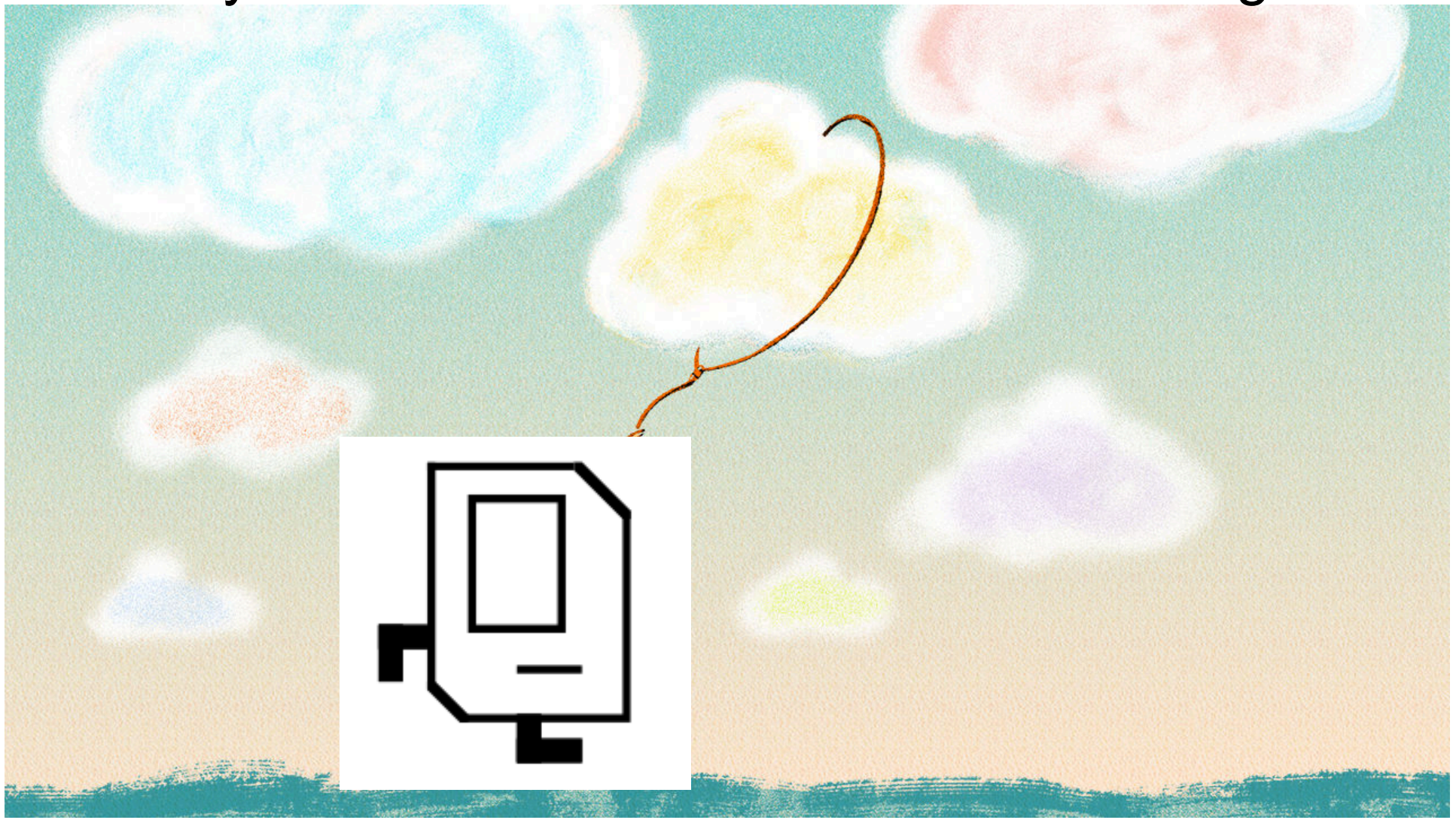
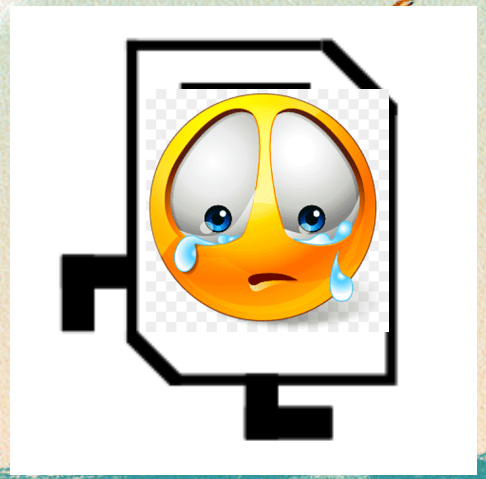

Control Flow

Anyone who saw Karel in a dream last night?



