

Where's my car!

Problem Description

The year is 2050. The population surge has taken over our metropolitan cities. High rise buildings are on a rise. And as usual, with the increase in population, the problem of parking in the city has increased manifold.

To reduce the problem of parking, the government has built multi-storey parking lots all over the city. Imagine the city as an X-Y grid. And there are roads connecting all the neighbouring grid points except diagonals. And there is a pre-defined intersection interval 'I' for parking lots such that at every Ith intersection, there is a parking lot, starting from (0,0). For example, for a city of grid size 4×7 and I=3, you'll have 6 parking lots at (0,0), (0,3), (0,6), (3,0), (3,3) and (3,6).

Now all the cars have been fitted with self-driving mechanism. So whenever you get out of a car at any point in the grid, it will choose the nearest parking lot and automatically drive to it. If two parking lots are at equal distance from where you left, it will choose the parking lot with the lowest X-coordinate first, and if X-coordinates are same, the lowest Y-coordinate.

At the parking lot, the cars will start getting parked from the ground floor and in the first available slot. As each floor gets filled up, newer cars will start parking on floors above them. Assume all the parking lots in the city have unlimited number of floors and a common maximum capacity of each floor 'C'.

Now whenever the owner wants to know where his car is parked or wants to retrieve it, he'll open the app 'Where's my car!' and insert his car number and the app will tell him the coordinates of the parking lot, the floor number and the slot number.

Input Format:

The first line will contain a positive integer T determining the number of test cases.

For each test case, the first line will contain four positive integers X, Y, I and C, delimited by space where,

1. X, Y denote the grid size of the city,
2. I denote the intersection interval for parking lots
3. C denotes the maximum capacity of a floor in all the parking lots in the city.

The next line will contain a positive integer N denoting the number of events following.

1. An event can be parking of a car(P) or retrieval of a car (R).
2. The next N lines will begin with either a character 'P' or 'R'.
3. If the line begins with P, it will also contain two positive integers, x and y denoting the current coordinate of the car where it is left off and has to drive itself to the nearest parking lot, and the car number S which is a unique 10-digit number for each car and can contain a mix of numbers and alphabet.
4. If the line begins with 'R', it will contain the car number to retrieve.

Output Format:

For each test case, output the following.

First output the total number of parking lots in the city.

Then, for every retrieval event in the input list, output 4 space separated integers, p, q, r, s where

1. p, q is the coordinate of the parking lot where the car is parked in,
2. r is the floor number and s are the slot number.

3. For brevity, output each number p, q, r, s, modulo 10.

Constraints:

1. $1 \leq T \leq 10$
2. $1 \leq X, Y < 2,000,000,000$
3. $1 \leq C, N < 100,000$
4. $1 \leq I \leq X, Y$
5. $0 \leq p \leq X$
6. $0 \leq q \leq Y$
7. Max no. of parking lots $< 1,000,000$

Sample Input and Output

Sr. No	Input	Output
1	1 4 7 2 2 5 P 1 1 MH04CZ2123 P 3 2 BR11HY2863 R MH04CZ2123 P 3 3 KA14NU7644 R KA14NU7644	8 0 0 0 1 2 2 0 2
2	1 4 7 2 2 5 P 2 1 MH04CZ2123 P 3 5 BR11HY2863 R MH04CZ2123 P 3 4 KA14NU7644 R KA14NU7644	8 2 0 0 1 2 4 0 2