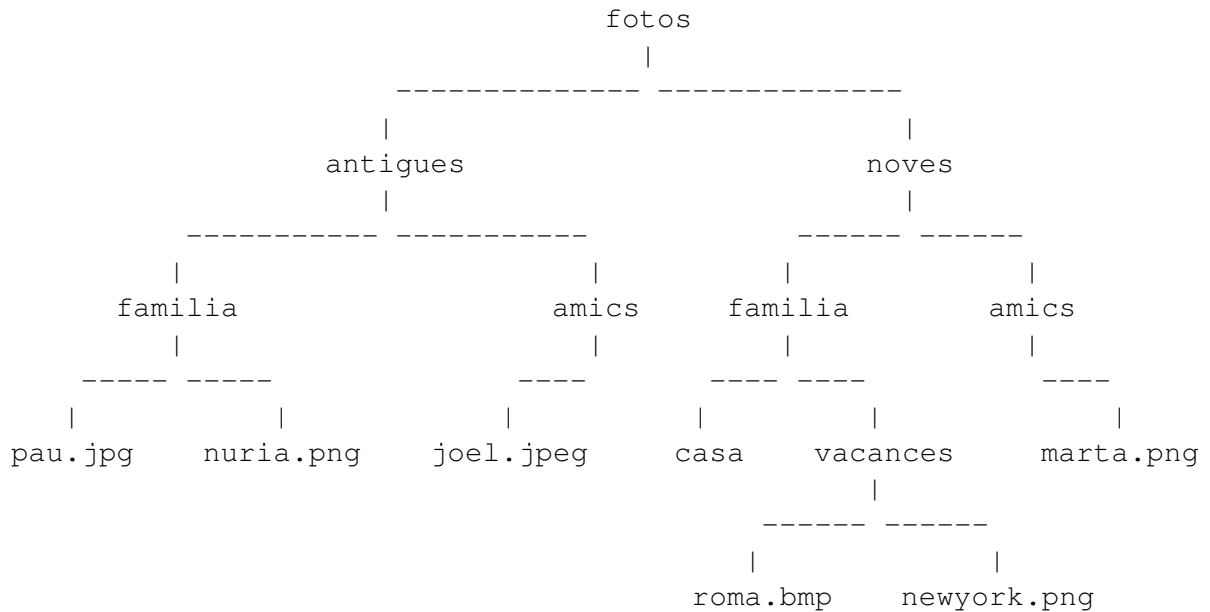

Esborra fitxers amb una extensió

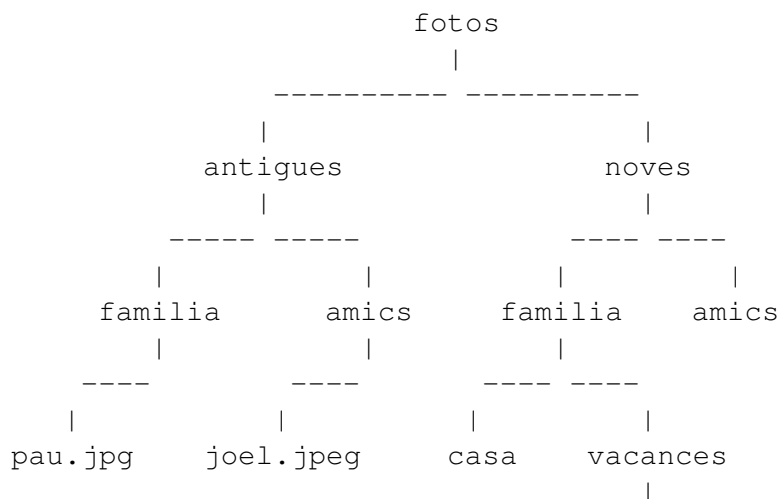
X24232_ca

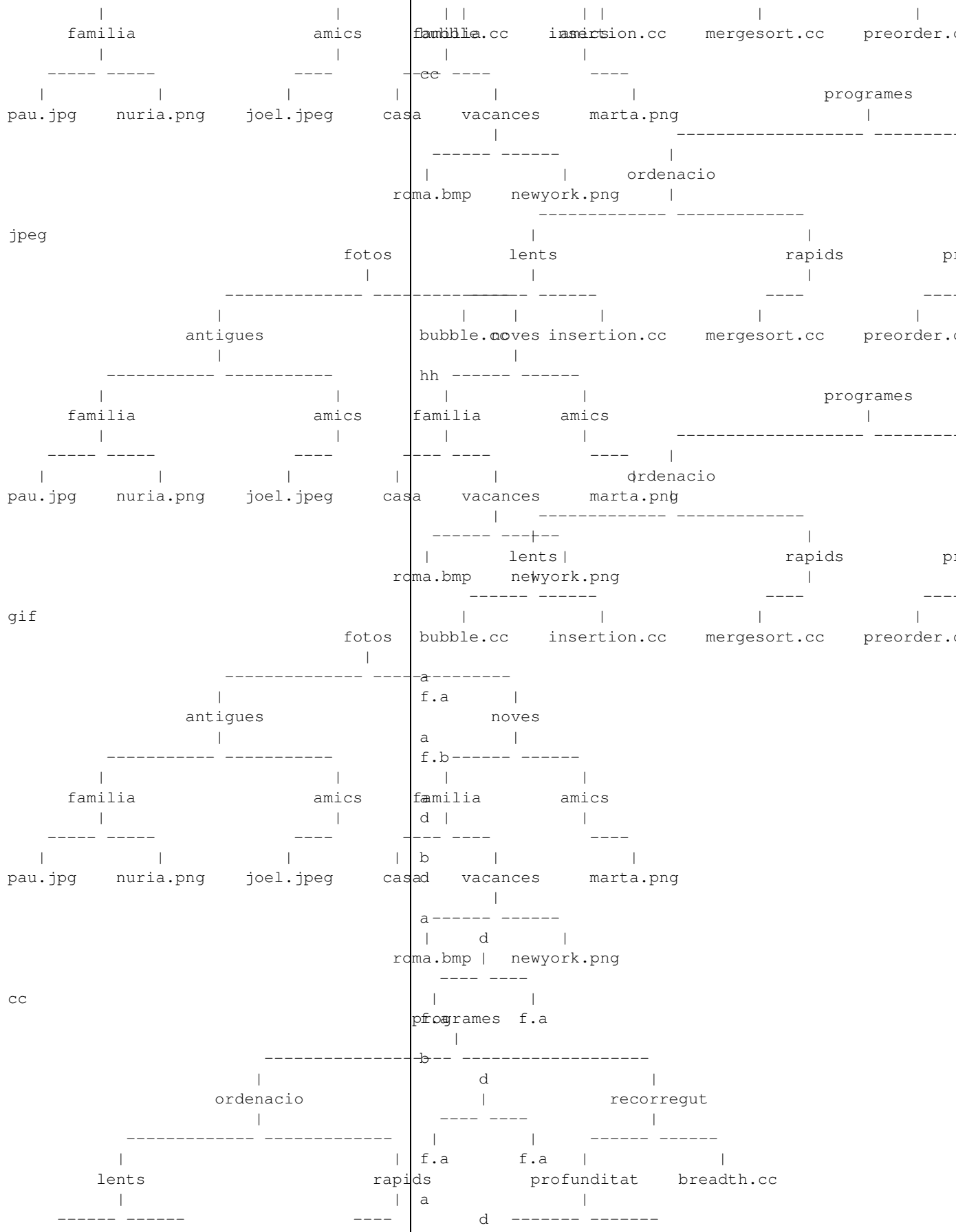
En aquest exercici, considerarem arbres binaris d'strings que representen arbres de directoris i fitxers. Observeu el següent exemple:

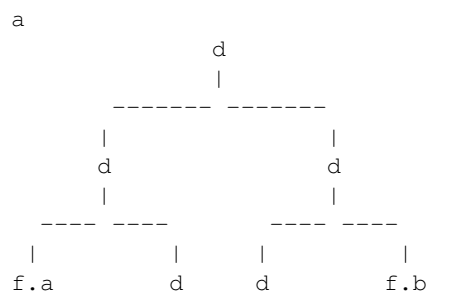
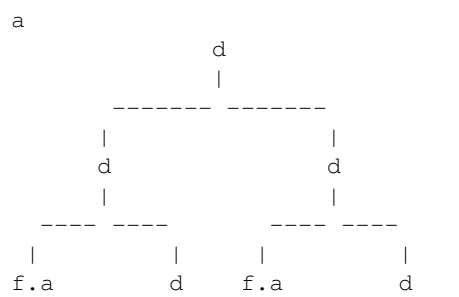
