Problem A. Huge Table

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	256 megabytes

We have a rectangular table consisting 2^n rows and 2^m columns. The integer numbers are written from left to right, from top to bottom, in the cells of table. for an instance, when $n = 2$ and $m = 3$, our table is like this:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

For convenience of presentation we use the notation $[r, c]$ to denote a cell of table. $V[r, c]$ is the number written in $[r, c]$.

Let's define three functions:

- $popcount(n)$ is sum of digits in the binary representation of n. n should be non-negative. For example $popcount(7) = 3$ and $popcount(10) = 2$.
- $minpath(r, c)$: Consider a sub-rectangle of our table that $[1, 1]$ is its top-left corner and $[r, c]$ is its right-bottom. We want to go from $[1, 1]$ to $[r, c]$ and in each step we are just allowed to move one cell right or down. For each cell $[x, y]$ there is an entrance cost which is equal to $popcount(V[x, y])$. The function $minpath(r, c)$ calculates the minimum cost we should pay to reach from $[1, 1]$ to $[r, c]$.
- $sum(r_1, c_1, r_2, c_2)$ calculates $\sum popcount(V[r, c])$ for all cells like $[r, c]$ that lie in the sub-rectangle described by $[r_1, c_1]$ and $[r_2, c_2]$. Formally $(r_1 \le r \le r_2, c_1 \le c \le c_2)$.

Now you should find answer of the following interesting queries:

1. MINPATH r c: Print $minpath(r, c)$
2. SUM r_1 c_1 r_2 c_2: Print $sum(r_1, c_1, r_2, c_2)$

Input

There are three integers in the first line: n, m and q $(0 \le n \le 50, 0 \le m \le 50, 1 \le q \le 100)$.
The next q lines contain the queries in the format, given in the statement.
For MINPATH queries: $(1 \le r \le 2^n, 1 \le c \le 2^m)$ For SUM queries: $(1 \le r_1 \le r_2 \le 2^n, 1 \le c_1 \le c_2 \le 2^m)$

Output

For each query, output a single integer in a line corresponding to the answer of each query. Answers may become very large, so output them modulo 1000000009.

Examples

stdin	stdout
2 3 5 MINPATH 1 1 SUM 2 2 3 5 MINPATH 3 3 SUM 1 1 4 8 MINPATH 4 7	1 20 9 81 24

Note

For the last query of the first sample " MINPATH 4 7 ", we can move in the following path:
$[1,1] \to [1,2] \to [1,3] \to [1,4] \to [2,4] \to [3,4] \to [3,5] \to [3,6] \to [4,6] \to [4,7]$
Cost of the path is: $popcount(1)+popcount(2)+popcount(3)+popcount(4)+popcount(12)+popcount(20)+popcount(21)+popcount(22)+popcount(30)+popcount(31) = 1+1+2+1+2+2+3+3+4+5 = 24$