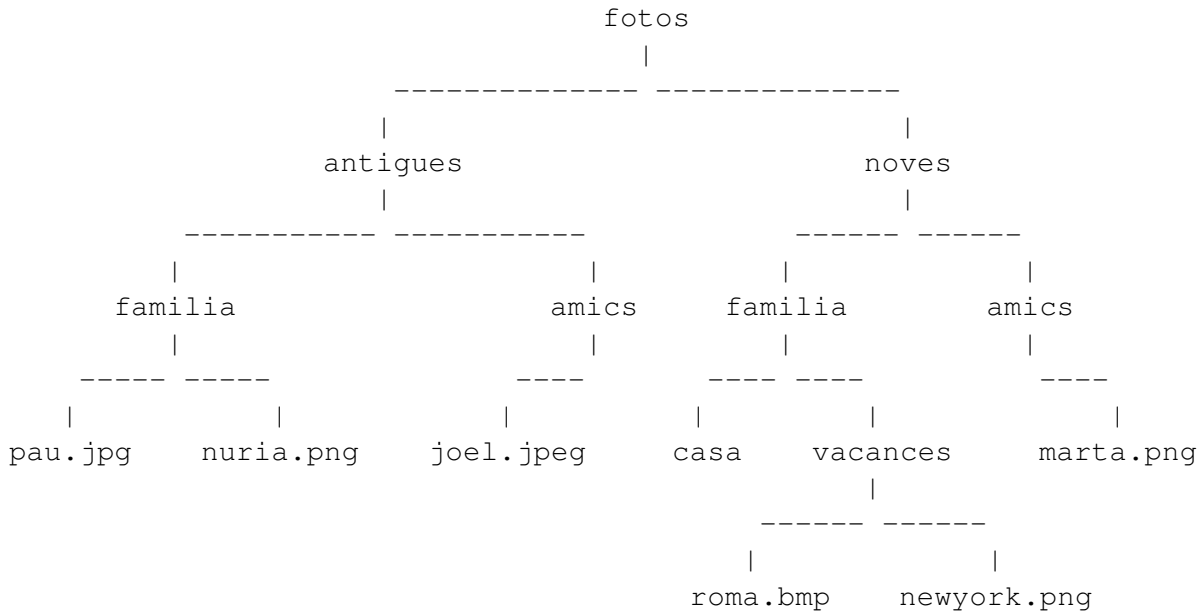
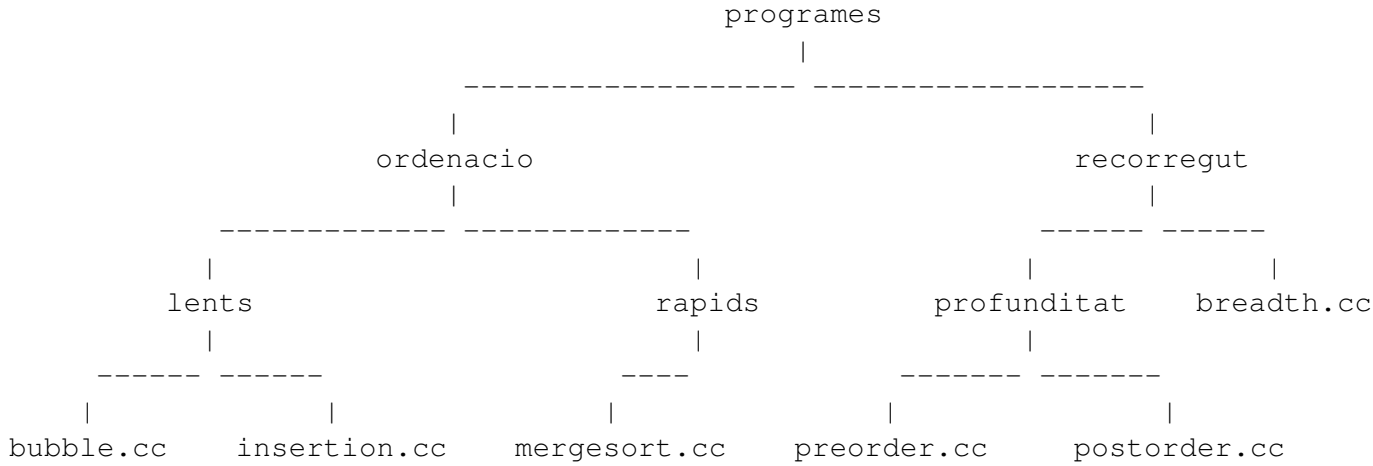
Fitxers amb la mateixa extensió X18348_ca

En aquest exercici, considerarem arbres binaris d'strings que representen arbres de directoris i fitxers. Observeu el següent exemple:



En un arbre de directoris i fitxers hi ha dos tipus de nodes: directoris i fitxers. L'string d'un directori és no buit i està format per lletres minúscules. L'string d'un fitxer està format per una primera seqüència no buida de lletres minúscules, seguida del caràcter '.', seguit d'una segona seqüència no buida de lletres minúscules. Aquesta segona seqüència s'anomena la extensió del fitxer. Els fitxers han de ser necessàriament fulles de l'arbre. Els directoris poden ser nodes interns i també fulles.
