**02. Collecting Coins**

*You are playing a game, and your goal is to collect 100 coins.*

On the first line, you will be given a **number** representing the **size of the field** with a **square** shape. On the following few lines, you will be given the **field** with:

* **One player** - randomly placed in it and marked with the symbol "**P**"
* **Numbers** for coins placed at different positions of the field
* **Walls** marked with "**X**"

After the field state, you will be given **commands** for the **player's movement**. Commands can be: "**up**", "**down**", "**left**", "**right**". If the command is invalid, you should ignore it.

The player **moves** in the given **direction** with one **step for each command and collects all the coins that come across**. If he goes out of the field, he should **continue to traverse the field** from the **opposite side** in the **same direction**.

**Note:** He can go through the **same path many times**, but he can **collect** **the coins** **just once** (the first time).

There are only **two** **possible outcomes** of the game:

* The player **hits a wall**, **loses the game**, and **his coins are reduced to 50% and rounded down** tothe next-lowest number.
* The player collects **at least** **100 coins** and wins the game.

For more clarifications, see the examples below.

### Input

* **A number** representing the size of the field (matrix NxN)
* **A matrix** representing the field (each position **separated by a single space**)
* On each of the following lines, you will get a move command.

### Output

* If the player won the game, print: "**You won! You've collected {total\_coins} coins.**"
* If the player loses the game, print: "**Game over! You've collected {total\_coins} coins.**"
* Collected coins have to be **rounded down** tothe next-lowest number.
* The player's path as **cooridnates** **in lists on separate lines:**

"**Your path:**

**[{row\_position1}, {column\_position1}]**

**[{row\_position2}, {column\_position2}]**

**…**

**[{row\_positionN}, {column\_positionN}]**"

### Constrains

* There will be nocase in which less than 100 coins will be in the field
* All given numbers will be valid integers in the range [0, 100]

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  1 X 7 9 11  X 14 46 62 0  15 33 21 95 X  P 14 3 4 18  9 20 33 X 0  left  right  right  up  up  right | You won! You**'**ve collected 125 coins.  Your path:  [3, 0]  [3, 4]  [3, 0]  [3, 1]  [2, 1]  [1, 1]  [1, 2] |
| 8  13 18 9 7 24 41 52 11  54 21 19 X 6 4 75 6  76 5 7 1 76 27 2 37  92 3 25 37 52 X 56 72  15 X 1 45 45 X 7 63  1 63 P 2 X 43 5 1  48 19 35 20 100 27 42 80  73 88 78 33 37 52 X 22  up  down  up  left | Game over! You**'**ve collected 0 coins.  Your path:  [5, 2]  [4, 2]  [5, 2]  [4, 2]  [4, 1] |