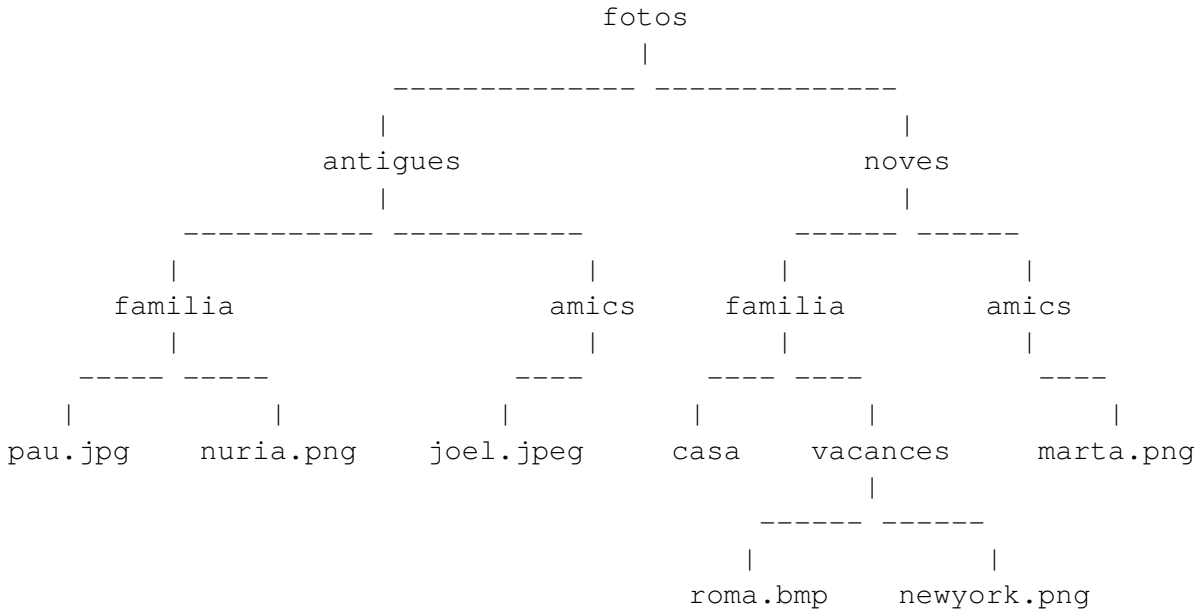
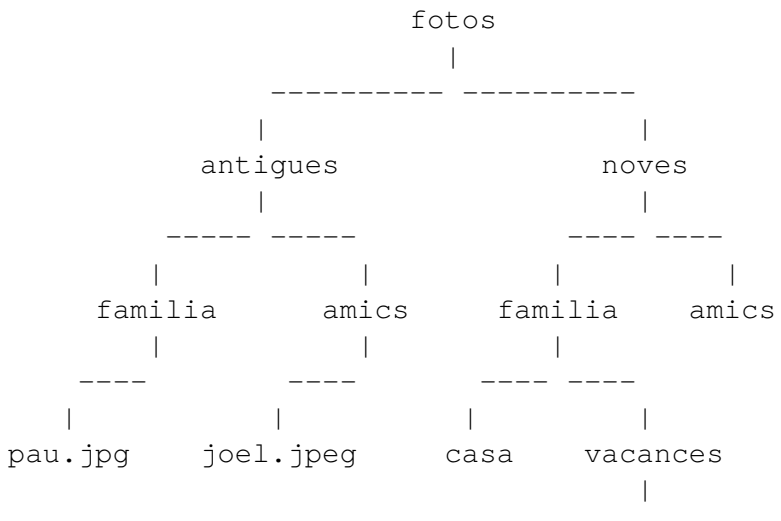
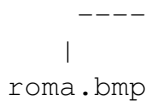
Esborra fitxers amb una extensió X24232_ca

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 pengen 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:





Heu d'implementar una funció `RECURSIVA` que rep un arbre de directoris i fitxers, i també rep un string que representa una extensió, i retorna un nou arbre de directoris i fitxers que és el resultat d'esborrar tots els fitxers amb aquella extensió de l'arbre rebut. 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 l'arbre resultant d'esborrar de t tots els nodes amb strings d
//       la forma "s.e" a on 'e' és igual a 'extension'.
BinTree<string> removeAll(const string &extension, const BinTree<string> &t);
```

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

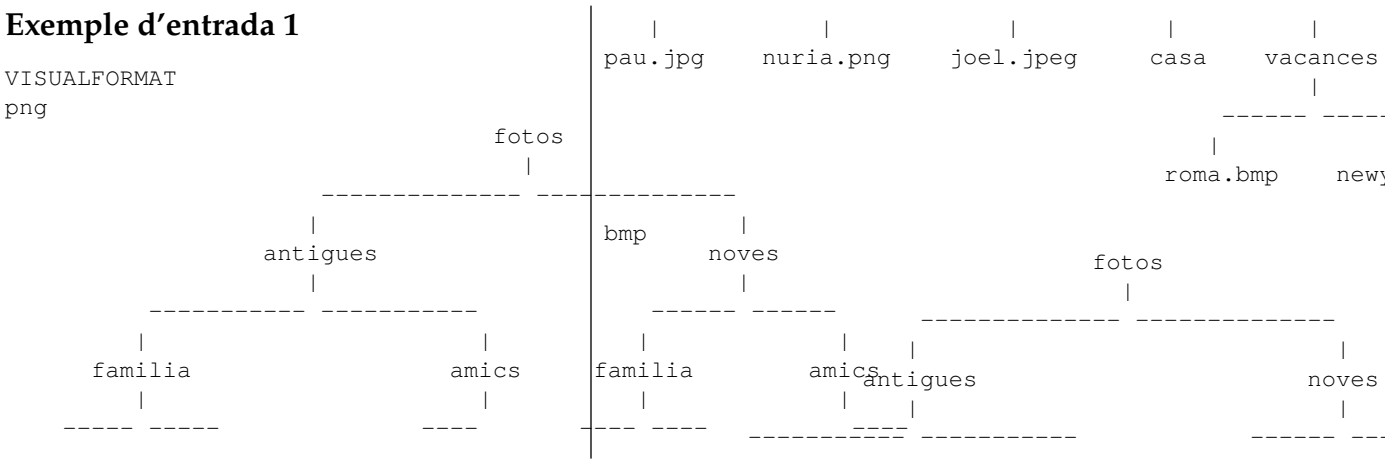
