              a
              |
              -----
             |           |
             b           c
             |           |
             |           |
             |           e
             |           |
             |           -----
             |           |
             |           f
             |           |
             |           -----
             |           |
             |           g
             |           |
             |           -----
             |           |
             |           h
             |           |
             |           -----
             |           |
             |           i
             |           |
             |           a
             |           |
             |           -----
             |           |
             |           c           b
             |           |
             |           -----
             |           |
             |           e
             |           |
             |           -----
             |           |
```

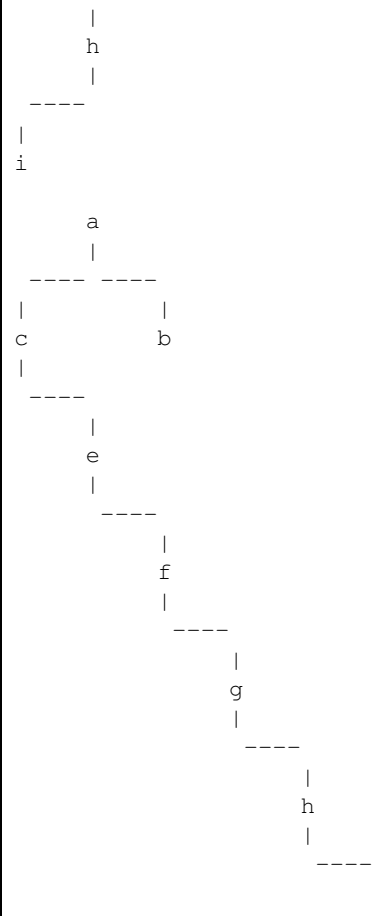
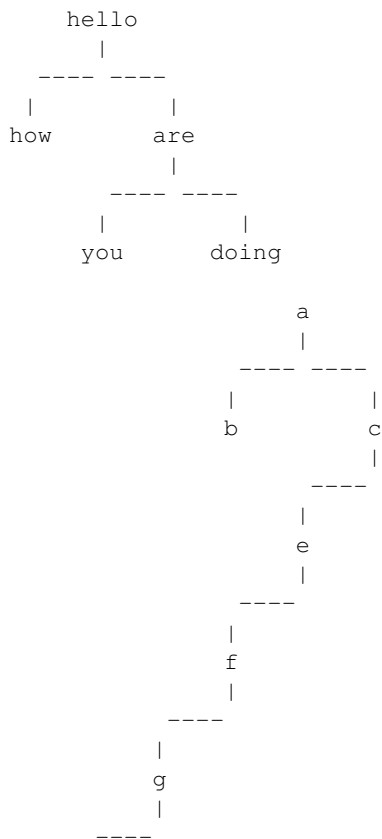


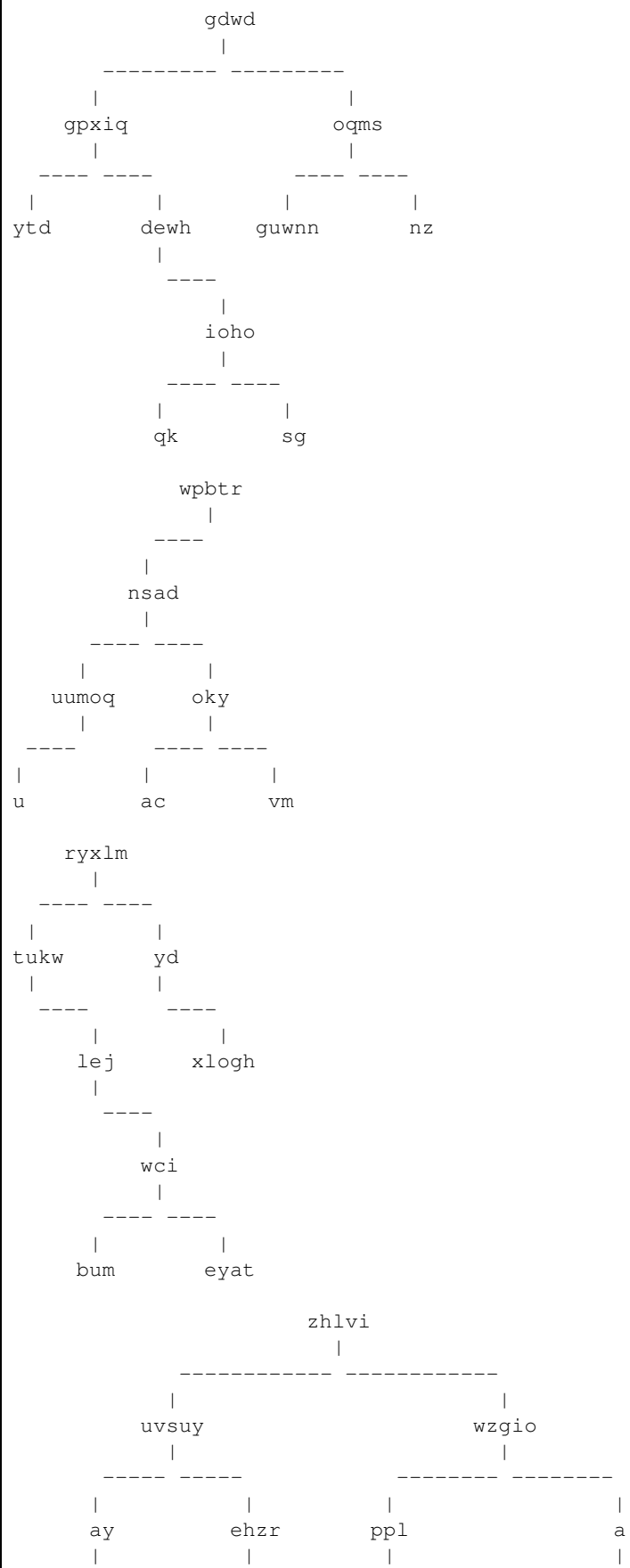
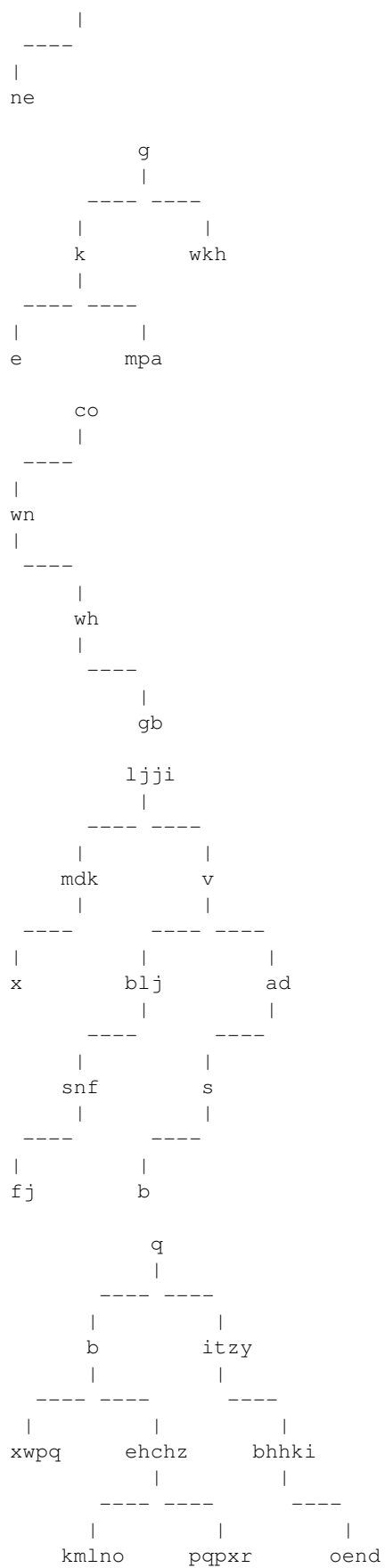
Supercalifragilisticexpialidocious



Exemple d'entrada 2

5 VISUALFORMAT