Heu d'implementar una funció que rep un arbre de directoris i fitxers, i retorna un booleà indicant si tots els fitxers tenen exactament la mateixa extensió. En l'exemple anterior, la funció ha de retornar `false`, doncs tenim més d'una extensió diferent: `jpg`, `png`, `jpeg` i `bmp`. En canvi, en l'exemple següent, la funció ha de retornar `true`, perquè totes les extensions son la mateixa: `cc`.



Aquesta és la capcelera:

```
// Pre:  Els nodes de t o bé son strings no buits de lletres minuscules, o bé
//       son de la forma "s.e", on s i e son strings no buits de lletres minúsc
//       En l'últim cas, el node ha de ser una fulla, i e s'anomena la extensió
// Post: Retorna cert si i només si totes les extensions de les fulles de t son
bool sameExtension(const BinaryTree<string> &t);
```

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `Makefile`, `program.cpp`, `BinaryTree.hpp`, `sameExtension.hpp`. Us falta crear el fitxer `sameExtension.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 sameExtension.cpp
```

Observació1: Donat un string `s` i un índex `i`, podeu utilitzar `s.substr(i)` per a obtenir el substring que és el sufix de `s` a partir de posició `i`.
Observació2: Convindrà que penseu bé en quines funcions auxiliars us convé implementar per tal que us surti una solució no massa complicada.

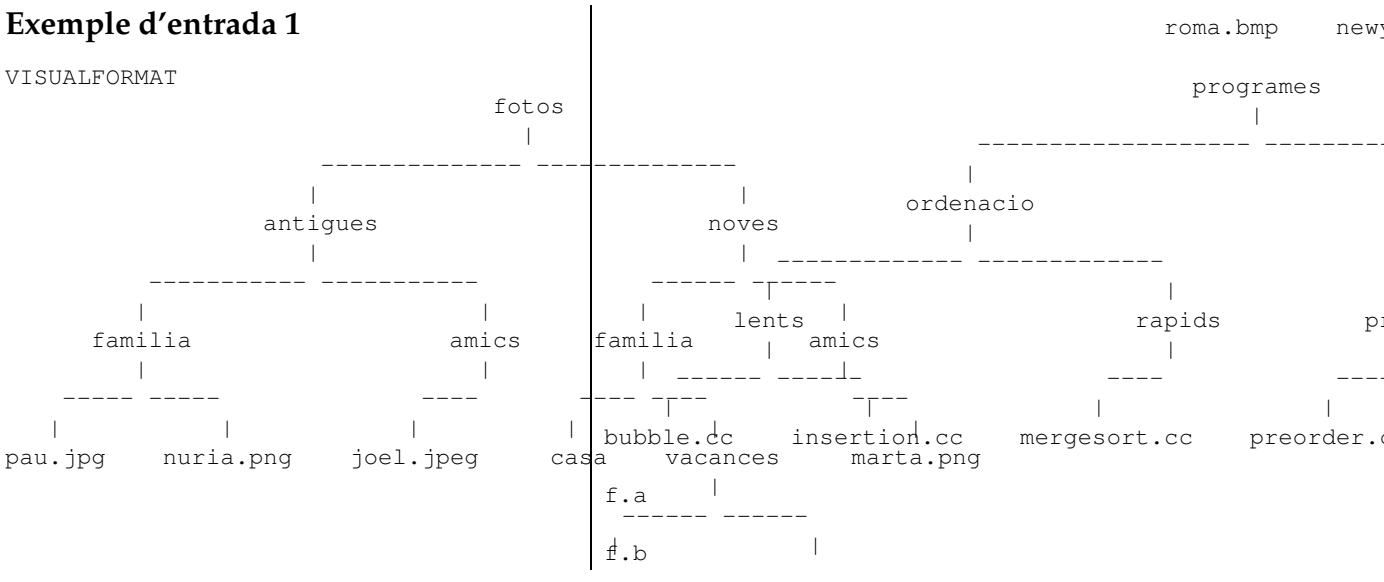
Entrada

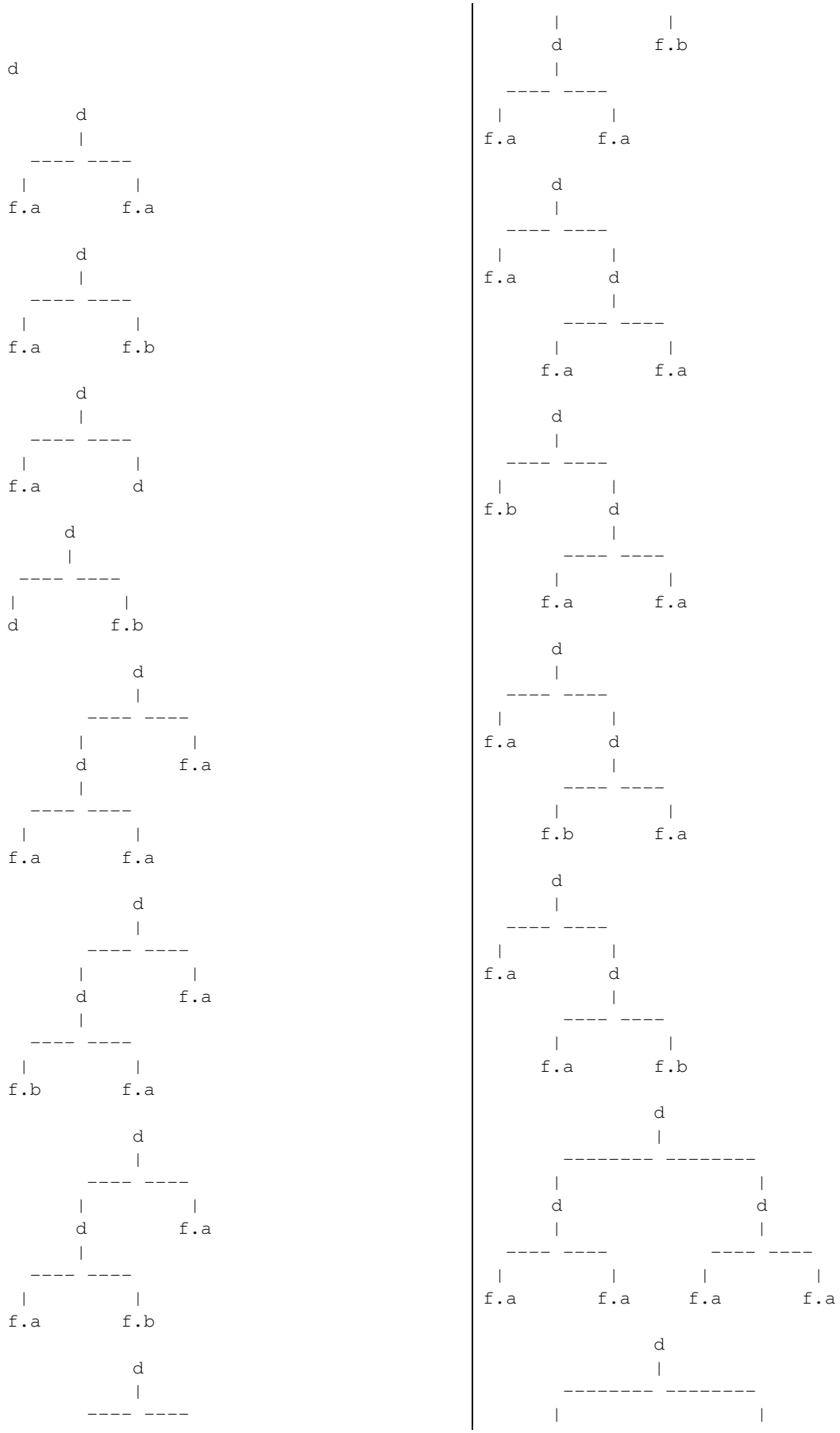
La primera linia de l'entrada descriu el format en el que es descriuen els arbres, o bé INLINEFORMAT o bé VISUALFORMAT. Després venen un nombre arbitrari de casos. Cada cas consisteix en una descripció d'un arbre binari d'strings que representa un arbre de directoris i fitxers. Fixeu-vos en que el programa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

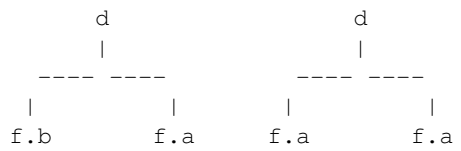
Sortida

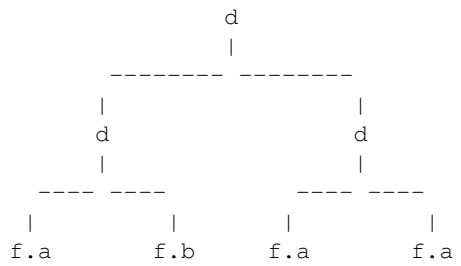
Per a cada cas, la sortida conté `true` si totes les extensions del corresponent arbre d'entrada son iguals, i `false` en cas contrari. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure el resultat. Només cal que implementeu la funció abans esmentada.