Entrada

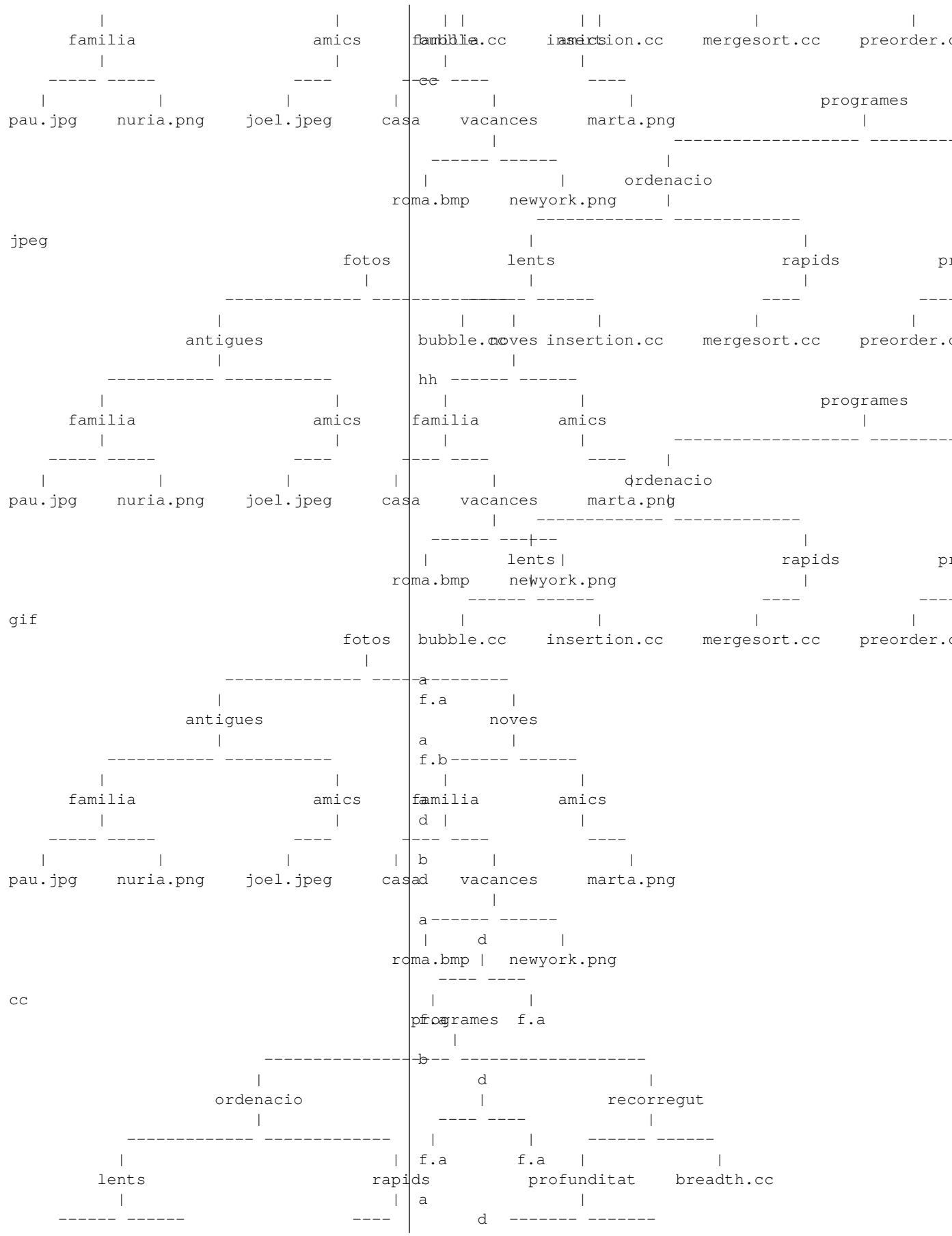
La primera linia de l'entrada descriu el format en el que es descriuen els arbres, o bé INLINEFORMAT o bé VISUALFORMAT. Després venen un nombre arbitrari de casos. Cada cas té una primera linia amb la extensió, un string no buit de lletre minúscules, i després una descripció d'un arbre binari que representa una expressió. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquesta entrada. Només cal que implementeu la funció abans esmentada.

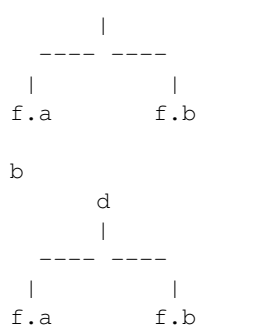
Sortida

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

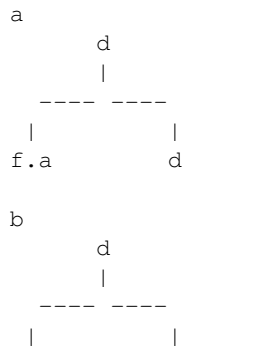
Exemple d'entrada 1

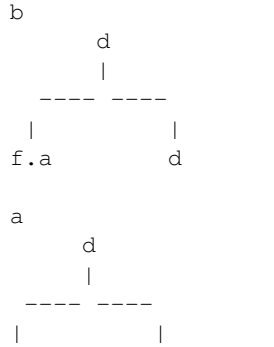


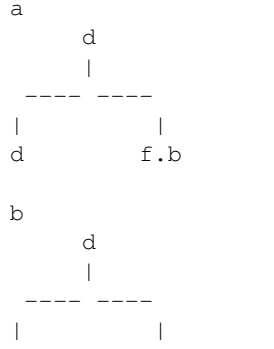

familia
amics
pau.jpg
nuria.png
joel.jpeg
casa
vacances
marta.png
programes
ordenacio
roma.bmp
newyork.png
jpeg
fotos
lents