You know I just know how to

```
move
turn left
put beeper
pick beeper
```



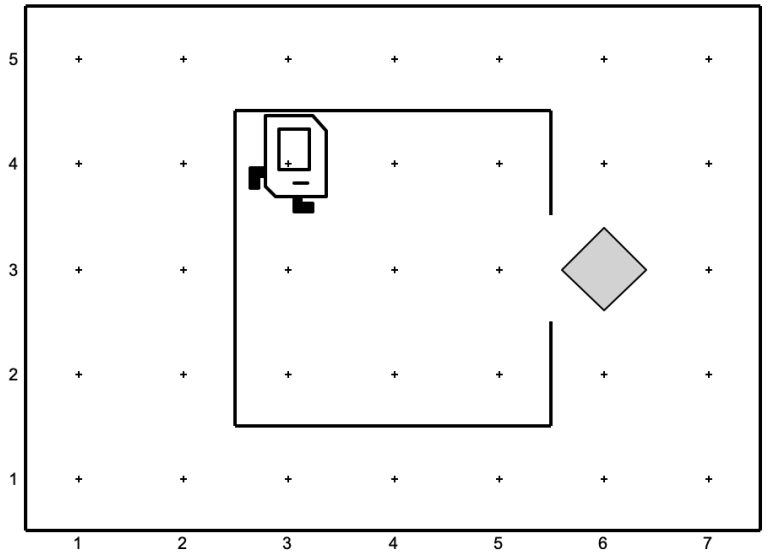
I am ashamed to say I cannot turn right

No worries, we can define it for you

```
def turn_right():
    turn_left()
    turn_left()
    turn_left()
```

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

You taught Karel how to pick newspapers, did she do it this morning?



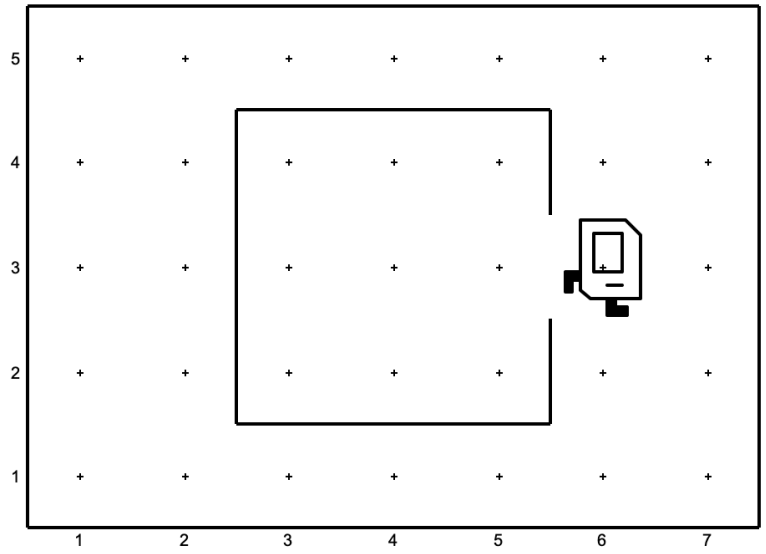

```
def main():
    move_to_newspaper()
    pick_beeper()
    move_to_start()

if __name__ == "__main__":
    run_karel_program()
```

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

def move_to_newspaper():
    move()
    move()
    turn_right()
    move()
    turn_left()
    move()
…
```

You taught Karel how to pick newspapers, did she do it this morning?




```
def main():
    move_to_newspaper()
    pick_beeper()
    move_to_start()

if __name__ == "__main__":
    run_karel_program()
```

```
…
def turn_around():
    for i in range(2):
        turn_left()

def move_to_start():
    turn_around()
    for i in range(3):
        move()
    turn_right()
    move()
    turn_right()
```

```python
from karel.stanfordkarel import *

"""
File: collect_newspaper_karel.py
-----------------------------
Karel picks beeper in front of his house.
"""


def main():
    pick_beeper()
    move_to_start()


def turn_right():
    for i in range(3):
        turn_left()


def move_to_start():
    # turn around
    turn_left()
    turn_left()
    # move back to start
    for i in range(3):
        move()
    turn_right()
    move()
    # reorient to face right
    turn_right()