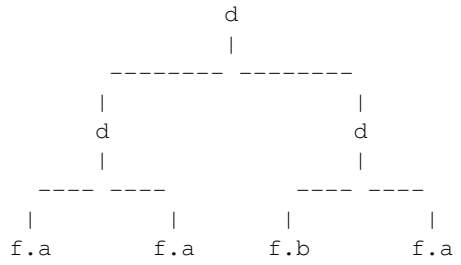
Exemple d'entrada 1

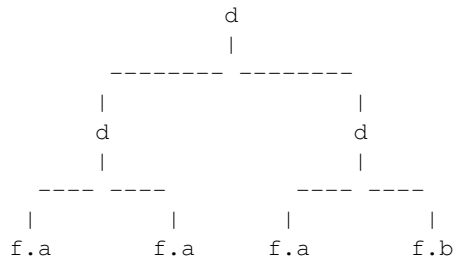


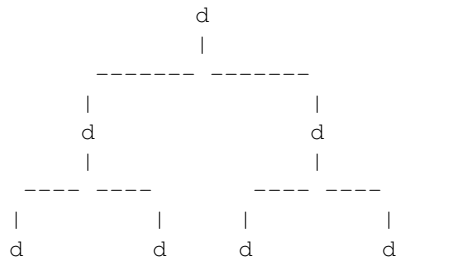

d
d
f.a f.a
d
f.a f.b
d
f.a d
d
d f.b
d
d f.a
f.a f.a
d
d f.a
f.b f.a
d
d f.a
f.a f.b
d
f.a f.a
d
f.b d
f.a f.a
d
f.a d
f.b f.a
d
f.a d
f.a f.b
d
d d
f.a f.a f.a f.a
d

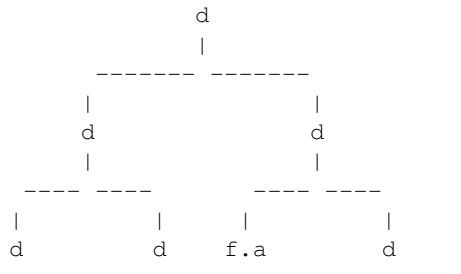

d d
f.b f.a f.a f.a

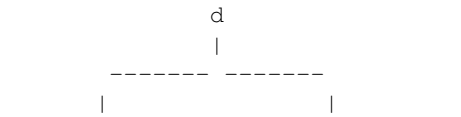

d
d d
f.a f.b f.a f.a

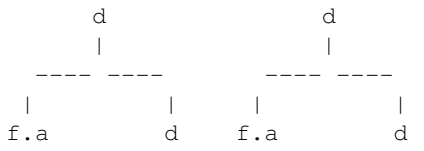

d
d d
f.a f.a f.b f.a

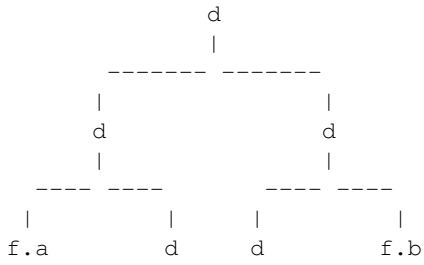

d
d d
f.a f.a f.a f.b


d
d d
d d d d


d
d d
d d f.a d


d
d


d d
f.a d f.a d


d
d d
f.a d d f.b

Exemple de sortida 1

```
false
true
true
true
true
true
false
true
true
true
false
false
false
true
false
false
false
true
false
false
false
false
true
true
true
false
```

Exemple d'entrada 2

```
INLINEFORMAT
fotos(antigues(familia(pau.jpg,nuria.png),amics(joel.jpeg)),noves(familia(casa,vacances(roma.bmp
programes(ordenacio(lents(bubble.cc,insertion.cc),rapids(mergesort.cc)),recorregut(profunditat(p
f.a
f.b
d