rapids
antigues
bubble.cc
noves
insertion.cc
mergesort.cc
preorder.c
hh
gif
cc
pr
ordenacio
recorregut
profunditat
breadth.cc

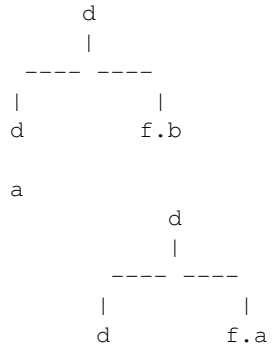

f.a
f.b

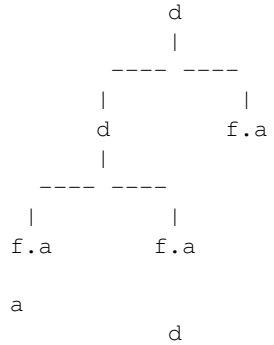
d
f.a
f.b

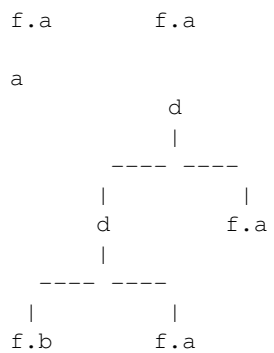

d
f.a
d

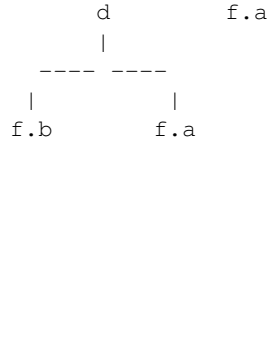

d
f.a
d

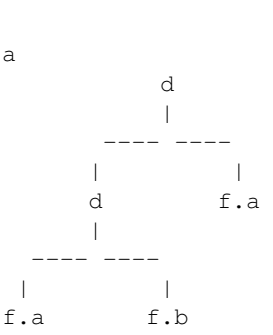

d
d
f.b


d
d
f.b

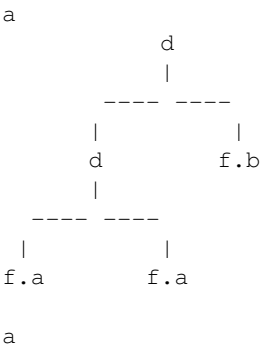

d
d
f.a
f.a

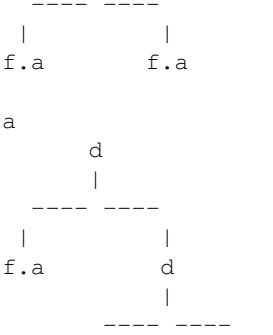

d
d
f.a
f.a

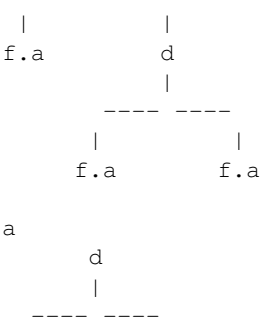

d
d
f.b
f.a

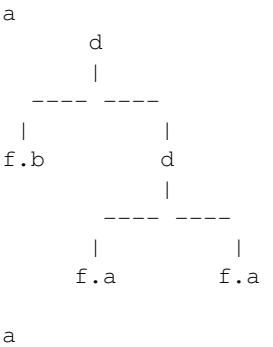

d
d
f.a
f.b

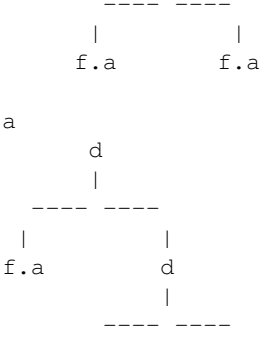
d
d
f.b
f.a

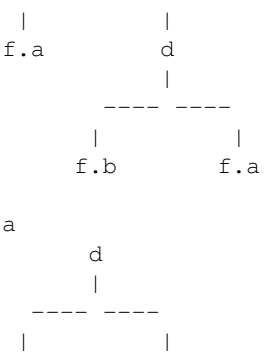

d
f.a
d
f.a
f.a

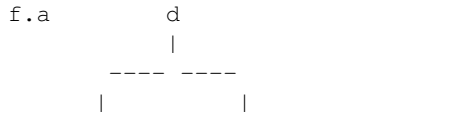

d
f.b
d
f.a
f.a

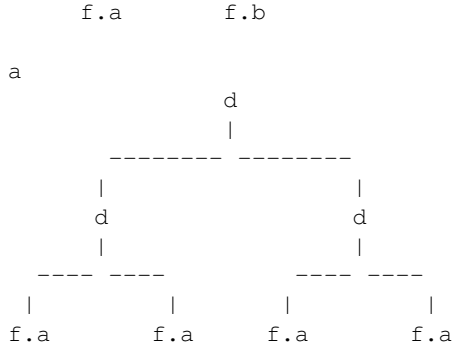

d
f.a
d
f.b
f.a

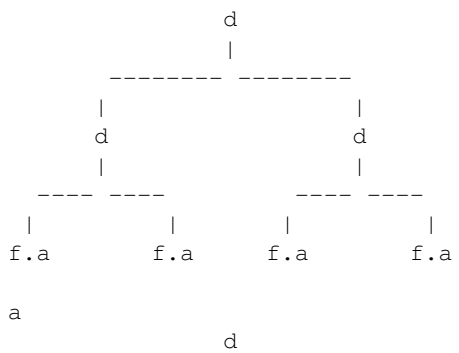

