

Fractions Calculator

The input consists of exactly 1000 test cases in the following recursive format:

Input

```
case i [i-th test]
< list_eq >:=< eq >^\n[< list_eq >]
< eq >:=< var >=< onp >,< fraction >
< onp >:=(< var >,< onp >)(< var >,< onp >)< op >
< var >:=< (a,c,g,t) [< var >]
< op >:=(< + >,< * >,< / >)
< fraction >:=< number >/< number >
< number >:=< (1-9) [< number >]
```

< id >:= the definition of the expression

< id > on the right side: just use the definition of the expression

(*x*,*y*) choose exactly one from the list: *x* or *y*

[*x*] process (choose) *x* or not

The definition of every variable (on the left side) appears only once and follows its last appearance on the right side.

Output

The output should contain the list of variables in nondecreasing lexicographic order of identifiers and all values should be represented as simple fractions in lowest terms, i.e. in the form N / D , where N and D are relatively prime.

Score

The score is equal to the number of correctly solved test cases divided by 100.

Example

Input

```
case 1
c=_g_a/_cg_g/*
cg=_a_ct_a++
g=_a_a/_ct*
ct=_a_a*
a=2/2
case 2
t=_ct_ta*_ta*
ta=_c_a_a**
c=_ct_a+
ct=_a_a+_a_a*+
a=2/4
case 3
c=_t_cg_cg//
t=_g_g+_cg_g**
cg=_g_ct/
```

ct=_g_g/
g=6/71
case 4
g=_tt_tt_gt+*
t=_gt_tt*_tt/
gt=_tt_tt+_a_a*/
a=_tt_tt_tt/*
tt=2/62
case 5
c=_cc_t*
ca=_a_a/_a/
a=_cc_cc_t*+
cc=_t_t_t/*
t=76/13

Output

case 1 Y [write 'Y' and the correct answer, write 'N' if you don't wish to answer]

a 1 1
c 3 1
cg 3 1
ct 1 1
g 1 1
case 2 Y
a 1 2
c 7 4
ct 5 4
t 245 1024
ta 7 16
case 3 Y
c 432 357911
cg 6 71
ct 1 1
g 6 71
t 432 357911
case 4 Y
a 1 31
g 1923 961
gt 62 1
t 62 1
tt 1 31
case 5 Y
a 6764 169
c 5776 169
ca 169 6764
cc 76 13
t 76 13

Score

0.05