d(f.a,f.a)
d(f.a,f.b)
d(f.a,d)
d(d,f.b)
d(d(f.a,f.a),f.a)
d(d(f.b,f.a),f.a)
d(d(f.a,f.b),f.a)
d(d(f.a,f.a),f.b)
d(f.a,d(f.a,f.a))
d(f.b,d(f.a,f.a))
d(f.a,d(f.b,f.a))
d(f.a,d(f.a,f.b))
d(d(f.a,f.a),d(f.a,f.a))
d(d(f.b,f.a),d(f.a,f.a))
d(d(f.a,f.b),d(f.a,f.a))
d(d(f.a,f.a),d(f.b,f.a))
d(d(f.a,f.a),d(f.a,f.b))
d(d(d,d),d(d,d))
d(d(d,d),d(f.a,d))
d(d(f.a,d),d(f.a,d))
d(d(f.a,d),d(d,f.b))
```

Exemple de sortida 2

```
false
true
true
true
true
true
false
true
true
true
false
false
false
true
false
false
false
true
false
false
false
false
true
true
true
false
```

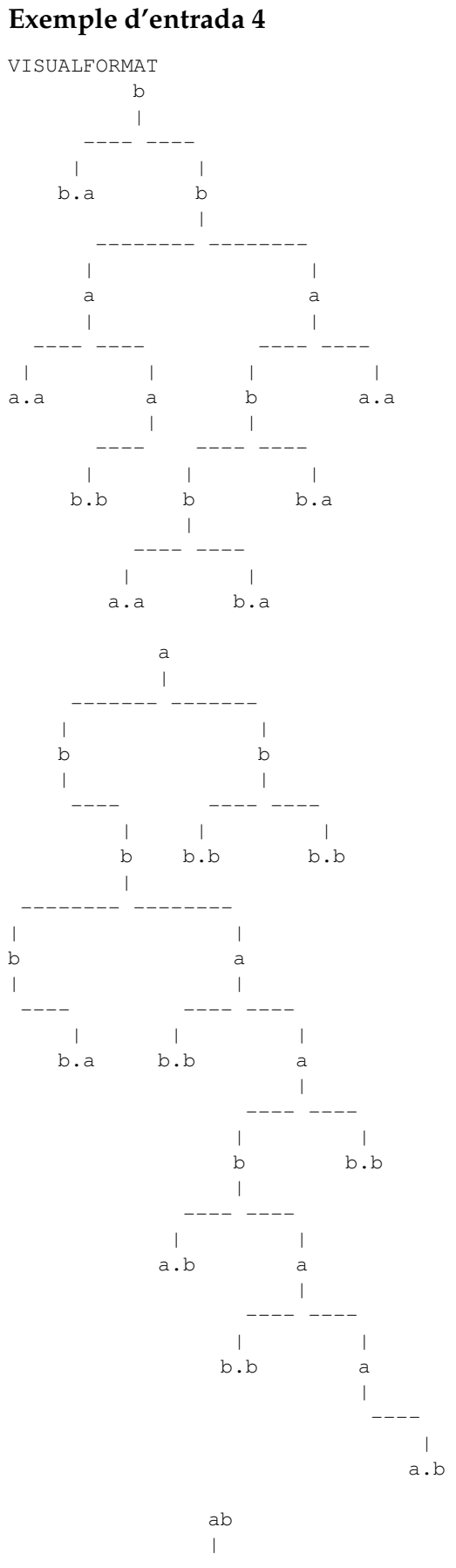
Exemple d'entrada 3

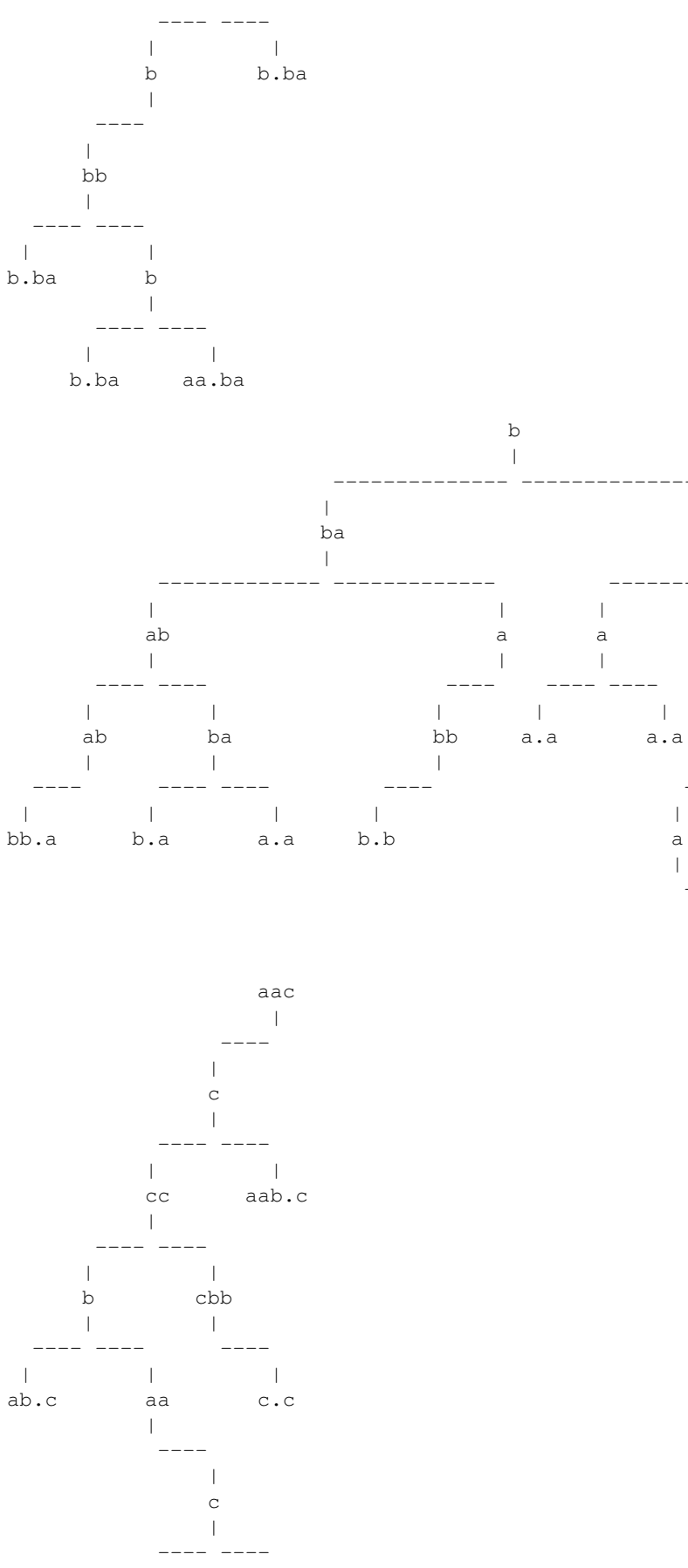
```
INLINEFORMAT
b(b.a,b(a(a.a,a(b.b,)),a(b(b(a.a,b.a),b.a),a.a)))
a(b(,b(b(,b.a),a(b.b,a(b(a.b,a(b.b,a(,a.b)),b.b)))),b(b.b,b.b))
ab(b(bb(b.ba,b(b.ba,aa.ba)),),b.ba)
b(ba(ab(ab(bb.a,),ba(b.a,a.a)),a(bb(b.b,),b(a(a.a,a.a),aa(b(a(,b.a),),ba(a.a,b.a))))
aac(c(cc(b(ab.c,aa(,c(cab(,a.c),a.c))),cbb(,a.c)),aab.c),)
dadb(aca(ab.ddba,a.ddba),bc(cbc.ddba,bbd(,ca(bbac(,c.ddba),bb(bc(c(b.ddba,b.ddba),caa.ddba),bbac.
