Although testing is a technique which does not allow us to prove the correctness of a program, it is used to assess its reliability, thus increasing our confidence in its correctness. In the literature, there are many testing methods to select test cases which are applied as an input to the program.

A programmer has the intention of testing a program. With this aim in mind, he needs to know how many test cases he should obtain such that each distinct path of execution is covered with a test case. Your task is to write a program which computes the number of different paths of execution from the initial sentence to the final sentence, in a given program P.

Input

The first line of the input contains the number of programs. Subsequently, each program is listed. Each program consists of a set of sentences with a keyword per line. The program finishes with the keyword `ENDPROGRAM`

There are only two types of sentences: conditional and non-conditional. A non-conditional sentence is represented by means of the keyword `S`. A conditional sentence is represented by the keywords `IF`, `ELSE` and `END_IF`. The condition of the `IF` sentence is omitted (i.e., only the `IF` keyword appears) and it is supposed that all `IF` sentences have a corresponding `ELSE`. So, for example, the shortest program with a conditional sentence would be:

```
IF
ELSE
END_IF
ENDPROGRAM
```

which has 2 paths of execution.

Output

For every program you are to print a line containing the number N of different paths of execution. It is supposed that N is never greater than $2 \wedge 32$.

Sample Input

```
4
IF
ELSE
END_IF
ENDPROGRAM
IF
S
ELSE
END_IF
IF
ELSE
S
END_IF
ENDPROGRAM
S
S
S
ENDPROGRAM
S
IF
S
S
ELSE
IF
IF
S
ELSE
S
END_IF
S
ELSE
S
END_IF
END_IF
S
ENDPROGRAM
```

Sample Output

```
2
4
1
4
```