# There is no need to edit code beyond this point
if __name__ == "__main__":
    run_karel_program()
```

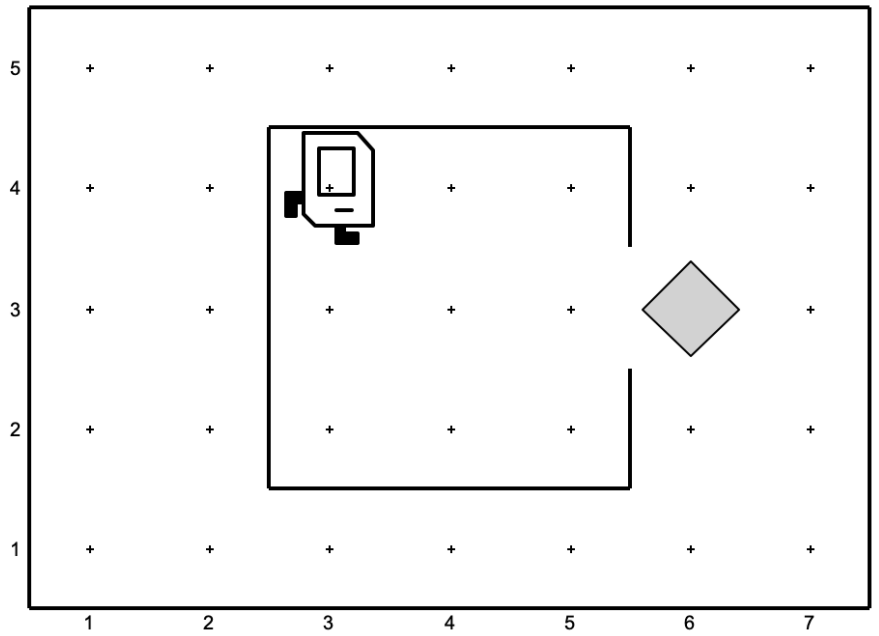
Multi-line comment

Main function

Our function

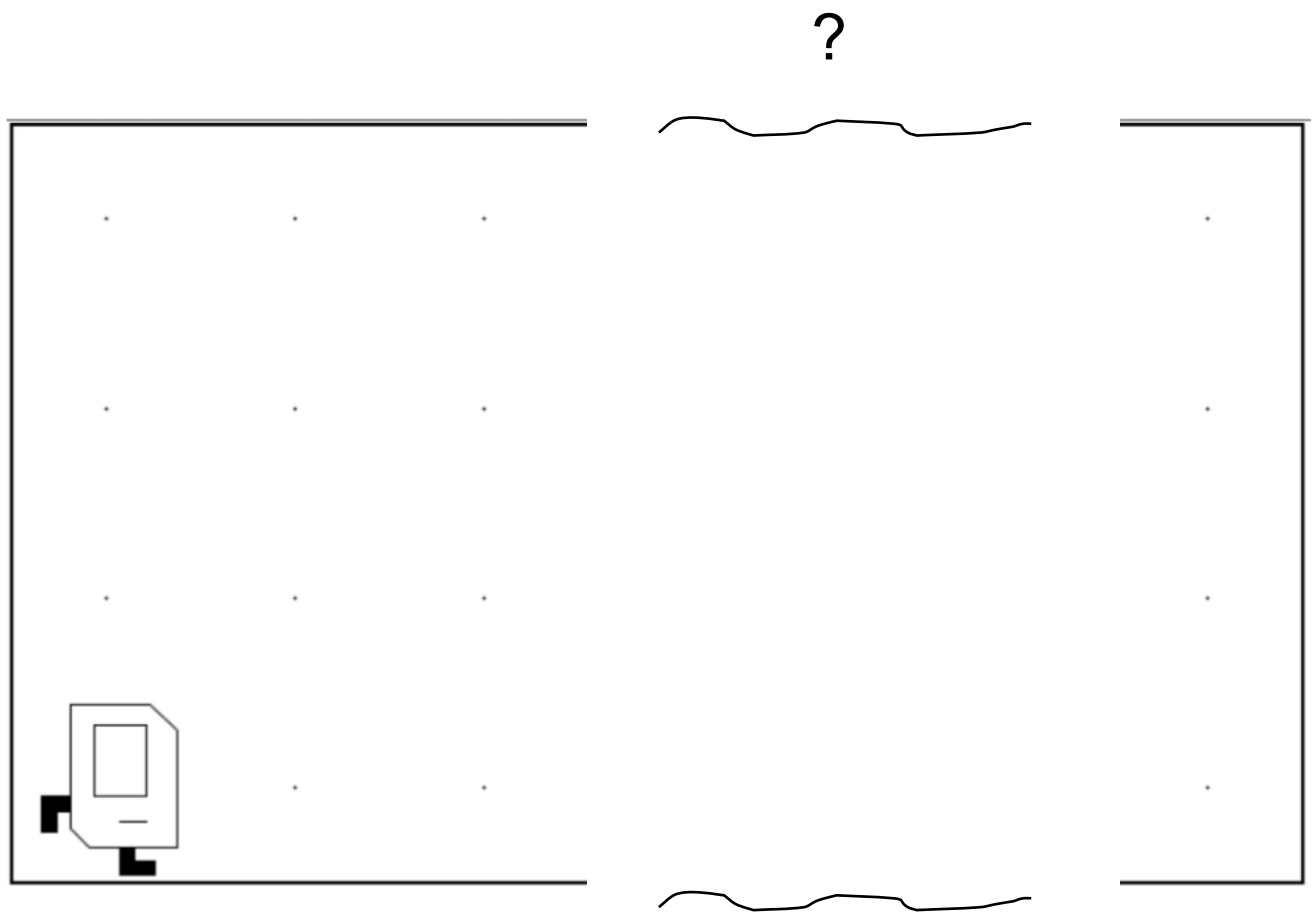
