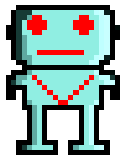


PROGRAM YOUR PARENT



This is Ree-borg, the only robot available to help you fix the *Candium*-drive in your ship. He is old, leaky, and can only turn left and move forward. You need to create a sequence of instructions so that he can retrieve a sample of Candium for you to test without getting exposing to candium radiation.

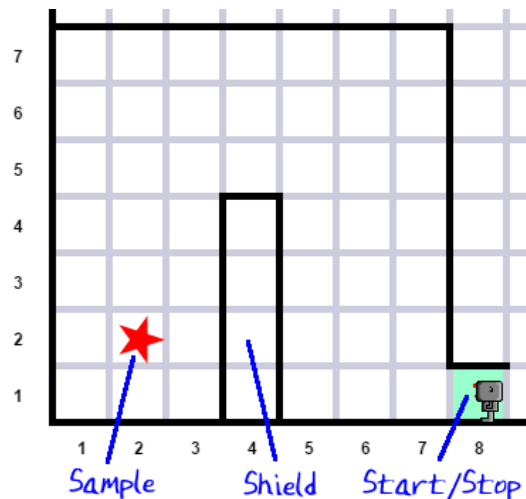
Symbol	Instruction	Symbol	Instruction	Symbol	Create your own
↑	Move forward one square.	↗	Pick up sample from the square robot is in.		
↶	Turn left 90 degrees.	↘	Put down any object you are holding.		

For example, to move forward one space and up one space, you need:



Write your instructions here:

Start									
									Stop



If you have a set of instructions you repeat often, like to take a number of steps forward, you could define a function that represents it like $\text{forward}(\text{steps}) = \uparrow \times \text{steps}$

Can you write a function to turn right? ↷ =

I hope you had fun with your robot! Programming can be challenging, but like solving any puzzle, very rewarding! You can program Reeborg online at <http://reeborg.ca/world.html>. There is even a “Program Your Parent” world to load and you can learn to use languages like Python and JavaScript!

Learn more at MAKERSBOX.BLOGSPOT.COM

```
world.txt (save as txt file, open from "World" box:
{"robots":[{"x":8,"y":1,"tokens":0,"orientation":2,"_is_leaky":true,"_prev_x":8,"_prev_y":1,"_prev_orientation":2,"triangle":0,"square":0,"star":0,"start_positions":[[8,1]]},"large_world":false,"walls":{"1,7":["north"],"2,7":["north"],"3,7":["north"],"4,7":["north"],"5,7":["north"],"6,7":["north"],"7,7":["north","east"],"7,6":["east"],"7,5":["east"],"7,4":["east"],"7,3":["east"],"7,2":["east"],"4,1":["east"],"4,2":["east"],"4,3":["east"],"4,4":["east","north"],"3,4":["east"],"3,3":["east"],"3,2":["east"],"3,1":["east"],"8,1":["north"]},"goal":{"possible_positions":[[8,1]],"position":{"x":8,"y":1},"shapes":{"8,1":"star"}},"shapes":{"2,2":"star"}}
```

Example solution:

```
def forward(steps):
    for x in range(steps):
        move()
```

```
def turn_right():
    for x in range(3):
        turn_left()
```

```
forward(2)
turn_right()
forward(4)
turn_left()
forward(4)
turn_left()
forward(3)
take("star")
turn_left()
turn_left()
forward(3)
turn_right()
forward(4)
turn_right()
forward(4)
turn_left()
forward(2)
put("star")
```