ac(cbc(cab.a,aac(abc.baa,)),ca(bac(ab(acb(aab(a.a,aaa.c),a.cbb),aca.ac),cc(,c(ba.c,))),ca.bbc))
y.ze
adc(dba.cda,c(a.da,b(bcbd.da,bb.cda)))
cba(bb(bdbd.bcd,dd.bcd),adb(cd(bbac(ba.bcd,dbac.bcd),cc(ba.ad,ddba(caac(ad.ad,ddcb.bcd),bcc.ad))
dbd(,babd(ad.dad,cad(bb(bcad(dbad.bdb,ab.dca),),d.acb)))
cdd(cd(c(c.caad,acd(dba(c.ada,bb(b.caad,bbdd.caad)),)),aa(dac(d.d,),caa(,acbb(bcd.ca,dacb.caad))
tgt(nupy(cuq(ub(i.pw,ck.pw),),szbe.pw),m(vfa.pw,djeq(g.pw,azq(,pjvo(dp.pw,)))))
bda(b(d(,bc.b),db.b),ca(,a(dbc(dac.d,b(d(c(,abd.aca),dd(cda.d,add(,c(aaa.aca,cc.b)))),bdab.d)),b
```

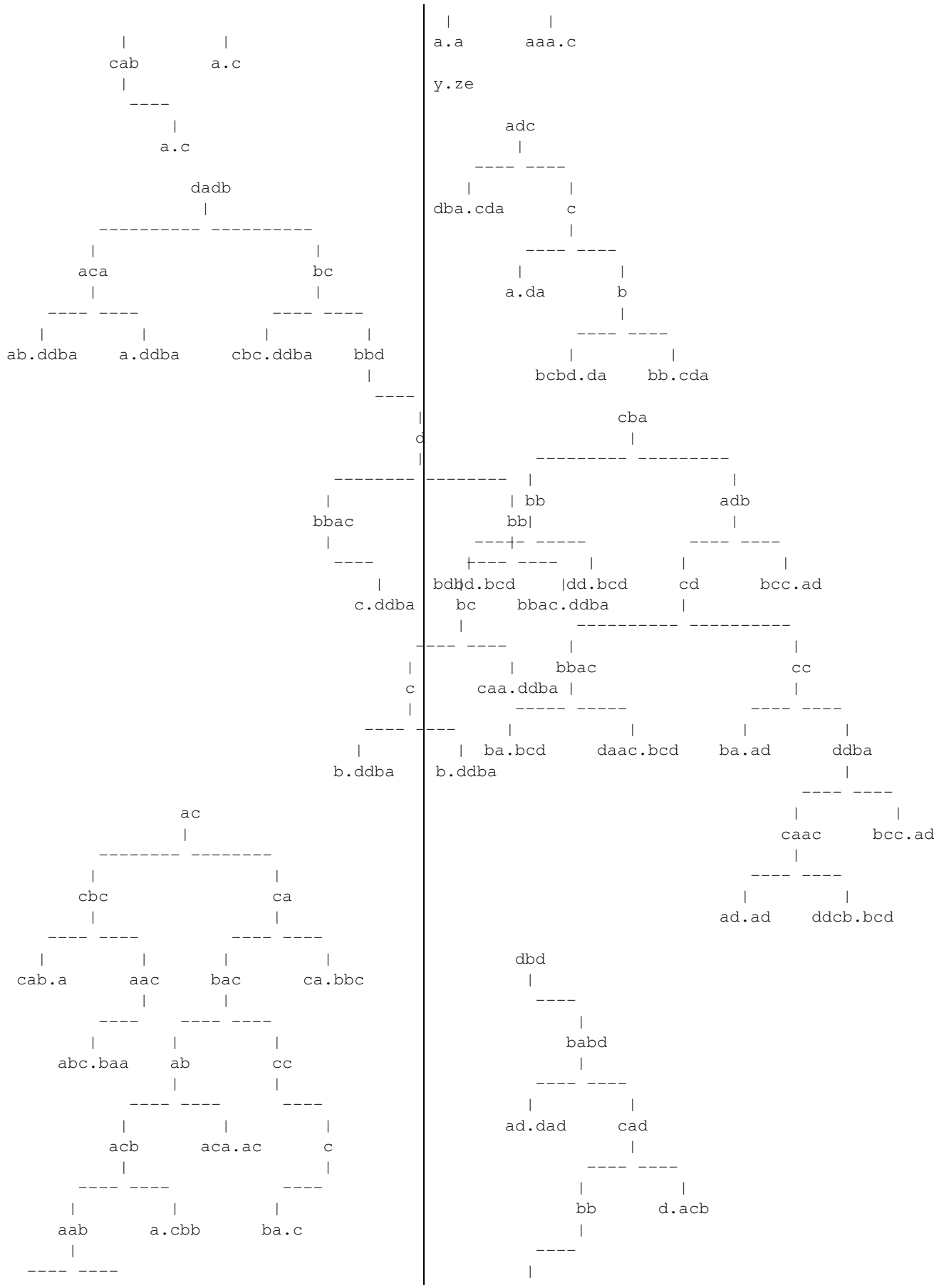
Exemple de sortida 3

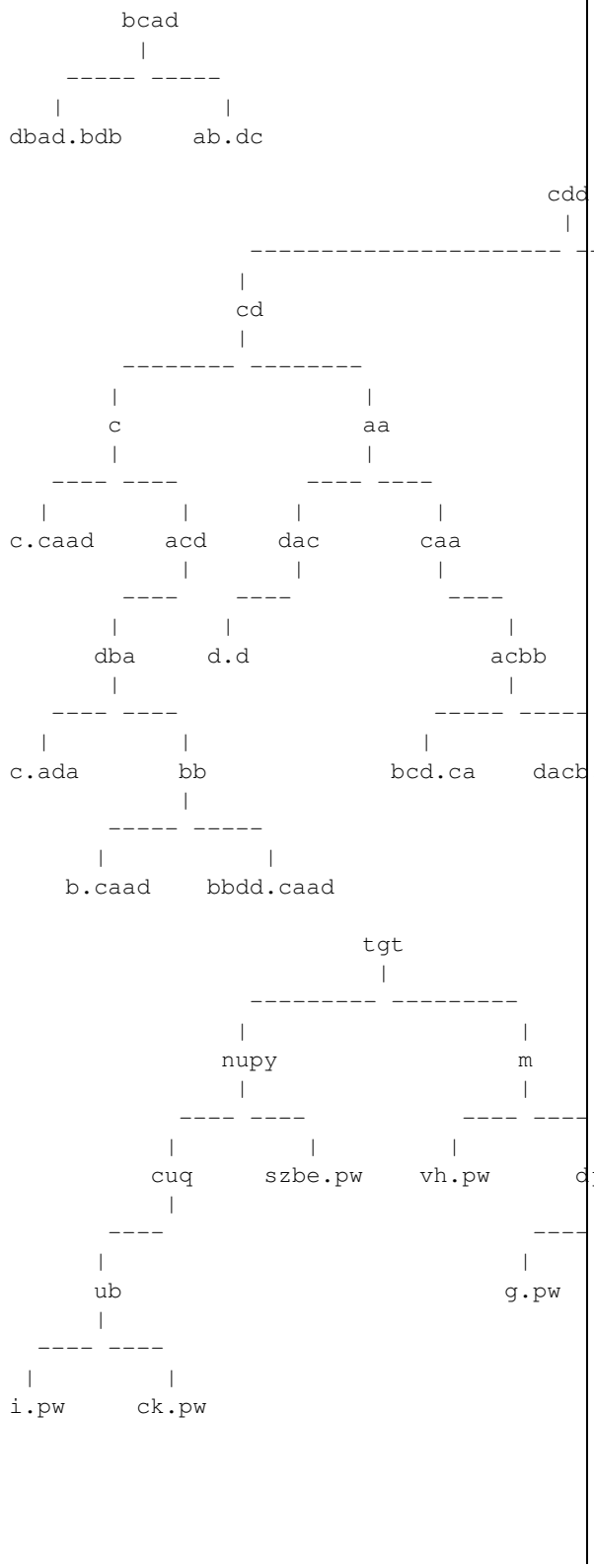
```
false
false
true
false
true
true
false
true
false
false
false
false
true
false
```

Exemple d'entrada 4



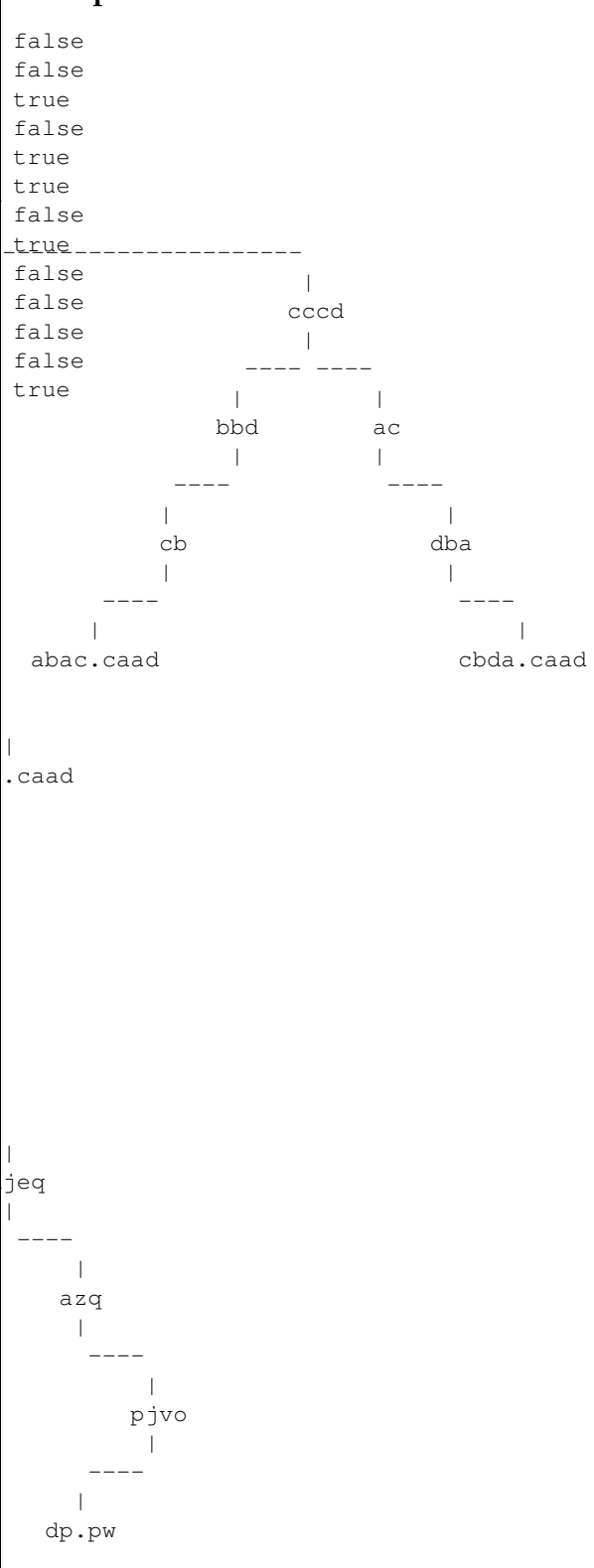



cab
a.c
a.c
dadb
aca
bc
ab.ddba
a.ddba
cbc.ddba
bbd
bbac