Single-line comment

As long as Karel knows the world, its size, where the beepers are, she can do anything.

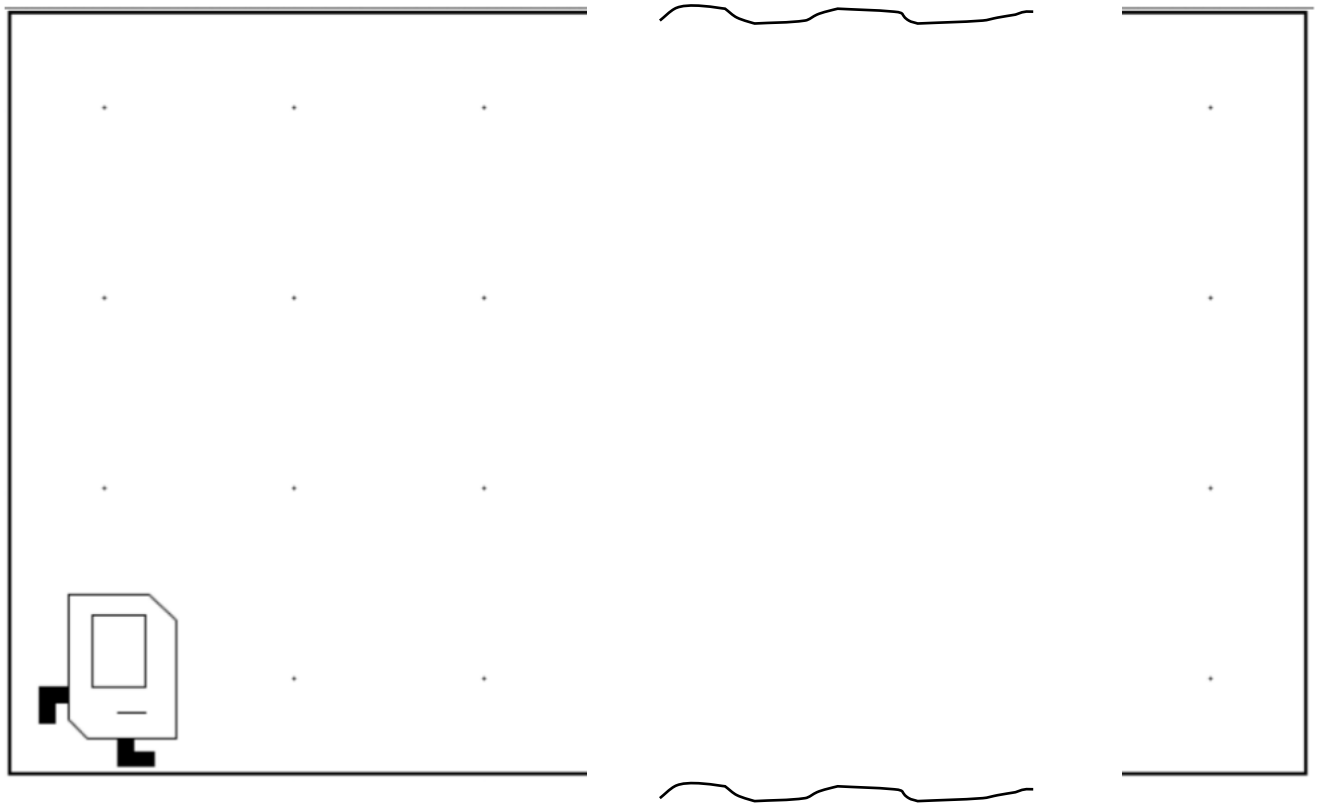


What if she does not know where the walls and beepers are?

