



# [CS 11] Prac 9e – Parallel Universes II

## Problem Statement

Even after his previous maneuvers, Mario is still QPU misaligned. Worse, because of a mishap during scuttlebug raising, he can now only turn clockwise!

We can think of the multiverse as an infinite rectangular grid of parallel universes. Thus, we can label each parallel universe with a **lattice point**, that is, a point on the Cartesian plane with integer coordinates.

Mario is currently at universe  $(x_0, y_0)$  and is currently heading north. He then performs  $n$  movements. Each movement consists of the following steps:

- moving some number of steps toward the direction Mario is facing. One "step" means going to the universe one unit across.
- turn clockwise 90 degrees. (For example, if he's facing north, then he will turn east.)

Please provide the sequence of all intermediate universes that Mario ends up in, including the starting one.

### Notes:

- "North" is the positive  $y$  direction.
- "East" is the positive  $x$  direction.

## Task Details

Your task is to implement a function called `move2d`. This function has two parameters:

- the first parameter is a pair of `int`s  $(x_0, y_0)$  denoting Mario's initial location universe.
- the second parameter is an iterable of `int`s describing the movements. The `int` represents the number of steps for that movement.

The function must return a *generator* that generates pairs of `int`s denoting the intermediate universes, including the first one.

Note that your generator must be **as lazy as possible**. It should yield each resulting next element as soon as it has enough information, and it should produce these results while advancing the input generators for as little as possible.

## Restrictions

(See 9a for more restrictions)

For this problem in particular:

- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- The source code limit is 2000.

## Example Calls

### Example 1 Function Call

```
*move2d((-1, 1), (3, 2, 2, 3, 1, 4))]
```

Copy

### Example 1 Return Value

```
[(-1, 1),
 (-1, 4),
 (1, 4),
 (1, 2),
 (-2, 2),
 (-2, 3),
 (2, 3),]
```

Copy

### Constraints

- The function `move2d` will be called at most 200 times.
- At most 500 elements will be consumed from the returned generator.
- $|x_0|, |y_0| \leq 10^{10}$
- Each element of the input sequence is a positive integer at most  $10^{10}$ .

## Scoring

- You get 125 ❤ points if you solve all test cases.

## Clarifications

Report an issue

No clarifications have been made at this time.