d
f.a
d
f.a
f.b

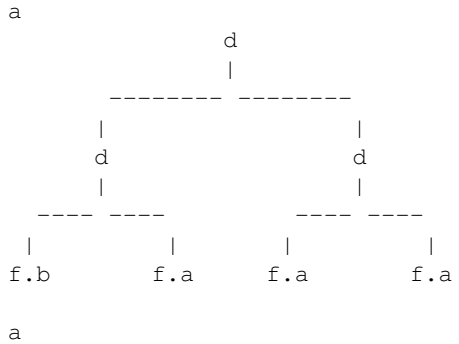

d
f.a
d
f.b
f.a

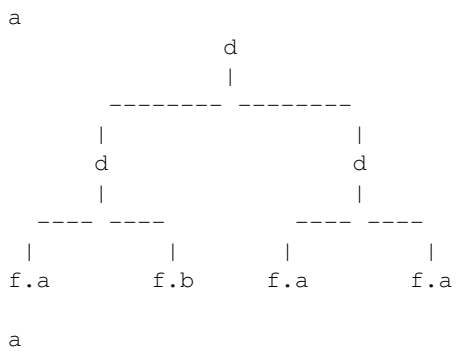

d

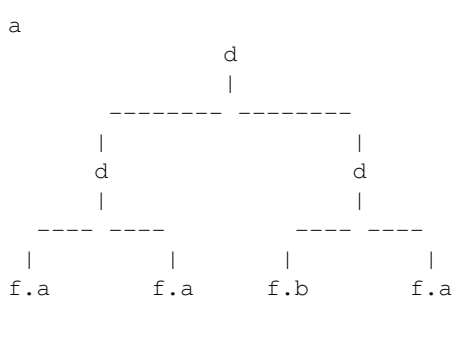

f.a
d
f.a
f.b

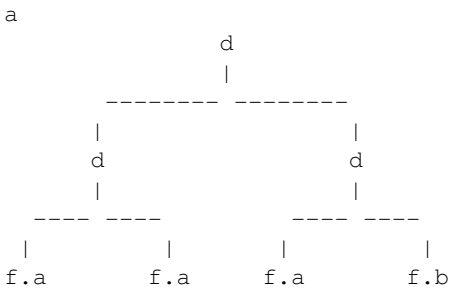

a
d
d
d
f.a
f.a
f.a
f.a

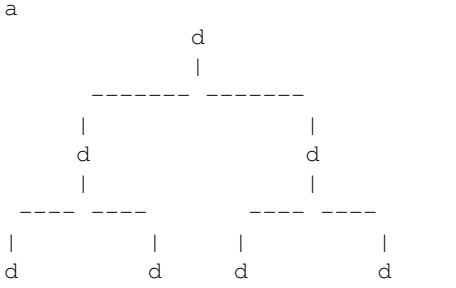

b
d
d
d
f.a
f.a
f.a
f.a

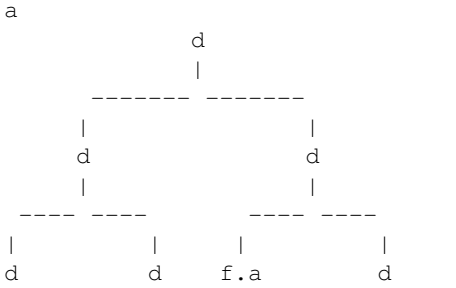

a
d
d
d
f.b
f.a
f.a
f.a

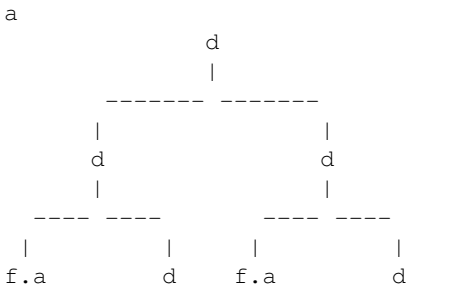

a
d
d
d
f.a
f.b
f.a
f.a

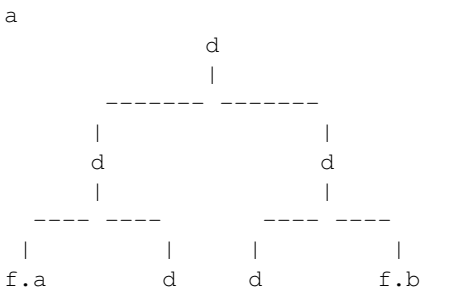

a
d
d
d
f.a
f.a
f.b
f.a


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

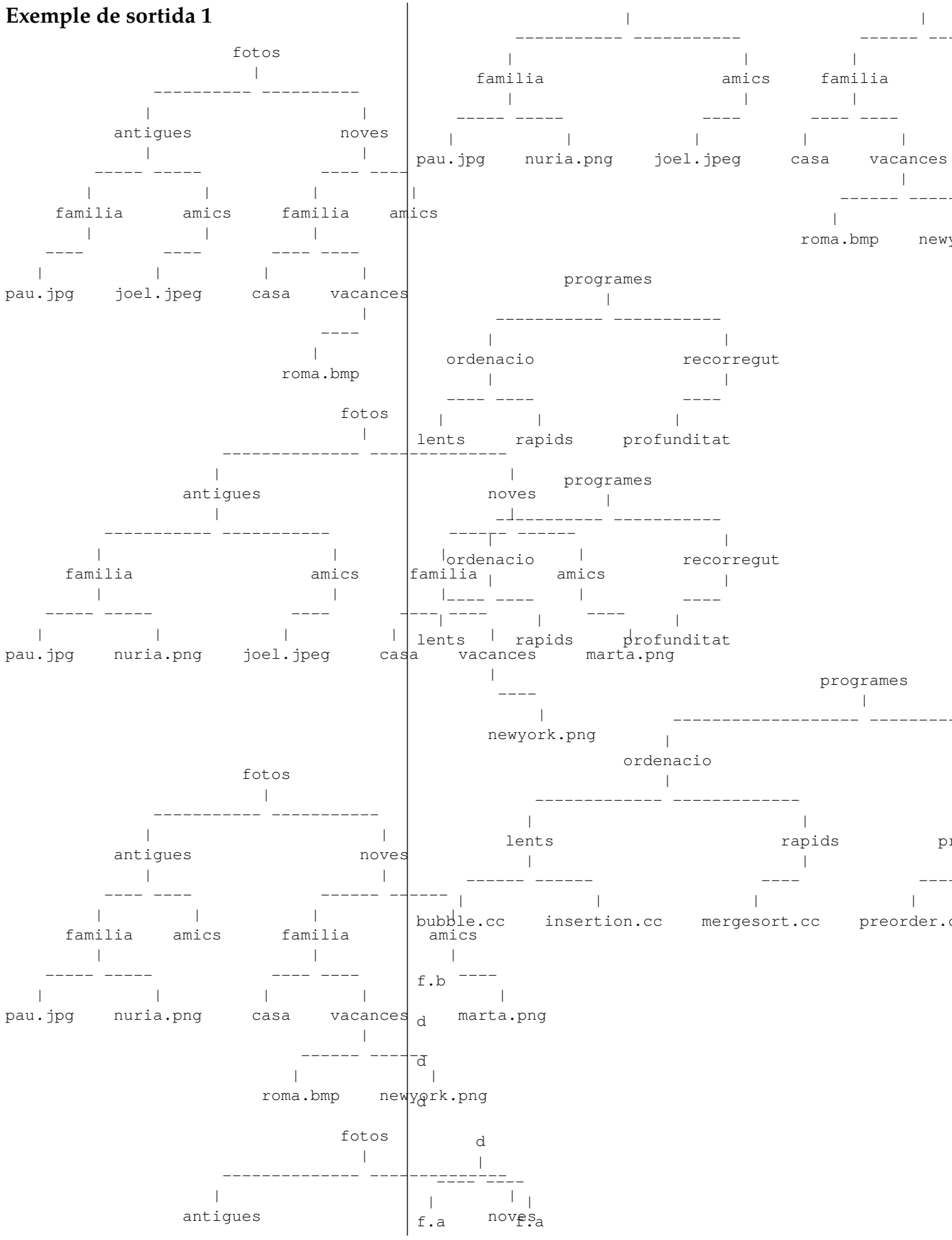

a
d
d
d
d
d
d
d
d


a
d
d
d
d
d
d
f.a
d


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

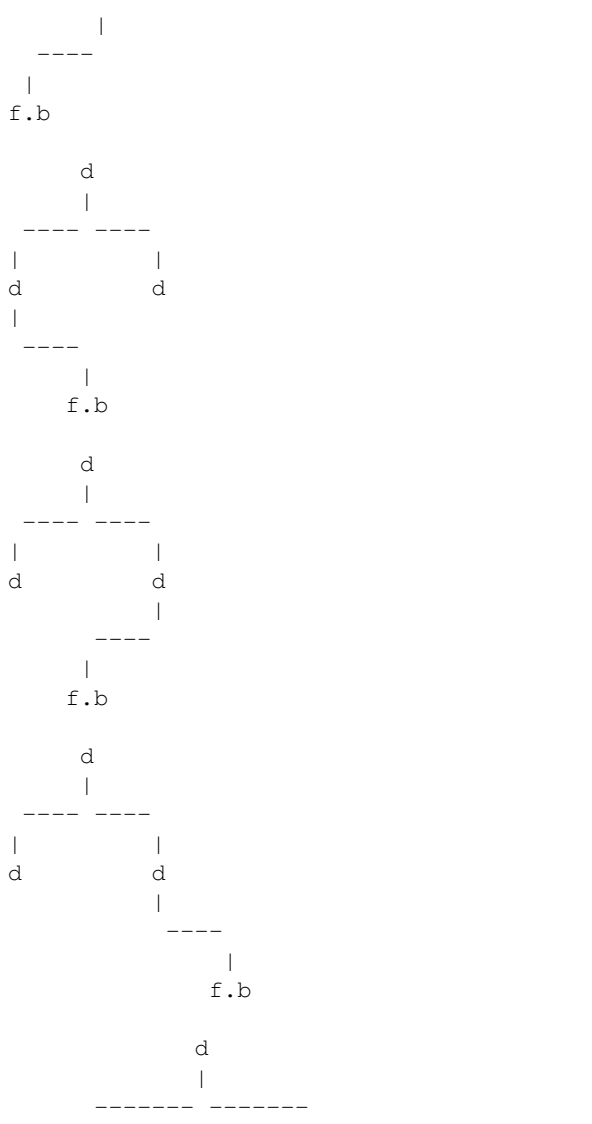

a
d
d
d
f.a
d
d
f.b

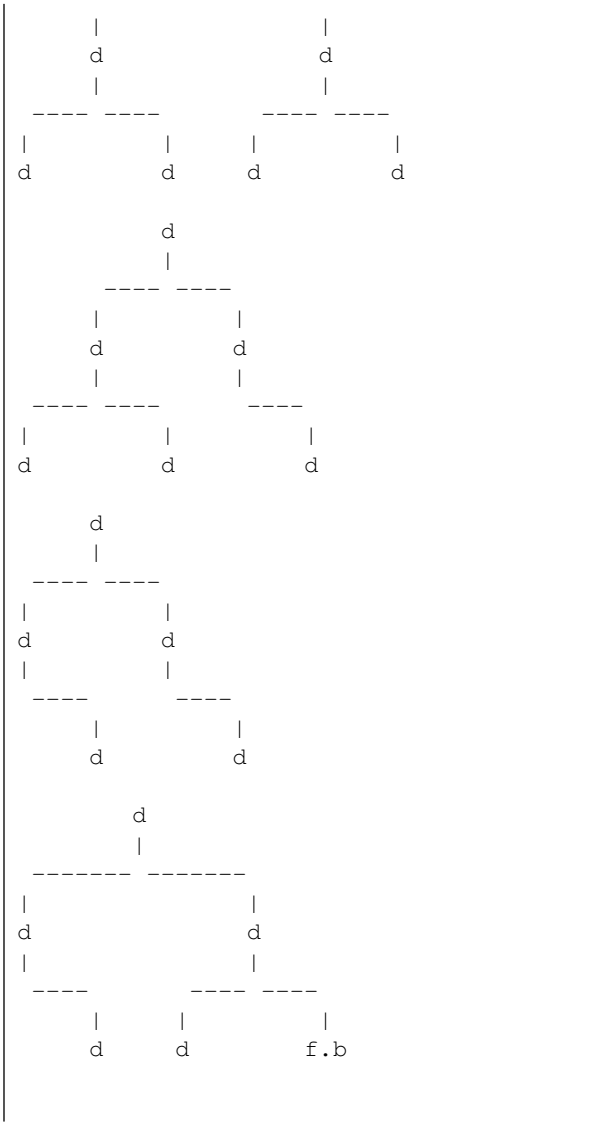
Exemple de sortida 1



```
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                 d