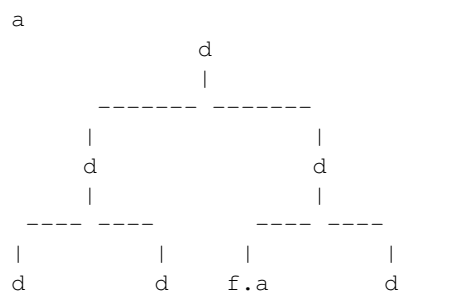
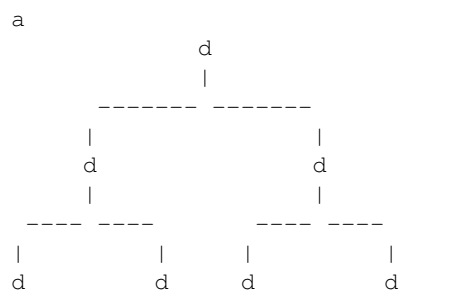
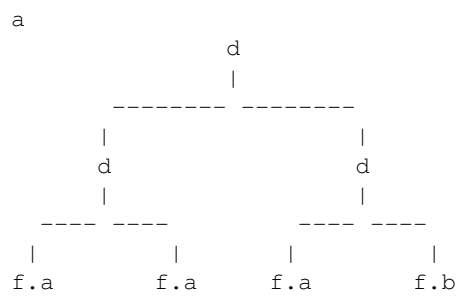
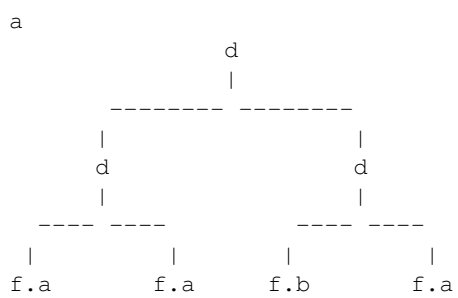
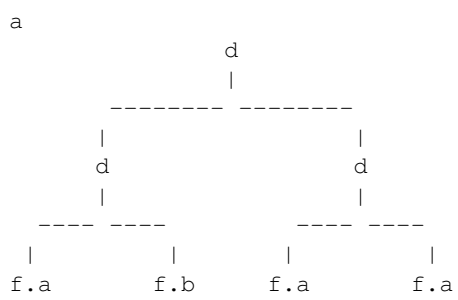
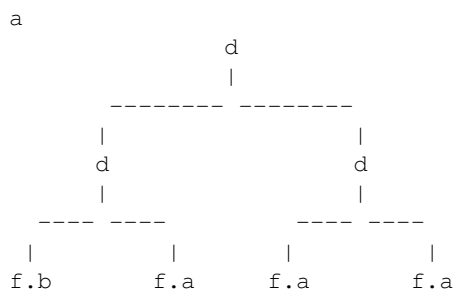
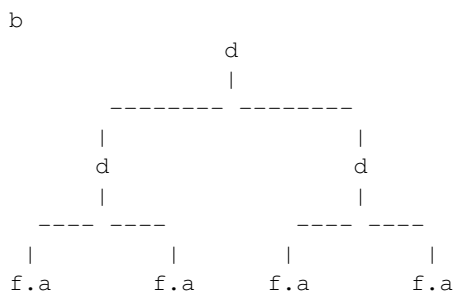
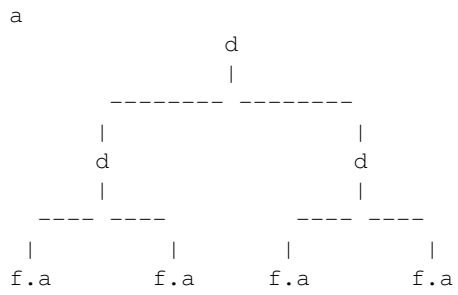
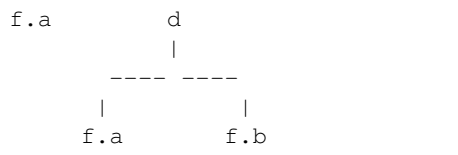