```

    -----
    |           |           |           |
    eim      fskpg          zuc      wpha
    |           |
  
```

```

    -----
    |           | bip     | lu     | kg
    |           | wdtxj
  
```

Exemple d'entrada 4

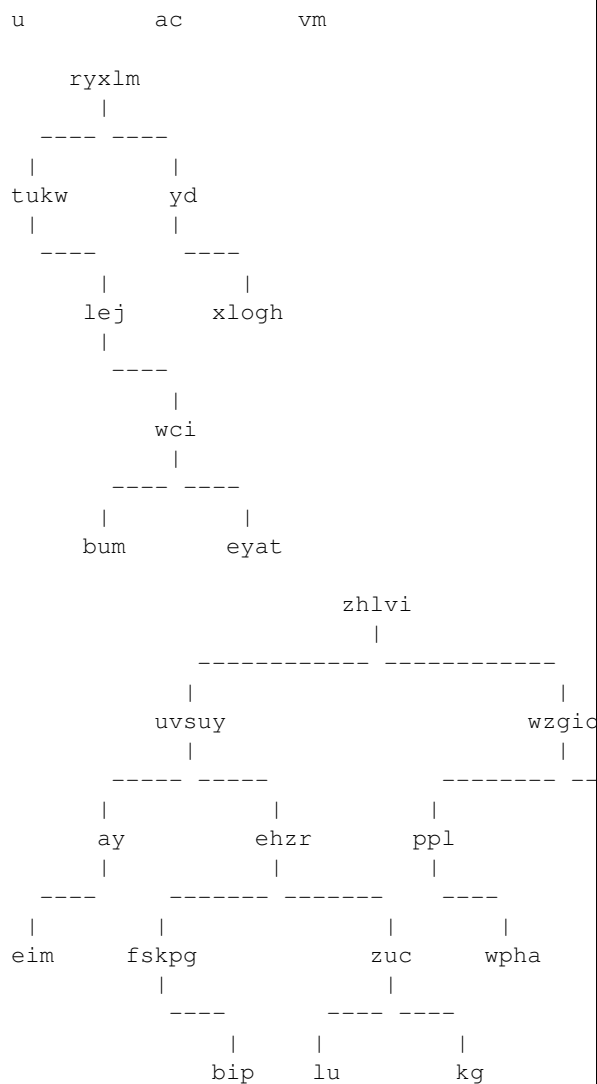
10 VISUALFORMAT

```

    lr
    |
    -----
    |         |
    qb       rxsjy
    |
    -----
    |         |
    arz       ky
    |
    -----
    |         |
    dqs       rjm
    |
    efsar
    |
    -----
    |
    ne
    |
    g
    |
    -----
    |         |
    k         wkh
    |
    -----
    |         |
    e         mpa
    |
    co
    |
    -----
    |
    wn
    |
    -----
    |
    wh
    |
    -----
    |
    gb
    |
    ljji
    |
    -----
    |         |
    mdk       v
    |         |
    -----
  
```

```

    x         blj         ad
    |         |         |
    -----
    |         |         |
    snf       s
    |         |
    -----
    |         |
    fj        b
    |
    q
    |
    -----
    |         |
    b         itzy
    |         |
    -----
    |         |         |
    xwpq      ehcz      bhhki
    |         |         |
    -----
    |         |         |
    kmlno     pqpxr     oend
    |
    gdwd
    |
    -----
    |         |         |
    gpxiq     oqms
    |         |
    -----
    |         |         |         |
    ytd       dewh      guwnn     nz
    |
    -----
    |
    ioho
    |
    -----
    |         |
    qk        sg
    |
    wpbtr
    |
    -----
    |
    nsad
    |
    -----
    |         |
    uumoq     oky
    |         |
    -----
  
```



Exemple de sortida 4

```

lr(qb(arz,ky(dqs,rjm)),rxsjy)
efsar(ne,)
g(k(e,mpa),wkh)
co(wn(,wh(,gb)),)
ljji(mdk(x,),v(blj(snf(fj,)),),ad(s(b,)))
q(b(xwpq,ehcz(kmlno,pppxr)),itzy(,bhhki(,oend)))
gdwd(gpxiq(ytd,dewh(,ioho(qk,sg))),oqms(guwnn,nz))
wpbtr(nsad(uumoq(u,),oky(ac,vm)),)
ryxlm(tukw(,lej(,wci(bum,eyat))),yd(,xlogh))
zhlvi(uvsuy(ay(eim,),ehzr(fskpg(,bip),zuc(lu,kg))),wzgi

```

Observació

Després de llegir n i el format, cal que afegiu la següent instrucció en el vostre programa per a que la lectura de l'arbre comenci en una línia nova, doncs el format `VISUALFORMAT` llegeix una seqüència de línies amb `getline`.

```
cin.ignore();
```

A més a més, recordeu afegir un salt de línia després d'escriure cada arbre.

Informació del problema

Autor : PRO1

Generació : 2023-02-25 19:31:36

© Jutge.org, 2006–2023.

<https://jutge.org>