c.ddba
b.ddba
a.a
aaa.c
y.ze
adc
dba.cda
c
a.da
b
bcbd.da
bb.cda
cba
bb
adb
bdbd.bcd
dd.bcd
cd
bcc.ad
bc
bbac.ddba
bbac
cc
caa.ddba
ba.bcd
daac.bcd
ba.ad
ddba
b.ddba
caac
bcc.ad
ad.ad
ddcb.bcd
ac
cbc
ca
cab.a
aac
bac
ca.bbc
abc.baa
ab
cc
acb
aca.ac
c
aab
a.cbb
ba.c
dbd
babd
ad.dad
cad
bb
d.acb



Exemple de sortida 4

```
false
false
true
false
true
true
false
true
false
false
false
false
true
```



Observació

Podeu implementar les funcions auxiliars que considereu necessàries, però la vostra funció i subfuncions que creeu han de treballar només amb arbres. Per a resoldre aquest exercici, no

us permetem utilitzar cap mecanisme d'enmagatzemament massiu de dades a part del propi arbre binari que es rep com a paràmetre, i els strings que siguin necessaris. Us recomanem, doncs, que resolgueu aquest exercici recursivament. En les crides recursives, incloeu la hipòtesi d'inducció, és a dir una explicació del que es cumpleix després de la crida, i també la funció de fita/decreixement o una justificació de perquè la funció recursiva acaba. Sí que podeu utilitzar un enfocament iteratiu per a recorrer els strings, si ho considereu oportú.
Avaluació sobre 10 punts:

- Solució lenta: 6 punts.
- Solució lenta + justificació: 8 punts.
- solució ràpida: 8 punts.
- solució ràpida + justificació: 10 punts.

Entenem com a solució lenta una que és correcta i capaç de superar els jocs de proves públics. Entenem com a solució ràpida una que és correcta i capaç de superar els jocs de proves públics i privats. La justificació val 2 punts i consisteix en definir correctament les PRE/-POST de les funcions auxiliars que afegiu i en definir correctament les hipòtesis d'inducció i funcions de fita.

Informació del problema

Autor : PRO1
Generació : 2023-06-13 21:20:45