

## Problem 2 – Sneaking

After our hero Sam got the recipe from the first problem, there is another thing he needs to check off from his to-do list. In order to make the recipe even more valuable, he needs to “eliminate” anyone who possesses the knowledge of it. That person is Sam’s sworn enemy - **Nikoladze**. Sam needs to get through a rectangular room of **patrolling enemies** until he finally **reaches Nikoladze**.

A standard room looks like this:

Room	Legend
.....N.... b..... ..d..... .....d... .....S....	<b>S → Sam</b> , the player character <b>b/d → left/right-facing patrolling enemy</b> <b>N → Nikoladze</b> <b>. → Empty space</b>

Each turn proceeds as follows:

- **First, Enemies** move either **left** or **right**, depending on which **direction** they are **facing** (**b** goes **right**, **d** goes **left**)
  - If an enemy is standing on the **edge** of the room, he flips his **direction** (from **d** to **b** or from **b** to **d**) and **doesn’t move** for the rest of the turn.
- If an enemy is on the **same row** as Sam, and also **facing Sam** (eg. **.b.S.**), the **enemy kills Sam**.
- After that, Sam moves in the **direction** he is instructed to (either **U/D/L/R** or **W**).
  - **U -> Up, D -> Down, L -> Left, R -> Right, W -> Wait (Sam doesn’t move)**
- If **Sam** moves **onto an enemy** (**same row and column**), Sam kills the enemy and **leaves no trace of him**.
- If Sam reaches the **same row** as **Nikoladze**, Sam kills **Nikoladze** (replacing him with an **X**)

### Problem 1. Input

- On the **first line** of input, you will receive **n** – the **number of rows** the **room** will consist of. Range: **[2-20]**
- On the next **n lines**, you will receive the **room**, which Sam will have to navigate.
- On the **final line** of input, you will receive a sequence of **directions** – one of (**U, D, L, R, W**)

### Problem 2. Output

- If Sam is **killed**, print **“Sam died at {row}, {col}”**
- If Nikoladze is **killed**, print **“Nikoladze killed!”**
- Then, in both cases, **print the final state of the room** on the **console**, with either **Sam** or **Nikoladze’s symbols** replaced by an **X**.

## Problem 3. Constraints

- The room will always be **rectangular**.
- There will **always** be enough moves for **Sam** to reach **Nikoladze**
- There will be **no case** where **Sam** is instructed to move **out of the bounds of the room**.
- There will be **no case** with **two enemies on the same row**.
- There will be **no case** with an **enemy and Nikoladze** standing on the **same row**.
- There will be **no case** where Sam reaches the **same row and column** as **Nikoladze**.

## Problem 4. Examples

Input	Output	Comments
5 .....N... b..... ..d..... .....d... .....S... UUUUR	Sam died at 2, 5 .....N... ...b..... b....X.... ..... ..... .....	Turn 1: Enemies move, then Sam <b>steps on</b> the enemy on the 4 <sup>th</sup> row. Turn 2: Enemies move, then Sam moves. Turn 3: Enemy 2 <b>turns around, sees Sam</b> and <b>kills him</b> .
3 N..... .b..... ..dS... WUUU	Nikoladze killed! X..S... ..... b.....	Turn 1: Enemies move, Sam waits. Turn 2: Enemies move, Sam goes <b>up, steps on an enemy</b> . Turn 3: Enemies move, Sam goes <b>up, kills Nikoladze</b> .
6 ..... .....S..... .b..... .....d. ..... .....N..... WWWDWWDDRD	Nikoladze killed! ..... ..... .....b d..... ..... ....XS.....	Turn 1/2/3: Enemies <b>move</b> , Sam <b>waits</b> . Turn 4: Enemies <b>move</b> , Sam goes <b>down</b> . Turn 5/6/7: Enemies <b>move</b> , Sam <b>waits</b> . Turn 8/9: Enemies <b>move</b> , Sam goes <b>down</b> . Turn 10: Enemies <b>move</b> , Sam goes <b>right</b> . Turn 11: Enemies <b>move</b> , Sam goes <b>down</b> and <b>kills Nikoladze</b> .