Before bridges were common, ferries were used to transport cars across rivers. River ferries, unlike their larger cousins, run on a guide line and are powered by the river's current. Cars drive onto the ferry from one end, the ferry crosses the river, and the cars exit from the other end of the ferry.

There is an l-meter-long ferry that crosses the river. A car may arrive at either river bank to be transported by the ferry to the opposite bank. The ferry travels continuously back and forth between the banks so long as it is carrying a car or there is at least one car waiting at either bank. Whenever the ferry arrives at one of the banks, it unloads its cargo and loads up cars that are waiting to cross as long as they fit on its deck. The cars are loaded in the order of their arrival; ferry's deck accommodates only one lane of cars. The ferry is initially on the left bank where it broke and it took quite some time to fix it. In the meantime, lines of cars formed on both banks that await to cross the river.

The first line of input contains c, the number of test cases. Each test case begins with l, m. m lines follow describing the cars that arrive in this order to be transported. Each line gives the length of a car (in centimeters), and the bank at which the car arrives ("left" or "right").

For each test case, output one line giving the number of times the ferry has to cross the river in order to serve all waiting cars.

Sample input

4

20 4

380 left

720 left

1340 right

1040 left

15 4

380 left

720 left

1340 right

1040 left

15 4

380 left

720 left

1340 left

1040 left

15 4

380 right

720 right

1340 right

1040 right

Output for sample input

3

3

5

problem link: http://uva.onlinejudge.org/index.php?option=com\_onlinejudge&Itemid=8&category=355&page=show\_problem&problem=1975

6

In this problem we consider expressions containing brackets that are properly nested. These expressions are obtained by juxtaposition of properly netsted expressions in a pair of matching brackets, the left one an opening and the right one a closing bracket.

( a + $ ( b = ) ( a ) ) is properly nested

( a + $ ) b = ) ( a ( ) is not

In this problem we have several pairs of brackets, so we have to impose a second condition on the expression: the matching brackets should be of the same kind. Consequently (()) is OK, but ([)) is not. The pairs of brackets are:

( )

[ ]

{ }

< >

(\* \*)

The two characters `(\*' should be interpreted as one symbol, not as an opening bracket `(' followed immediately by an asterisk, and similarly for `\*)'. The combination `(\*)' should be interpreted as `(\*' followed by `)'.

Write a program that checks wheter expressions are properly nested. If the expression is not properly nested your program should determine the position of the offending bracket, that is the length of the shortest prefix of the expression that can not be extended to a properly nested expression. Don't forget `(\*' counts as one, as does `\*)'. The characters that are not brackets also count as one.

Input

The input is a text-file. Each line contains an expression to be checked followed by and end-of-line marker. No line contains more than 3000 characters. The input ends with a standard end-of-file marker.

Output

The output is a text file. Each line contains the result of the check of the corresponding inputline, that is YES (in upper case), if the expression is OK, and (if it is not OK) NO followed by a space and the position of the error.

Sample Input

(\*a++(\*)

(\*a{+}\*)

Sample Output

NO 6

YES

problem link:

<http://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=354&page=show_problem&problem=492>