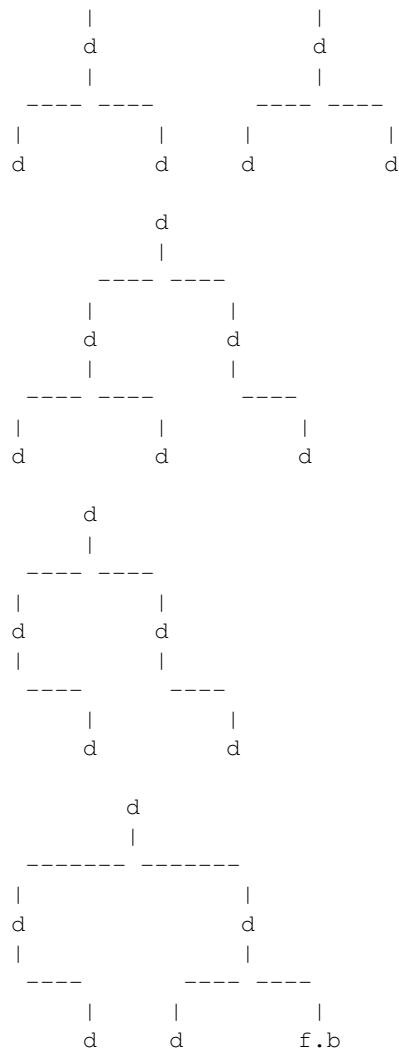
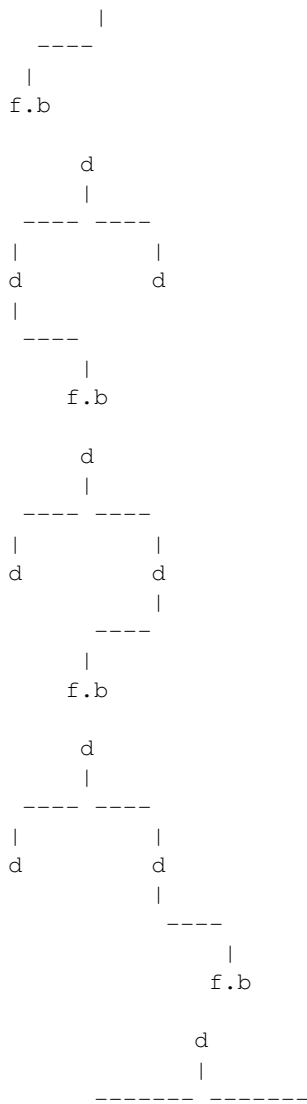
En un arbre de directoris i fitxers hi ha dos tipus de nodes: directoris i fitxers. L'string d'un directori és no buit i està format per lletres minúscules. L'string d'un fitxer està format per una primera seqüència no buida de lletres minúscules, seguida del caràcter '.', seguit d'una segona seqüència 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.

