Strange Lottery Simulator

Filename: `lottery`

Billy came up with a unique lottery idea: participants sign up with their names, and a random string *s* is generated in the draw, and the winners are the people whose name has prefix *s*. Yet there is a special rule, sometimes the names of all participants are reversed.

So for example, if Sharon signed up, then his name was reversed, and the string generated was "no," then he would win! However, if the string generated was "sha," he would not win. His name could be reversed an arbitrary number of times.

Billy would like you to help him write a simulator for the lottery. It should handle queries of three types (*s* represents a string):

1 *s*, Add a participant with name *s*.

2 *s*, Generate a string *s* in the draw and print the number of winners.

3, Reverse the names of all participants added.

Can you help Billy write an efficient simulator?

The Problem:

Write a program to simulate Billy's lottery. It is guaranteed that there is at least one query of type 2.

The Input:

The first line of the input file begins with a single, positive integer, *t*, representing the number of lotteries. For each lottery, multiple lines follow. The first contains a single integer $1 \leq q \leq 10^5$ representing the queries. Then, *q* lines follow, beginning with a single integer *t*, which is either 1, 2, or 3, denoting the type of the query. In queries of types 1 and 2, the integer will be followed by a space and then a string *s* of lowercase letters, whose length is at most 10^5. The sum of all string lengths in a test case is at most $6*10^5$.

The Output:

For each test case, output a single line saying "`Lottery #i:`" without the quotes, where `i` is the number of the lottery. Then, multiple lines should follow, one for each query of type 2, each with a single integer denoting the number of winners. Print a blank line after each test case.

(Sample Input and Output are on the next page)

Sample Input:

```
2
7
1 silly
1 billy
1 billiam
2 bill
2 ylli
3
2 ylli
5
1 carson
2 no
3
1 sharon
2 no
```

Sample Output:

```
Lottery #1:
2
0
2

Lottery #2:
0
1
```

Note: There is a blank line after the second test case.