Problem J
Pascal Meets Boole
Time Limit: 1 second, Memory limit: 2G

Many people are familiar with Pascal's Triangle, a triangular arrangement of integers named after the French mathematician and philosopher Blaise Pascal (1623–1662). If we number the rows of Pascal's Triangle $1, 2, 3, \ldots$, starting from the top, then row r contains r elements, which we will number $1, 2, \ldots, r$ from left to right. The 1^{st} and r^{th} elements in row r are set equal to 1, and for $r \geq 3$ and $1 < i < r$, element i in row r is the sum of elements $i-1$ and i in row $r-1$. More informally, each "non-edge" element is the sum of the two elements immediately above it. Figure J.1(a) depicts the first 8 rows of Pascal's Triangle.

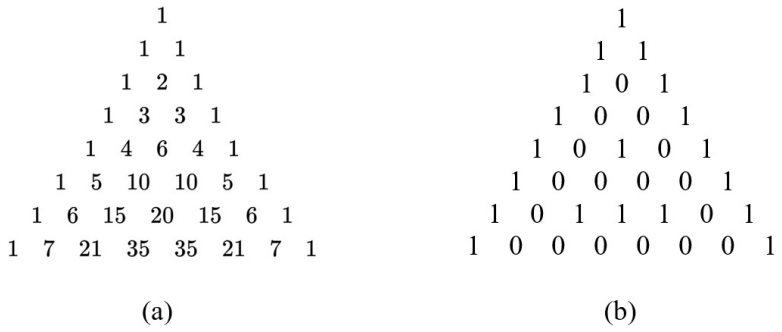


Figure J.1: (a) Pascal's Triangle, (b) Pascal-Boole triangle for function 1000

But what if we consider a rule other than standard addition for combining values? Since the edge elements are bits $(1$'s$)$, a natural option is to use any two-input Boolean function, named after the English mathematician and philosopher George Boole (1815–1864). For example, the Boolean function given by the following truth table generates the triangle depicted in Figure 1(b) (where we also show the first 8 rows$)$. In this truth table, x and y correspond to elements $i-1$ and i, respectively, in row $r-1$, and $f(x, y)$ is the resulting element i in row r.

x	y	$f(x, y)$
0	0	1
0	1	0
1	0	0
1	1	0

In general, if we label the bits in the rightmost column of any such truth table $b_{00}, b_{01}, b_{10}, b_{11}$ from top to bottom, then we can compactly represent a two-input Boolean function by the 4-bit string $b_{00}b_{01}b_{10}b_{11}$. So the example function above is represented by 1000.

Your challenge is to answer two kinds of questions about "Pascal-Boole" triangles:

1. For a given Boolean function, f, what is the bit in row r, position i?
2. For a given Boolean function, f, how many 1's are there in the first r rows?

Input

The first line of input contains an integer, n $(1 \leq n \leq 250)$, the number of test cases. This is followed by n lines, each of which has one of two forms:

1. f `B` r i
2. f `N` r

In both cases, f is a 4-bit binary string representing a two-input Boolean function, and r is an integer $(1 \leq r \leq 10^6)$. In the first case, i is an integer $(1 \leq i \leq r)$.

Output

For a test case of the form f `B` r i, output a line containing the bit in row r, position i of the Pascal-Boole triangle generated using f. For a test case of the form f `N` r, output a line containing the number of 1's in the first r rows of the Pascal-Boole triangle generated using f.

Sample Input 1	Sample Output 1
3 1000 B 5 3 1111 N 7 0100 B 6 4	1 28 0