Problem E
Reprogram the Robot

You have a programmable robot that is able to move around following a given command. A command is a string containing only `N`, `S`, `E`, and `W` characters. Each character represents a move order.

- `N` – move 1 unit to the north, i.e. from (x, y) to (x, y+1)
- `S` – move 1 unit to the south, i.e. from (x, y) to (x, y-1)
- `E` – move 1 unit to the east, i.e. from (x, y) to (x+1, y)
- `W` – move 1 unit to the west, i.e. from (x, y) to (x-1, y)

Each character in the command is executed one by one sequentially. For example, let the robot starts at (4, 7) and the command be `NWNEEESEN`, it will end up at (7, 9). The movements are as follows: (4, 7) → (4, 8) → (3, 8) → (3, 9) → (4, 9) → (5, 9) → (6, 9) → (6, 8) → (7, 8) → (7, 9).

Initially, the robot is located at (x_s, y_s) and you want to move the robot to (x_t, y_t) with a command—specifically, the robot's **final** location should be at (x_t, y_t).

There is an existing command C ready to be executed by the robot but this command might or might not take the robot from (x_s, y_s) to (x_t, y_t). You don't want to write an entirely new command; instead, you are going to remove zero or more characters from C such that the command will take the robot to your target location from its initial location when it's executed.

Your task is to find the minimum number of characters you need to remove from C to achieve your goal.

For example, let the robot's initial location be (0, 0), the target location be (3, 3), and the existing command C is `NNSNNEEEESE`. In this case, you need to remove 3 characters, e.g., $C_3 =$ `S`, $C_6 =$ `E`, and $C_7 =$ `E`, so that the command becomes `NNNNEESE`. This commands will take the robot from (0, 0) to (3, 3) with the following movements: (0, 0) → (0, 1) → (0, 2) → (0, 3) → (0, 4) → (1, 4) → (2, 4) → (2, 3) → (3, 3). There are other combinations of characters to be removed that can achieve the same goal, but none is fewer than 3 characters.

Sometimes it might not be possible for the robot to move to the target location. In such a case, you should output -1.

Input

Input begins with an integer T $(1 \leq T \leq 1000)$ representing the number of cases.

Each case begins with two integers x_s y_s $(0 \leq x_s, y_s \leq 1000)$ representing the initial location of the robot. The next line contains two integers x_t y_t $(0 \leq x_t, y_t \leq 1000)$ representing the target location. The last line on each case contains a string C $(C_i \in \{\texttt{N}, \texttt{S}, \texttt{E}, \texttt{W}\})$ representing the given command. The length of string C is at least 1 and no more than $100\,000$.

It is guaranteed that the total length of C over all cases does not exceed 10^6.

Output

For each case, output in a line "`Case #X: Y`" (without quotes) where `X` is the case number (starts from 1) and `Y` is the output for the respective case.

Sample Input #1

```
4
0 0
3 3
NNSNNEEEESE
10 20
8 23
WWWNSNNWNEE
5 2
5 7
NSEWNSEWNSEWNSEW
0 0
3 0
EEEWEWEW
```

Sample Output #1

```
Case #1: 3
Case #2: 0
Case #3: -1
Case #4: 1
```

Explanation for the sample input/output #1

For the 2^{nd} case, you don't need to remove any character. The command will take the robot to the target location. The robot's movements are: (10, 20) → (9, 20) → (8, 20) → (7, 20) → (7, 21) → (7, 20) → (7, 21) → (7, 22) → (6, 22) → (6, 23) → (7, 23) → (8, 23).

For the 3^{rd} case, it is not possible for the robot to move to the target location.

For the 4^{th} case, you need to remove one character, e.g., $C_4 =$ `W`, so that the command becomes `EEEEWEW`. With this command, the robot's movements are: (0, 0) → (1, 0) → (2, 0) → (3, 0) → (4, 0) → (3, 0) → (4, 0) → (3, 0). It ends up at (3, 0), the target location.