

Goto & labels

Vero is working on an important company, the company noticed that Vero codes using goto labels, her boss does not want Vero to code this kind of programs, the boss tells her that he will cut her payment by the number of lines contained between the label and the goto.

Vero does not know any iterative structures, and she is too worried to code the program she is asking you for, Code a solution for Vero's problem.

The solution to this is to count the lines of code that are between the label and the goto statement, having in consideration that the goto will always *return* to an existent label, the label will consist on a single string formed by letters from the 'a' to the 'z' and from the 'A' to the 'Z', all labels will have the format `_` as the goto will always contains the format `goto_`; however, remember that before the labels and the gotos you can have an important amount of spaces that will count as an indentation of a code.

Note: An uppercase letter label is distinct from a lowercase letter label, by example: 'abc' is not equal to 'ABC'

INPUT:

The first line of input will contain an integer T denoting the T test cases, then, T cases will follow. Each of the following line will contain an integer N, then, N lines will follow, each of the next N lines will contain a string S.

OUTPUT:

Output the string "Scenario #i: " where i is the test case you are analyzing followed by the sum of the number of lines between the label and the goto.

SAMPLE DATA:

INPUT	OUTPUT
3	Scenario #1: 2
6	Scenario #2: 7
A: SPACES will InDeNt the CoDe	Scenario #3: 7

No TaBs ArE iN the >> test << data

```
goto A;
```

```
B:
```

```
goto B;
```

```
7
```

```
B:
```

```
A:
```

```
cin >> n;
```

```
    if(n<0) then
```

```
        goto A;
```

```
    else
```

```
        goto B;
```

```
7
```

```
    A:
```

```
        TumbaBicho:
```

```
            printf("goto TumbaBicho");
```

```
            printf("goto TumbaBicho");
```

```
            goto TumbaBicho;
```

```
AA:
```

```
    goto A;
```

CONSTRAINTS:

$1 \leq N \leq 10000$

$1 \leq \text{Length of string} \leq 1000$