      |                 |
  ---- ----         ---- ----
 |         |       |         |
f.a       f.a     f.a       f.a

           d
           |
       ---- ----
      |         |
      d         d
```





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),rapids(mergesort.cc)),recorregut(profunditat(p
cc
programes(ordenacio(lents(bubble.cc,insertion.cc),rapids(mergesort.cc)),recorregut(profunditat(p
hh
programes(ordenacio(lents(bubble.cc,insertion.cc),rapids(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(f.a,d)
b
d(f.a,d)
a
d(d,f.b)
b
d(d,f.b)
a
d(d(f.a,f.a),f.a)
```

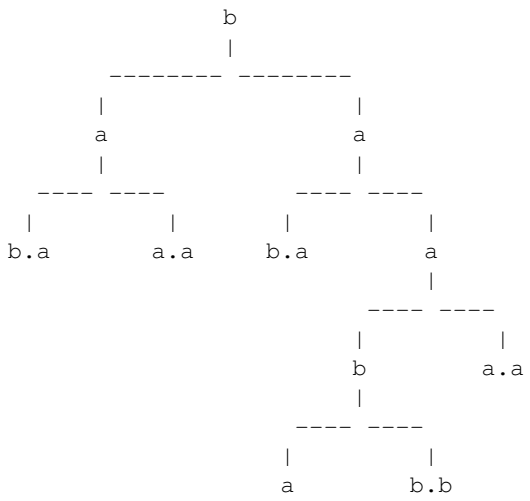
```
a
d(d(f.b,f.a),f.a)
a
d(d(f.a,f.b),f.a)
a
d(d(f.a,f.a),f.b)
a
d(f.a,d(f.a,f.a))
a
d(f.b,d(f.a,f.a))
a
d(f.a,d(f.b,f.a))
a
d(f.a,d(f.a,f.b))
a
d(d(f.a,f.a),d(f.a,f.a))
b
d(d(f.a,f.a),d(f.a,f.a))
a
d(d(f.b,f.a),d(f.a,f.a))
a