Volem executar una comanda per a esborrar tots els fitxers que penguin del directori principal i que tenen una certa extensió. Per exemple, si volguessim esborrar tots els fitxers amb extensió png (La típica comanda `rm -r *.png`), l'arbre quedaria transformat així:









Exemple d'entrada 2

INLINEFORMAT

png

fotos (antigues (familia (pau.jpg, nuria.png), amics (joel.jpeg)), noves (familia (casa, vacances (roma.bmp

bmp

fotos (antigues (familia (pau.jpg, nuria.png), amics (joel.jpeg)), noves (familia (casa, vacances (roma.bmp

jpeg

fotos (antigues (familia (pau.jpg, nuria.png), amics (joel.jpeg)), noves (familia (casa, vacances (roma.bmp

gif

fotos (antigues (familia (pau.jpg, nuria.png), amics (joel.jpeg)), noves (familia (casa, vacances (roma.bmp

cc

programes (ordenacio (lents (bubble.cc, insertion.cc), rapid (mergesort.cc)), recorregut (profunditat (p

cc

programes (ordenacio (lents (bubble.cc, insertion.cc), rapid (mergesort.cc)), recorregut (profunditat (p

hh

programes (ordenacio (lents (bubble.cc, insertion.cc), rapid (mergesort.cc)), recorregut (profunditat (p

a

f.a

a

f.b

a

```

d
b
d
a
d(f.a, f.a)
b
d(f.a, f.a)
a
d(f.a, f.b)
b
d(f.a, f.b)
a
d(d, f.b)
b
d(d, f.b)
a
d(d(f.a, f.a), f.a)
  
```



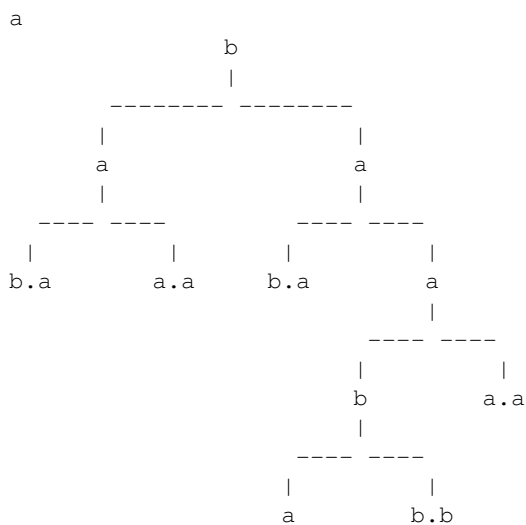
```

a
d(d(f.b, f.a), f.a)
a
d(d(f.a, f.b), f.a)
a
d(d(f.a, f.a), f.b)
a
d(f.a, d(f.a, f.a))
a
d(f.b, d(f.a, f.a))
a
d(f.a, d(f.b, f.a))
a
d(f.a, d(f.a, f.b))
a
d(d(f.a, f.a), d(f.a, f.a))
b
d(d(f.a, f.a), d(f.a, f.a))
a
d(d(f.b, f.a), d(f.a, f.a))
a
d(d(f.a, f.b), d(f.a, f.a))
a
d(d(f.a, f.a), d(f.b, f.a))
a
d(d(f.a, f.a), d(f.a, f.b))
a
d(d(d, d), d(d, d))
a
d(d(d, d), d(f.a, d))
a
d(d(f.a, d), d(f.a, d))
a
d(d(f.a, d), d(d, f.b))

