**Moving Balls**

**Time Limit: 3 Seconds**

Alice and Bob are going to participate in a gaming contest as a team. This is not traditional gaming contest where you play a game against your opponents. This is actually an idea contest where each team will demonstrate their gaming idea and judges will decide the winner based on creativity and gameplay. Being habitual in playing weird games among them, Alice and Bob came up with a very unusual board game idea for the contest.

The game board consists of a 2D grid. There are **K** moving balls in the board, initially placed on different **integer coordinate** points. Two ball will never placed on the same point. Each ball has an unique **ID** number from **1 to K**.

Each ball can move in a fixed direction, either horizontally or vertically. The direction of each ball is specified by the characters **N, S, E and W** where:

* 'N' means north (y coordinate increases)
* 'E' means east (x coordinate increases)
* 'S' means south (y coordinate decreases)
* 'W' means west (x coordinate decreases)

This is a two player game. All the balls are distributed among the two players in following ways: one player will take all the balls with **even ID** and the other player will take all the balls with **odd ID**. That means, if the number of total balls is odd then initially one player will always have one more ball than the other player.

Before the game begins, there will be a toss. The winner of the toss will decide if he will take the balls with even ID or, odd ID.

Initially all balls are stopped. As soon as the game begins, all balls started moving at the ***same speed*** in their pre-specified direction. When two or more balls collide, they get damaged and are removed from the game board. Note that, not only opponent balls can damage each other but also same player’s balls can get damaged when they collide.

The game will stop automatically when no more collision can be possible between the remaining balls. At the end of the game, the player who has greater number of undamaged balls remains in the board will win the game. If they have same number of remaining balls, then it is a draw.

Lets demonstrate a game scenario: consider there are 5 balls with following properties:

| ID | Direction | Initial Coordinate |
| --- | --- | --- |
| 1 | E | (0, 0) |
| 2 | W | (2, 0) |
| 3 | N | (10, 10) |
| 4 | N | (0, 1) |
| 5 | S | (0, 2) |

Here, balls with ID-1 and ID-2 will collide at point (1, 0). Again balls with ID-4 and ID-5 will collide at point (0, 1.5). Only ball with ID-3 will never collide and remain undamaged. So, number of undamaged balls with even ID is 0 and odd ID is 1. Hence, the player who took the balls with odd ID will win the game.

While everyone applauding this amazing idea, you immediately discovered the flaw of the game. You found that, it is possible to identify the balls that will never collide without waiting for the whole game to be finished. So if the winner of the toss is clever enough, he will never lose the game. To prove yourself, you have to determine the **ID** of the undamaged balls for different gaming scenarios.

**Input:**

Input start with an integer **T (1 <= T <= 100)**, denoting the number of test cases.

Each test case consists of **four** lines of input. First line will have an integer **K (1 <= K <= 50)**, denoting the number of balls in the game board. Second line will have a string of **K** characters, specifying the directions of the balls. Third line will have **K** integers denoting the initial **X** coordinates of the balls and the final line will have **K** integers denoting the initial **Y** coordinates of the balls where **(-100 <= X, Y <= 100)**. Note that, these inputs are given in the ascending order of the balls **ID**.

**Output:**

For each case, print the case number and the number of undamaged balls. Then print the **ID** of the undamaged balls in ascending order in the same line.

| **Sample Input** | **Sample Output** |
| --- | --- |
| 3  5  EWNNS  0 2 10 0 0  0 0 10 1 2  4  NWNE  0 10 20 30  0 10 20 30  4  NEWS  -10 0 0 10  0 -10 10 0 | Case 1: 1 3  Case 2: 2 3 4  Case 3: 0 |