

Multiverse engineering take-home challenge

- Use **any programming language** you want.
- It shouldn't take longer than **2-3 hours**.
- **Don't worry if you don't finish in the time.** Please send over what you've achieved and a few sentences on what you would add next.
- Please **do not implement any UI** for this. We just want to see the quality of your code - the console is fine for output.
- Put your code online (e.g. in GitLab/GitHub) and send us the link.

The problem - Mars Rover

Write a program that takes in commands and moves one or more robots around Mars.

- The world should be modelled as a grid with size $m \times n$
- Your program should read the input, update the robots, and print out the final states of the robots
- Each robot has a position (x, y) , and an orientation (N, E, S, W)
- Each robot can move forward one space (F), rotate left by 90 degrees (L), or rotate right by 90 degrees (R)
- If a robot moves off the grid, it is marked as 'lost' and its last valid grid position and orientation is recorded
- Going from $x \rightarrow x + 1$ is in the easterly direction, and $y \rightarrow y + 1$ is in the northerly direction. i.e. $(0, 0)$ represents the south-west corner of the grid

The input takes the form:

```
4 8
(2, 3, E) LFRFF
(0, 2, N) FFLRFF
```

The first line of the input '4 8' specifies the size of the grid. The subsequent lines each represent the initial state and commands for a single robot. $(0, 2, N)$ specifies the initial state of the form $(x, y, \text{orientation})$. FFLRFF represents the sequence of movement commands for the robot.

The output should take the form:

```
(4, 4, E)
(0, 4, W) LOST
```

Each line represents the final position and orientation of the robots of the form $(x, y, \text{orientation})$ and optionally whether the robot was lost.

Another example for the input:

```
4 8  
(2, 3, N) FLLFR  
(1, 0, S) FFRLF
```

The output would be:

```
(2, 3, W)  
(1, 0, S) LOST
```