d(d(f.a,f.b),d(f.a,f.a))
a
d(d(f.a,f.a),d(f.b,f.a))
a
d(d(f.a,f.a),d(f.a,f.b))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

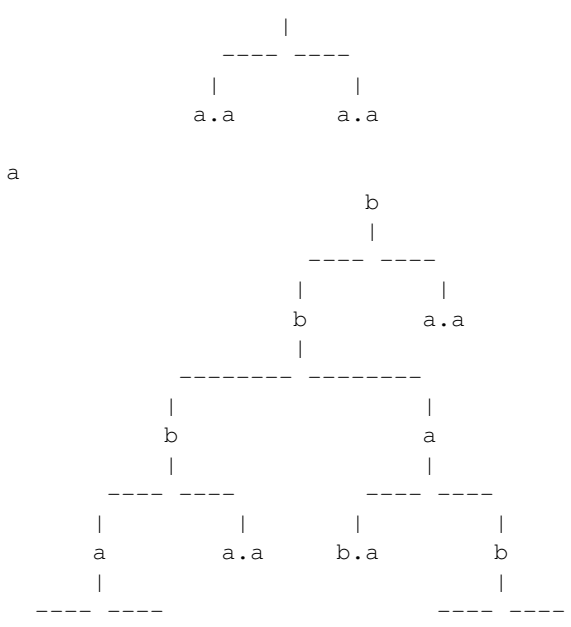
Exemple de sortida 2

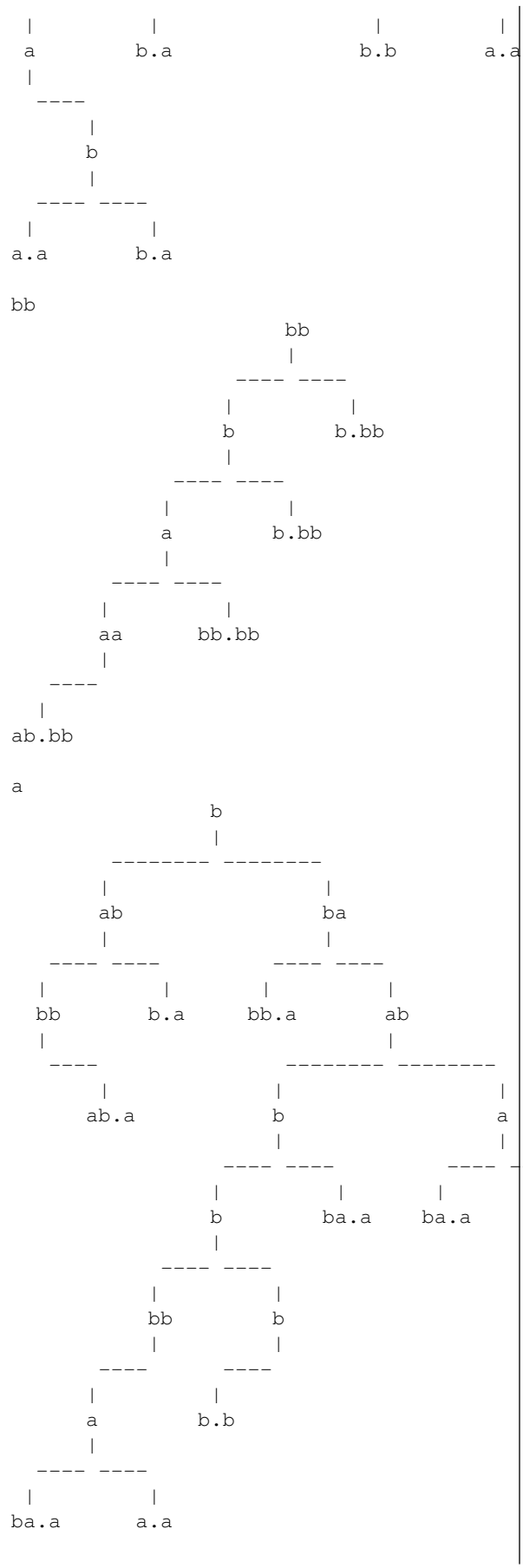
```
fotos(antigues(familia(pau.jpg,),amics(joel.jpeg,)),nov
fotos(antigues(familia(pau.jpg,nuria.png),amics(joel.jp
fotos(antigues(familia(pau.jpg,nuria.png),amics),noves
fotos(antigues(familia(pau.jpg,nuria.png),amics(joel.jp
programes(ordenacio(lents,rapids),recorregut(profundita
programes(ordenacio(lents,rapids),recorregut(profundita
programes(ordenacio(lents(bubble.cc,insertion.cc),rapid
()
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))
```

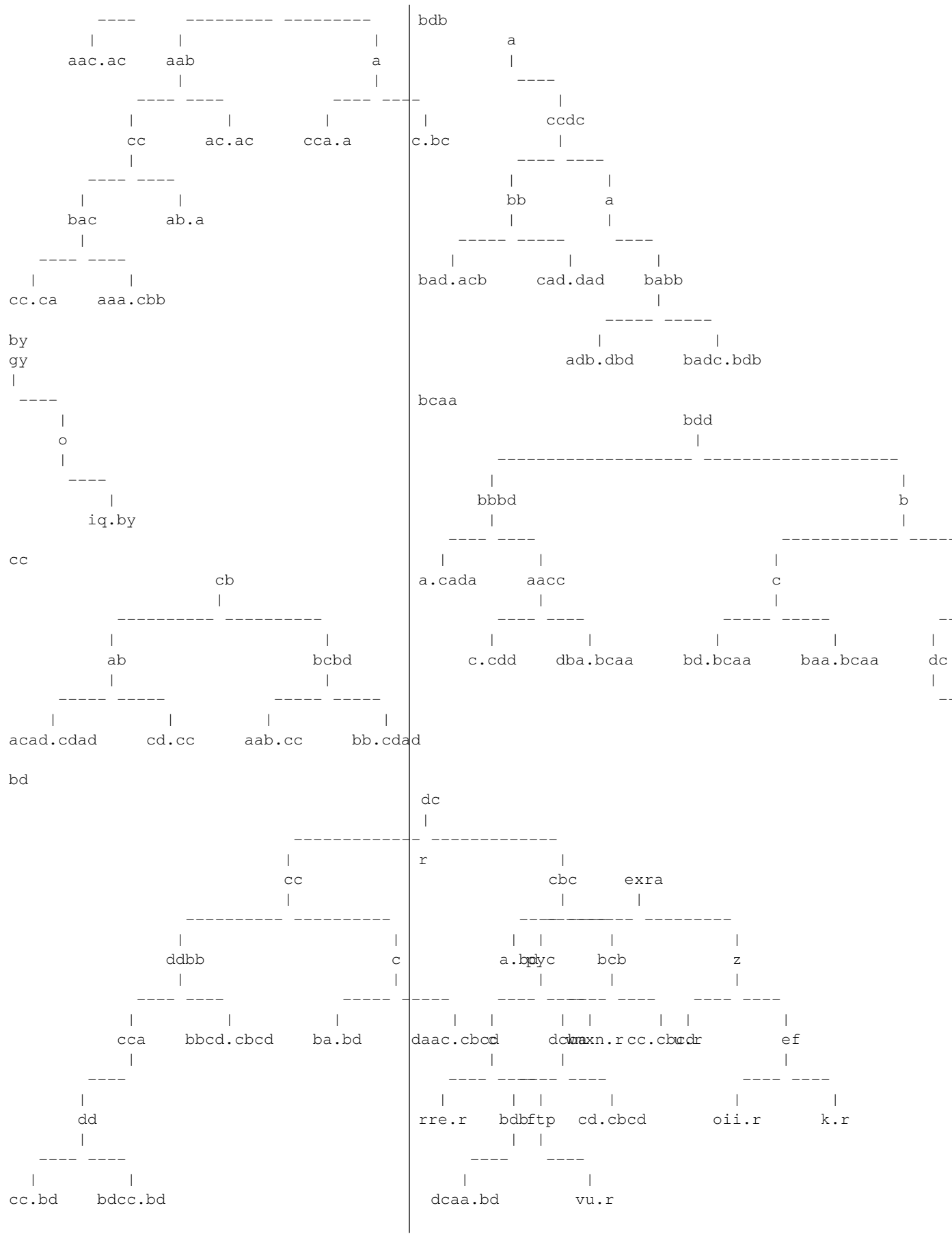
Exemple d'entrada 3

```
VISUALFORMAT
a
```

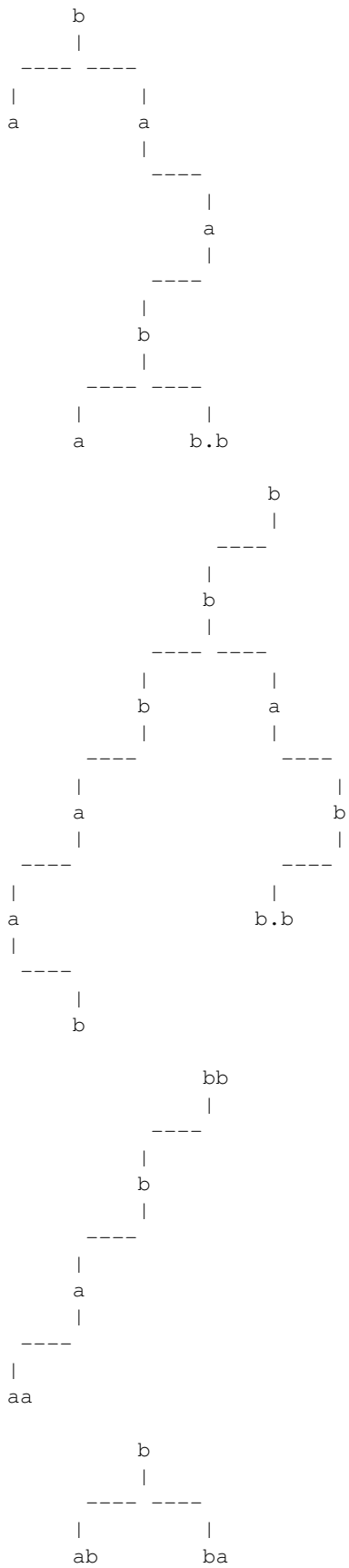


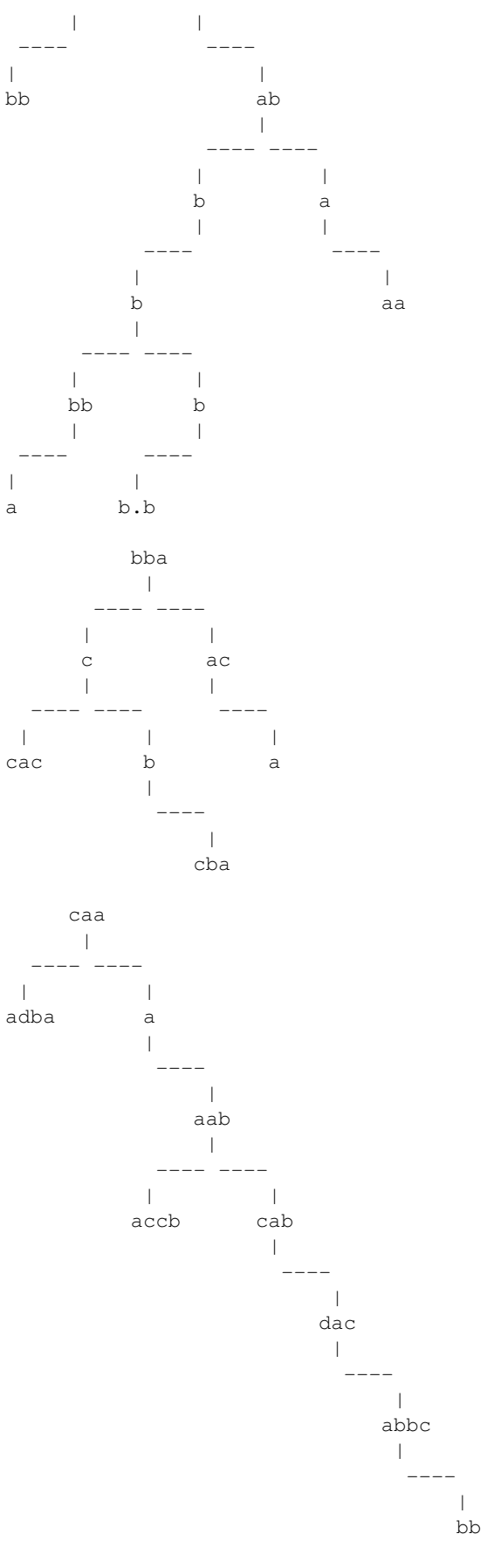


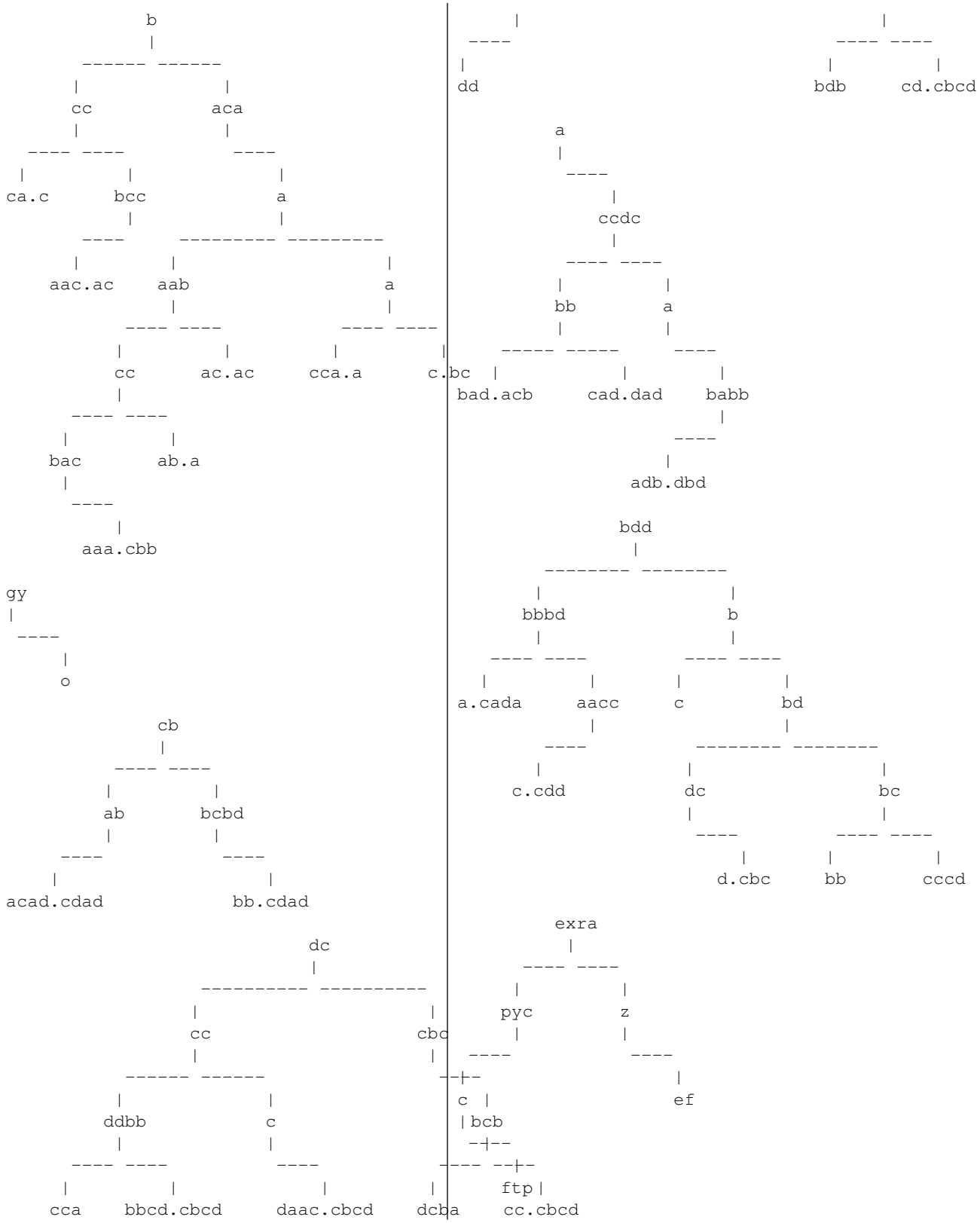