```

Exemple d'entrada 3

VISUALFORMAT

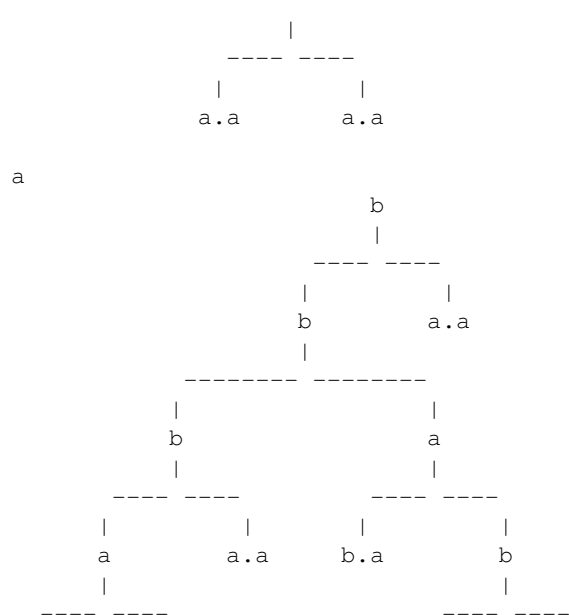


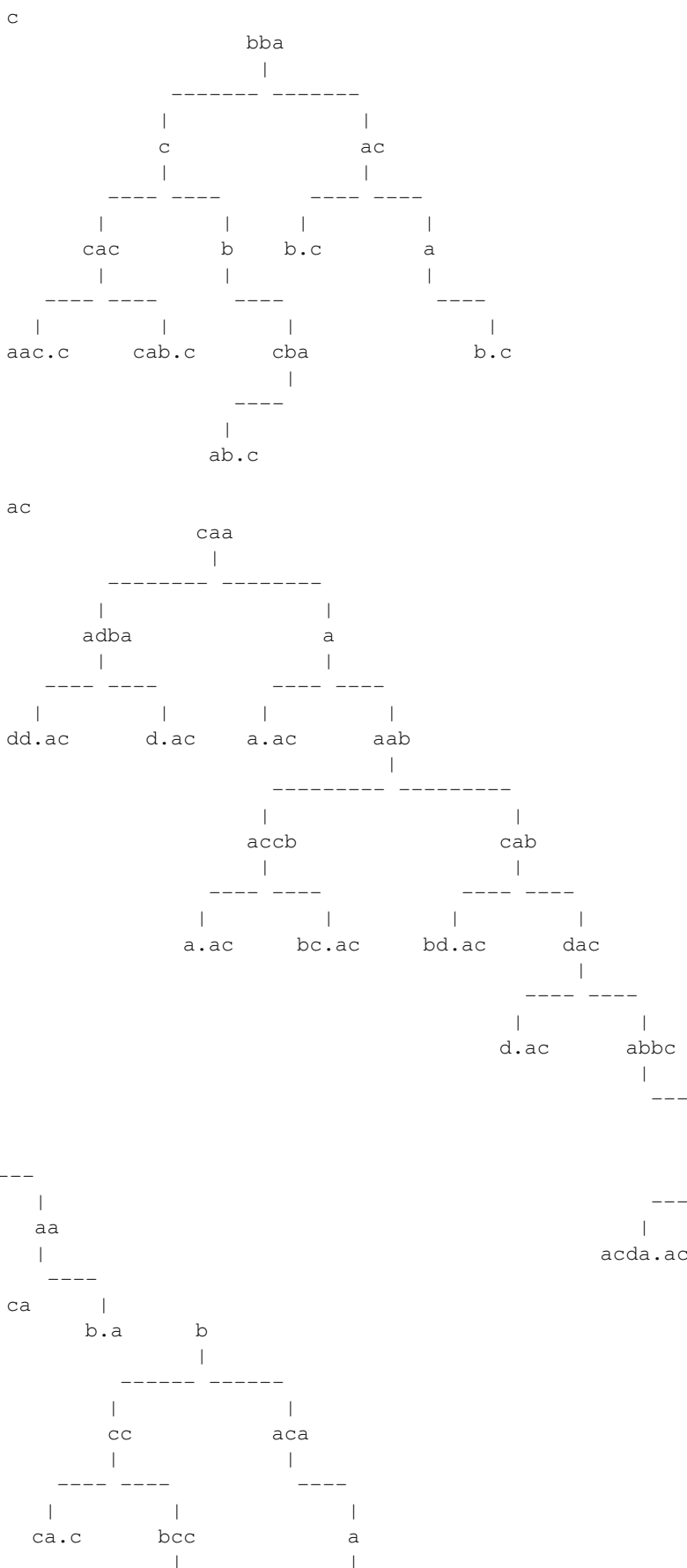
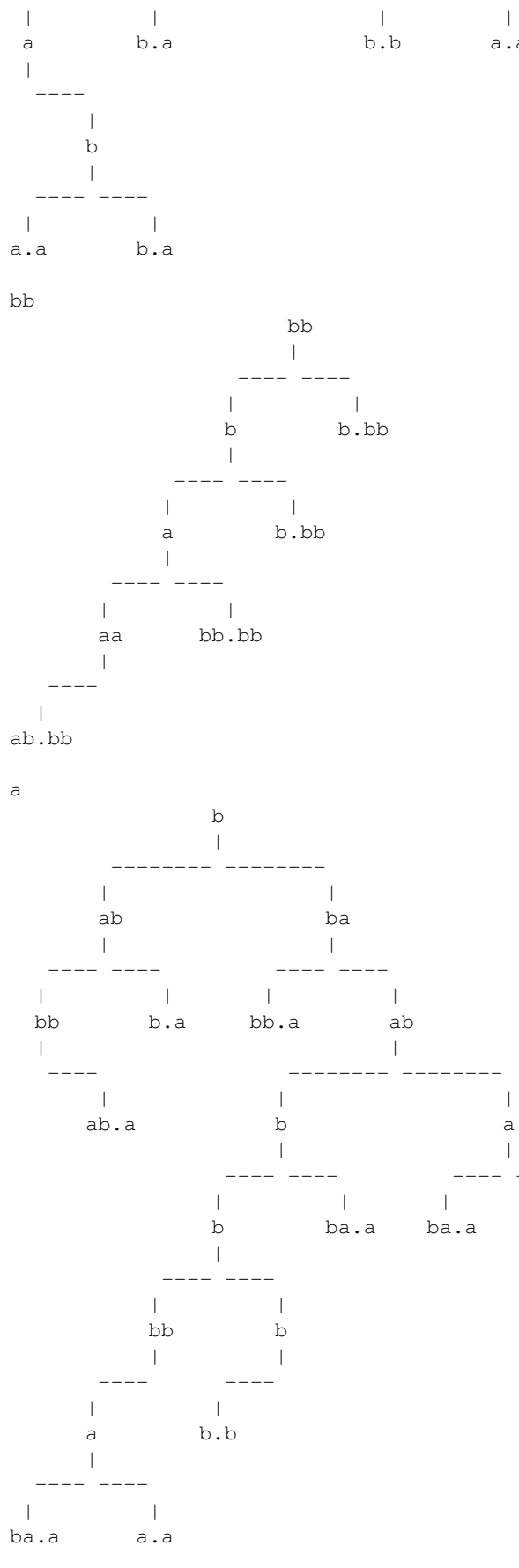
Exemple de sortida 2

```

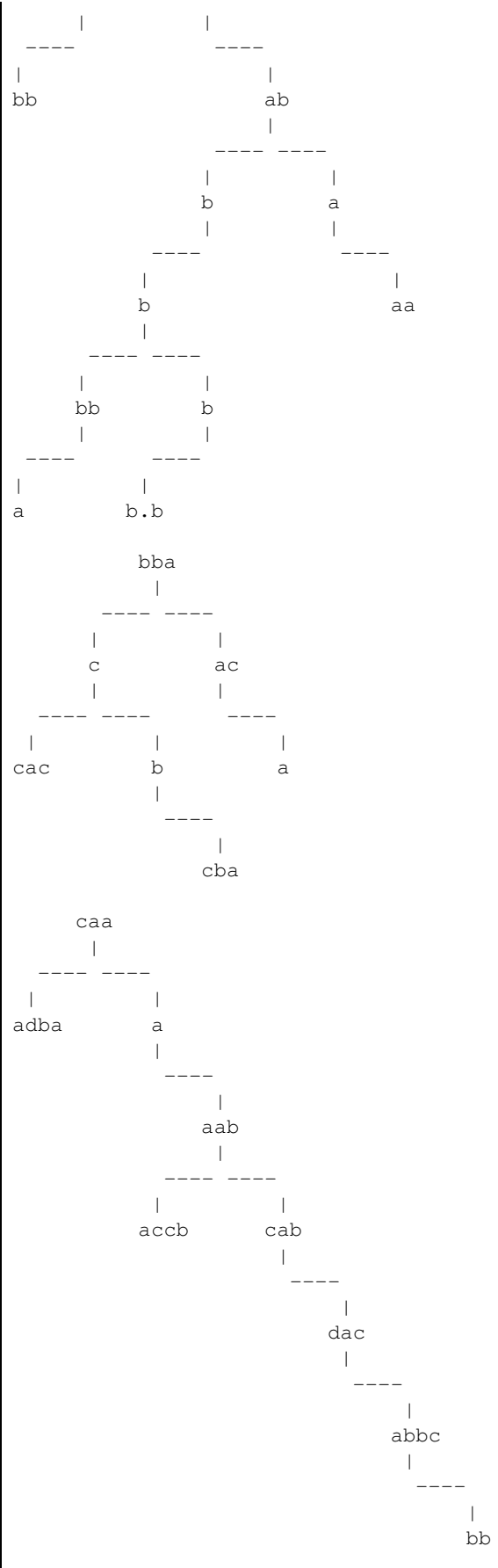
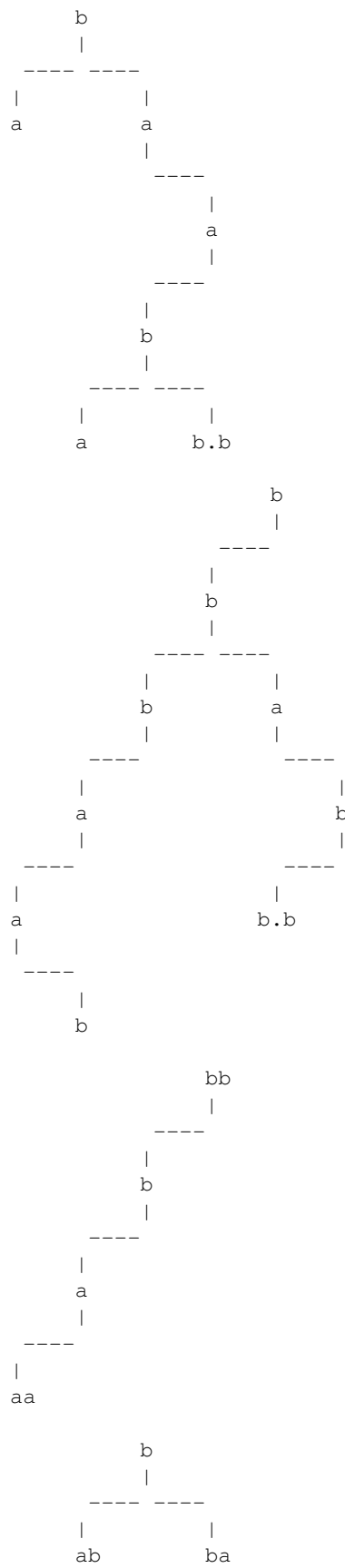
fotos (antigues (familia (pau.jpg, ) , amics (joel.jpeg, ) ) , no
fotos (antigues (familia (pau.jpg, nuria.png) , amics (joel.j
