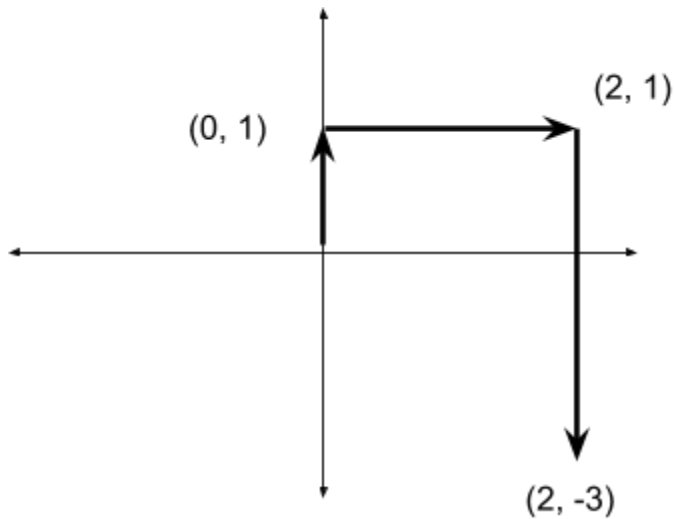


Robot Walk

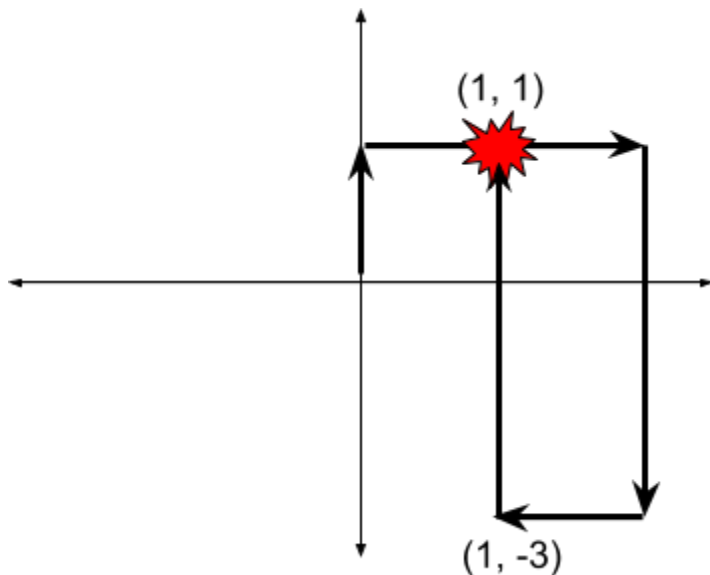
A robot is placed at the origin on a Cartesian coordinate system $(0, 0)$ and faces North. The robot is given a set of instructions, each of which are expressed as a positive integer. With each instruction, the robot takes an equal number of steps forward, then turns right. The robot covers 1 unit on the coordinate system with each step. However, if at any point the robot walks across a point it has already visited, it stops there and ignores all future instructions.

For example, with instructions $x = [1, 2, 4]$:



The robot moves 1 step north to $(0, 1)$, turns right and takes 2 steps east to $(2, 1)$, and finally turns right and takes 4 steps south to $(2, -3)$.

For example, with instructions $x = [1, 2, 4, 1, 5]$:



On its final instruction, the robot moves from (1, -3) north 5 steps towards (1, 2). However, after 4 steps, the robot realizes it has already been to coordinate (1, 1), and therefore stops there before completing this instruction.

Write a function:

```
function robotWalk($X);
```

that, given the robot instructions X in the form of zero-indexed arrays consisting of non-negative integers, returns the final Cartesian coordinate of the robot, expressed in a 2-element array.

For example, given $x = [1, 2, 4]$ the function should return $[2, -3]$, as explained above.

For example, given $x = [1, 2, 4, 1, 5]$ the function should return $[1, 1]$, as explained above.

Solution should have a $O(1)$ space complexity and $O(n)$ time complexity.