aac.ac
aab
a
cc
ac.ac
cca.a
bac
ab.a
cc.ca
aaa.cbb
by
gy
iq.by
cb
ab
bcbd
acad.cdad
cd.cc
aab.cc
bb.cdad
bd
cc.bd
bdcc.bd
bdb
a
ccdc
c.bc
bb
a
bad.acb
cad.dad
babb
adb.dbd
badc.bdb
bcaa
bdd
bbbd
b
a.cada
aacc
c
c.cdd
dba.bcaa
bd.bcaa
baa.bcaa
dc
dc
r
cbc
exra
z
cca
bbcd.cbcd
ba.bd
ef
dd
rre.r
cd.cbcd
oii.r
k.r
dcaa.bd
vu.r

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.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(b(bb(a(ba.a,a.a),),b(b.b,)),ba.a),a(ba.a,aa(,b.a)))))
c
bba(c(cac(aac.c,cab.c),b(,cba(ab.c,))),ac(b.c,a(,b.c)))
ac
caa(adba(dd.ac,d.ac),a(a.ac,aab(accb(a.ac,bc.ac),cab(bd.ac,dac(d.ac,abbc(,bb(acda.ac,)))))))
ca
b(cc(ca.c,bcc(aac.ac,)),aca(,a(aab(cc(bac(cc.ca,aaa.cbb),ab.a),ac.ac),a(cca.a,c.bc))))
by
gy(,o(,iq.by))
cc
cb(ab(acad.cdad,cd.cc),bcbd(aab.cc,bb.cdad))
bd
dc(cc(ddbb(cca(dd(cc.bd,bdcc.bd),),bbcd.cbcd),c(ba.bd,daac.cbcd)),cbc(a.bd,bcb(dcba(bdb(dcaa.bd,