fotos (antigues (familia (pau.jpg, nuria.png) , amics) , noves
fotos (antigues (familia (pau.jpg, nuria.png) , amics (joel.j
programes (ordenacio (lents, rapids) , recorregut (profundit
programes (ordenacio (lents, rapids) , recorregut (profundit
programes (ordenacio (lents (bubble.cc, insertion.cc) , rapi
()
f.b
d
d
d
d(f.a, f.a)
d(, f.b)
d(f.a, )
d(, d)
d(f.a, d)
d(d, f.b)
d(d, )
d(d, )
d(d(f.b, ), )
d(d(, f.b), )
d(d, f.b)
d(, d)
d(f.b, d)
d(, d(f.b, ))
d(, d(, f.b))
d(d, d)
d(d(f.a, f.a), d(f.a, f.a))
d(d(f.b, ), d)
d(d(, f.b), d)
d(d, d(f.b, ))
d(d, d(, f.b))
d(d(d, d), d(d, d))
d(d(d, d), d(, d))
d(d(, d), d(, d))
d(d(, d), d(d, f.b))

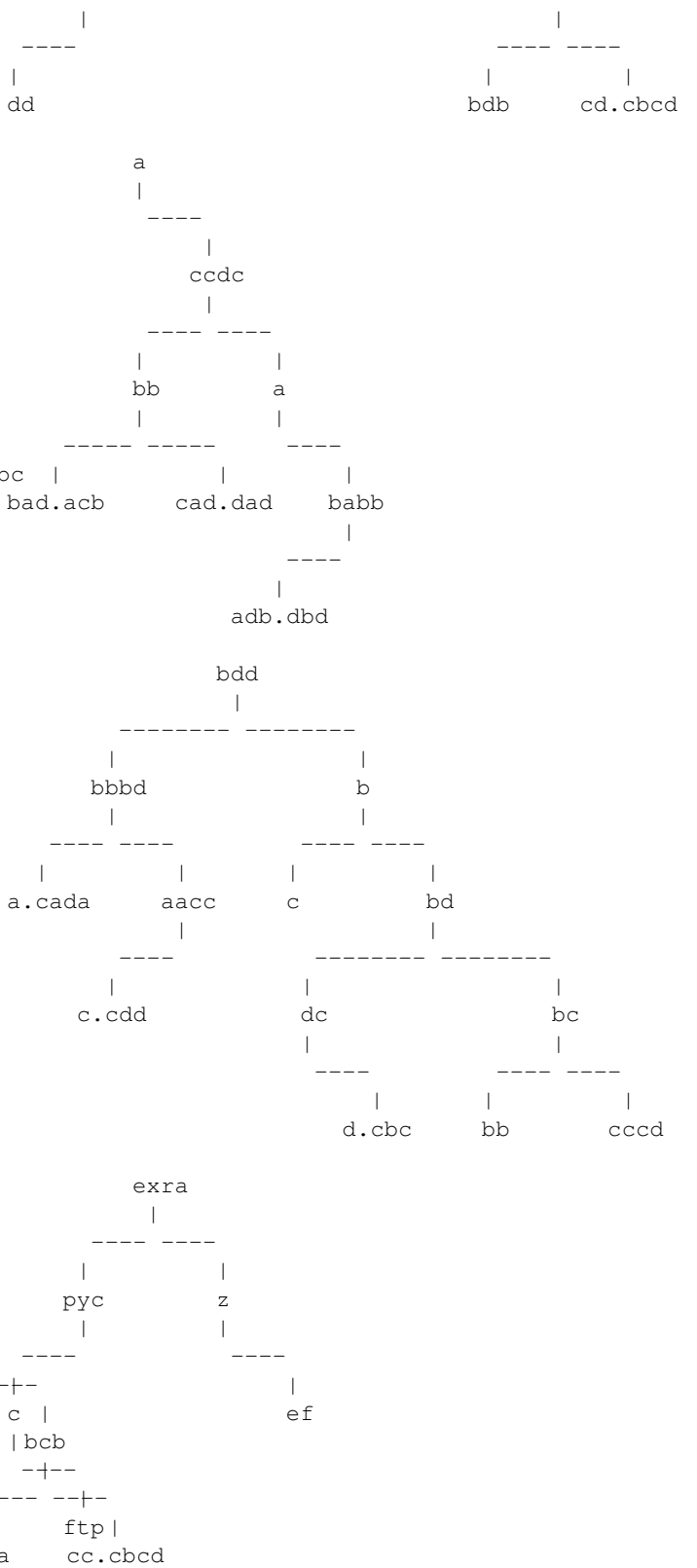
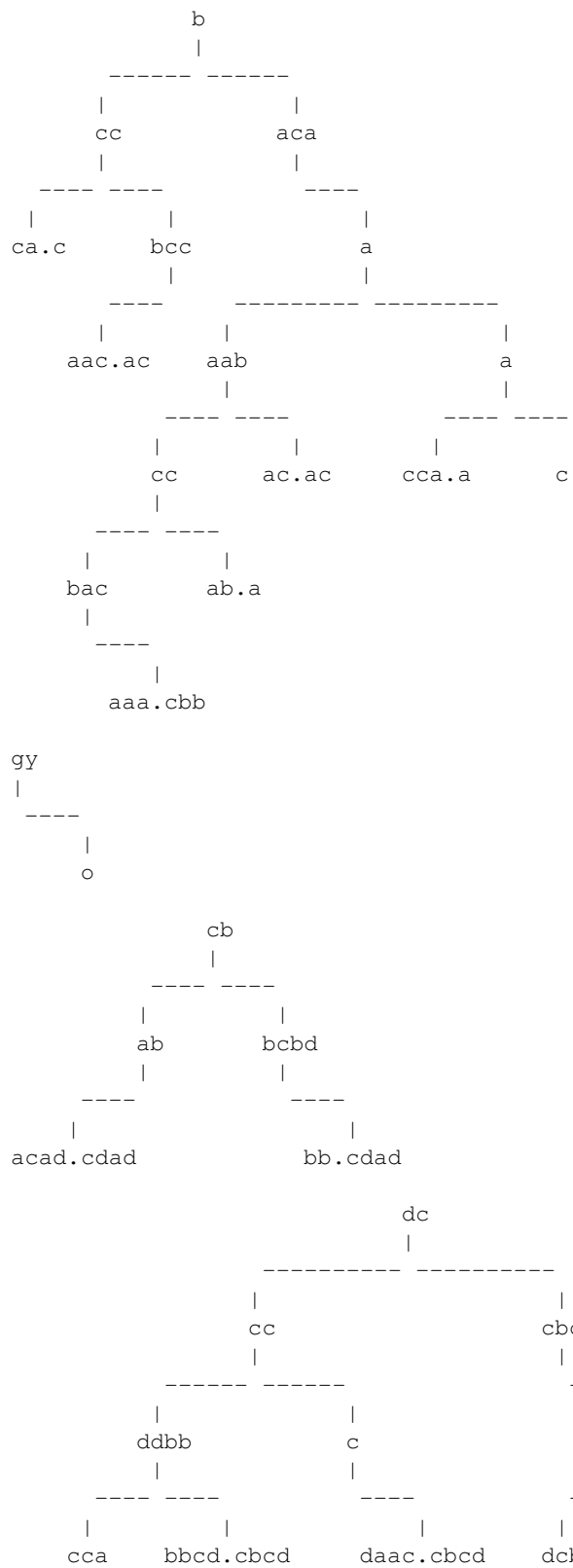
```





