Problem L. Patrol Drone

Input file:	standard input
Output file:	standard output
Time limit:	5 seconds
Memory limit:	512 mebibytes

You are trying to steal a very valuable diamond tiara from the museum. What makes it tempting is that (due to budget cuts) all the guards have been replaced by a single automatic drone that patrols the main hall of museum. What makes it less tempting, though, is that this drone is equipped with some very modern weaponry and you are likely to be disintegrated if you attract its attention.

Fortunately, you have done your homework and have a pretty good grasp of how the drone works. Imagine that the hall is an Euclidean plane, divided into unit square cells. The central cell, $(0,0)$, contains the tiara. The drone has a simple processing unit that stores two things: its current position (x, y) and the sequence of instructions, denoted by letters 'N', 'E', 'S', 'W'. Every minute, the drone moves to an adjacent spot according to the first letter of the sequence (North, South, West or East), changes the stored position accordingly, and then moves this first letter to the end of the sequence to access the next one in the next minute. If the instruction sequence is empty, the drone simply does nothing. It is guaranteed that these instructions describe a closed loop, and that the drone never enters the $(0,0)$ cell.

Right now, the drone is at position (x_0, y_0) and has a string T of instructions. You can modify the drone's memory by some clever hacking – your goal is to reach a situation in which the drone is at the same position (x_0, y_0), but with different string T'. Your hacking strategy, unfortunately, is somewhat limited – each minute, you can only access the two front letters of the string and perform any number of the following operations:

- remove two front letters from the string, but only if they are "NS", "SN", "EW" or "WE";

- add two letters "NS", "SN", "EW" or "WE" to the front of the string;

- swap two front letters of the string (any combination of letters may be swapped).

Also, there is a collision detection system implemented on the drone, which detects if the current set of instructions would possibly bring the drone to $(0,0)$. An alarm is raised in this case – this is the situation you want to avoid at all costs.

Find a sequence of hacking operations which will bring the drone to (x_0, y_0) with the desired sequence T' in its memory.

Input

The first line of input consists of a single number z ($1 \le z \le 100$), denoting the number of test cases. The descriptions of the test cases follow.

The first line of each test case consists of two integers x_0, y_0 ($-1000 \le x_0, y_0 \le 1000$), denoting the initial position of the drone. At least one of numbers x_0, y_0 is not zero.

The second line contains two numbers n, m ($2 \le n, m \le 2000$) – the length of the current (T) and target (T') string, respectively.

The next two lines contain one string each, of length n and m respectively, denoting T and T', consisting only of letters N, S, E, and W.

It is guaranteed that the current and target sequences are different. Moreover, both describe some closed loops and neither of these loops cross $(0,0)$ at any moment of the route.

The total number of letters in all test cases does not exceed 20 000.

Output

For each test case, if it's impossible to meet the task's requirements, output "NO" in a single line. Otherwise, output "YES", and then a solution description on the next line. The solution must consist only of symbols $\{C, N, S, E, W, -, R\}$, where each character indicates a single hacking operation:

- Symbol 'N' means adding "NS" at the front of the string.

- Similarly, symbols 'S', 'E' and 'W' mean adding "SN", "EW" and "WE" at the front.

- Symbol 'R' means removing two first letters from the string – this is only allowed if these letters are "NS", "SN", "EW" or "WE".

- Symbol 'C' means swapping two first letters.

- Symbol '-' means that you wait for the rest of the minute (until the drone moves to the next instruction).

Note that multiple hacks can be made in a single minute. **You do not need to minimize solution length**, but the actions description must have no more than $2 \cdot 10^7$ characters. By the end of the last minute of your output, the string and the position of the drone must match the desired ones. Removing or swapping elements of a string with at most one letter is not allowed. At no point, the loop described by the drone's sequence can go through $(0, 0)$.

Example

standard input	standard output
2	YES
1 0	-C-C-R--S-C-C---
10 10	NO
NNWWSSSEEN	
NWWSSSEENN	
-1 0	
8 8	
NEESSWWN	
SEENNWWS	