bdb
a(,ccdc(bb(bad.acb,cad.dad),a(,babb(adb.dbd,badc.bdb))))
bcaa
bdd(bbbd(a.cada,aacc(c.cdd,dba.bcaa)),b(c(bd.bcaa,baa.bcaa),bd(dc(,d.cbc),bc(bb(c.bcaa,),cccd(dd
r
exra(pyc(c(rre.r,ftp(,vu.r)),wmxn.r),z(u.r,ef(oii.r,k.r)))
```

Exemple de sortida 4

```
b(a,a(,a(b(a,b.b),)))
b(b(b(a(a(,b),),),a(,b(b.b,))),)
bb(b(a(aa,),),)
b(ab(bb(,),ba(,ab(b(b(bb(a,),b(b,b)),),),a(,aa))))
bba(c(cac,b(,cba)),ac(,a))
caa(adba,a(,aab(accb,cab(,dac(,abbc(,bb)))))
b(cc(ca.c,bcc(aac.ac,)),aca(,a(aab(cc(bac(,aaa.cbb),ab
gy(,o)
cb(ab(acad.cdad,),bcbd(,bb.cdad))
dc(cc(ddbb(cca(dd,),bbcd.cbcd),c(,daac.cbcd)),cbc(,bcb
a(,ccdc(bb(bad.acb,cad.dad),a(,babb(adb.dbd,))))
bdd(bbbd(a.cada,aacc(c,cdd,)),b(c,bd(dc(,d.cbc),bc(bb,
exra(pyc(c(,ftp),),z(,ef))
```

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.

Informació del problema

Autoria: PRO2

Generació: 2026-01-25T14:30:59.705Z