Exemple de sortida 3





Exemple d'entrada 4

INLINEFORMAT

a
 b(a(b.a,a.a),a(b.a,a(b(a(a.a,a.a),b.b),a.a)))

a
 b(b(b(a(a(,b(a.a,b.a)),b.a),a.a),a(b.a,b(b.b,a.a))),a.
 bb
 bb(b(a(aa(ab.bb,)),bb.bb),b.bb),b.bb)

a

b(ab(bb(, ab.a), b.a), ba(bb.a, ab(b(bb(a(ba.a, a.a), b(b.b,)), ba.a), a(ba.a, aa(, b.a))))

c

Exemple de sortida 4

b(a, a(, a(b(a, b.b),)))

bba(c(cac(aac.c, cab.c), b(, cba(ab.c,))), ac(b.c, a(, b.c),),), a(b(b(a(a, b)),),),

ac

bb(b(a(aa,)),),

caa(adba(dd.ac, d.ac), a(a.ac, aab(accb(a.ac, bc.ac), cab(bd.ac, dac(d.ac, abc(b(acba.ac, a))))))

ca

bba(c(cac, b(, cba)), ac(, a))

b(cc(ca.c, bcc(aac.ac,)), aca(, a(aab(cc(bac(cc.ca, aac.ccb), ab.a), ac.ac), a(cca.a, c.bc))))

by

b(cc(ca.c, bcc(aac.ac,)), aca(, a(aab(cc(bac(, aaa.cbb), ab

gy(, o(, iq.by))

gy(, o)

cc

cb(ab(acad.cdad,), bcbd(, bb.cdad))

cb(ab(acad.cdad, cd.cc), bcbd(aab.cc, bb.cdad)) dg(cc(dddb(cca(dd,), bbcd.cbcd), c(, daac.cbcd), cbc(, bcb

bd

a(, ccdc(bb(bad.acb, cad.dad), a(, babb(adb.dbd,)))

dc(cc(dddb(cca(dd(cc.bd, bdcc.bd),), bbcd.cbcd), c(ba.bd, daac.cbcd), cbc(a.bd, bcb(dca(bcb(, bc(bb, extra(pyc(c(, ftp),), z(, ef))

bdb

extra(pyc(c(, ftp),), z(, ef))

a(, ccdc(bb(bad.acb, cad.dad), a(, babb(adb.dbd, badc.bdb)))

bcaa

bdd(bbbd(a.cada, aacc(c.cdd, dba.bcaa)), b(d(bd.bcaa, baa.bcaa), bd(dc(, d.cbc), bc(bb(c.bcaa,), cccd(dd

r

extra(pyc(c(rre.r, ftp(, vu.r)), wmxn.r), z(u,r, ef(oii.r, k.r)))

Observació

La vostra funció i subfuncions que creu 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.

Informació del problema

Autor : PRO2

Generació : 2023-10-21 13:54:01

© Jutge.org, 2006–2023.

<https://jutge.org>