

Branch Prediction

As most of you might already know, the Intel-class hi-tech processors of today do a series of parallel tasks to help speedup instruction execution. The most complicated of those tasks is branch prediction. Since the instruction chunks on a modern processor are broken down into independent chunks and executed for a speed up, there is always a requirement to predict what branch an execution path will take (before the actual operands required for the condition to be evaluated to select the branch, are available). This complex task, is not addressed to fullest level today, but heuristics (as always) have helped.

The task we are going to consider now is much more simple compared to the actual branch prediction task. For our modelling, let us suppose that every instruction has the following syntax:

All labels are strings of alphabets only. Labels are case-sensitive.

Moreover the probability that a certain branch will be taken is **P** (it is equal for all branches). If a branch is taken, the point of execution (control) goes to the branched-label. Otherwise the next statement in that order is executed. Control starts at the "**start**" (lowercase) label and control ends at the "**end**" (lowercase) label. The branch-label of start and end are themselves, and when start is executed, the control goes to the next instruction, and when end is executed, processing ends, with 100% probability. The last statement in the program is always an "**end**".

It is required to find the expected number of times a statement executes.

Input

T – the number of test cases;

For each test case:

1st line contains one integer N (the number of lines to follow), one real P and one label L.

Each of the N lines that follow consist of instructions (two labels).

Output

For each test case, output one line containing:

"Expected number of times label L is executed is R",

where L - is the label given in the input

R - is the number of times the label is expected to be executed. It must be printed with exactly five decimal places.

Constraints:

$T \leq 20$

$3 \leq N \leq 120$.

P lies between 0.01 and 0.99, i.e. no jump is 100% sure.

Also you can assume no label occurs on the jump side, without being defined throughout the program.

Each label is less than 10 characters in length.

Also each line has a distinct label associated with it.

Example

Sample Input:

```
3
5 .5 B
C start
start start
B C
D C
end end
5 .99 C
start start
A B
B A
C end
end end
3 .5 label
start start
label label
end end
```

Sample Output:

```
Expected number of times label B is executed is 4.00000
Expected number of times label C is executed is 1.00000
Expected number of times label label is executed is 2.00000
```