Fill a street with beepers in a world of any size.



?



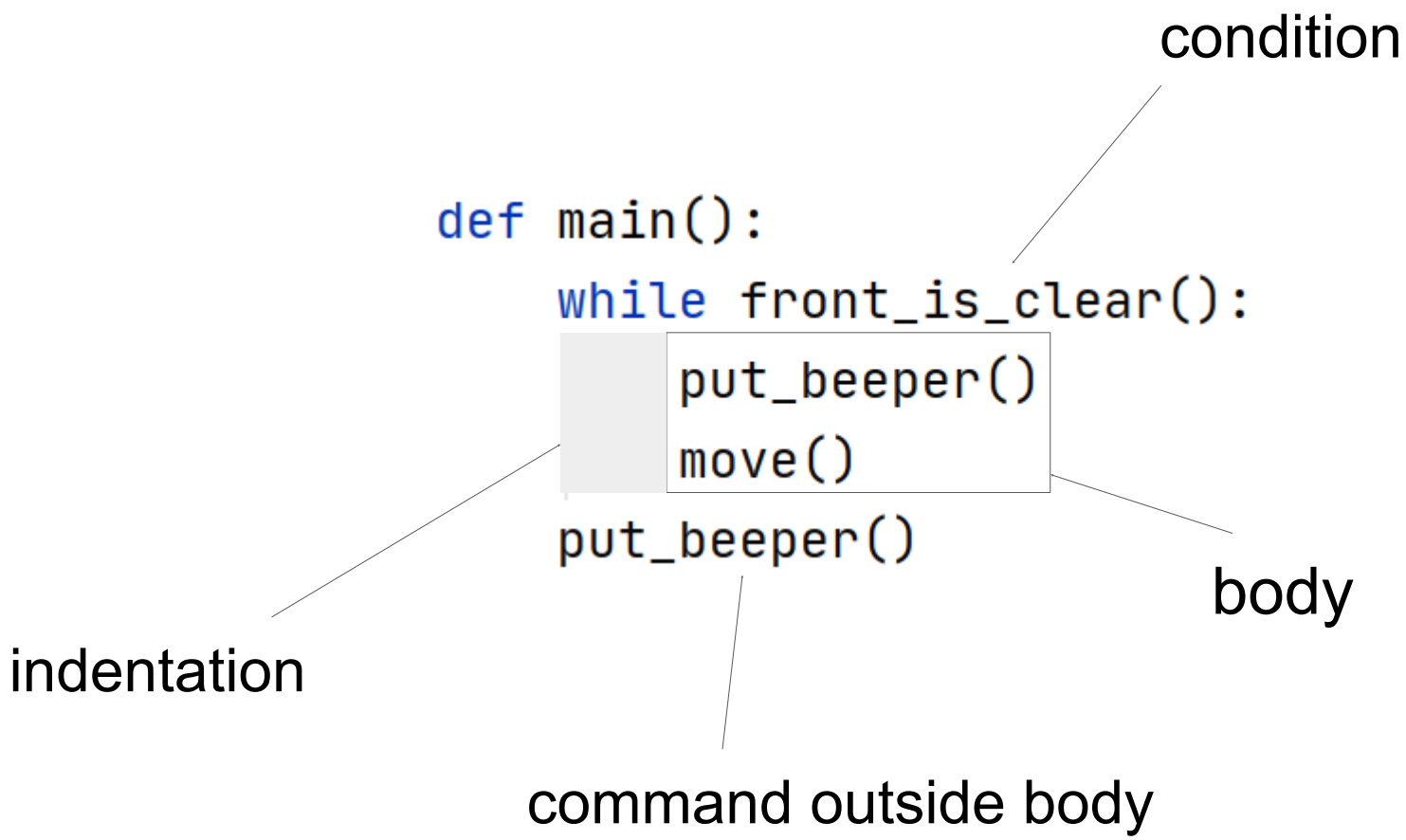
```
def main():
    for i in range(?):
        put_beeper()
        move()
```

We have an alternative for the **for loop**

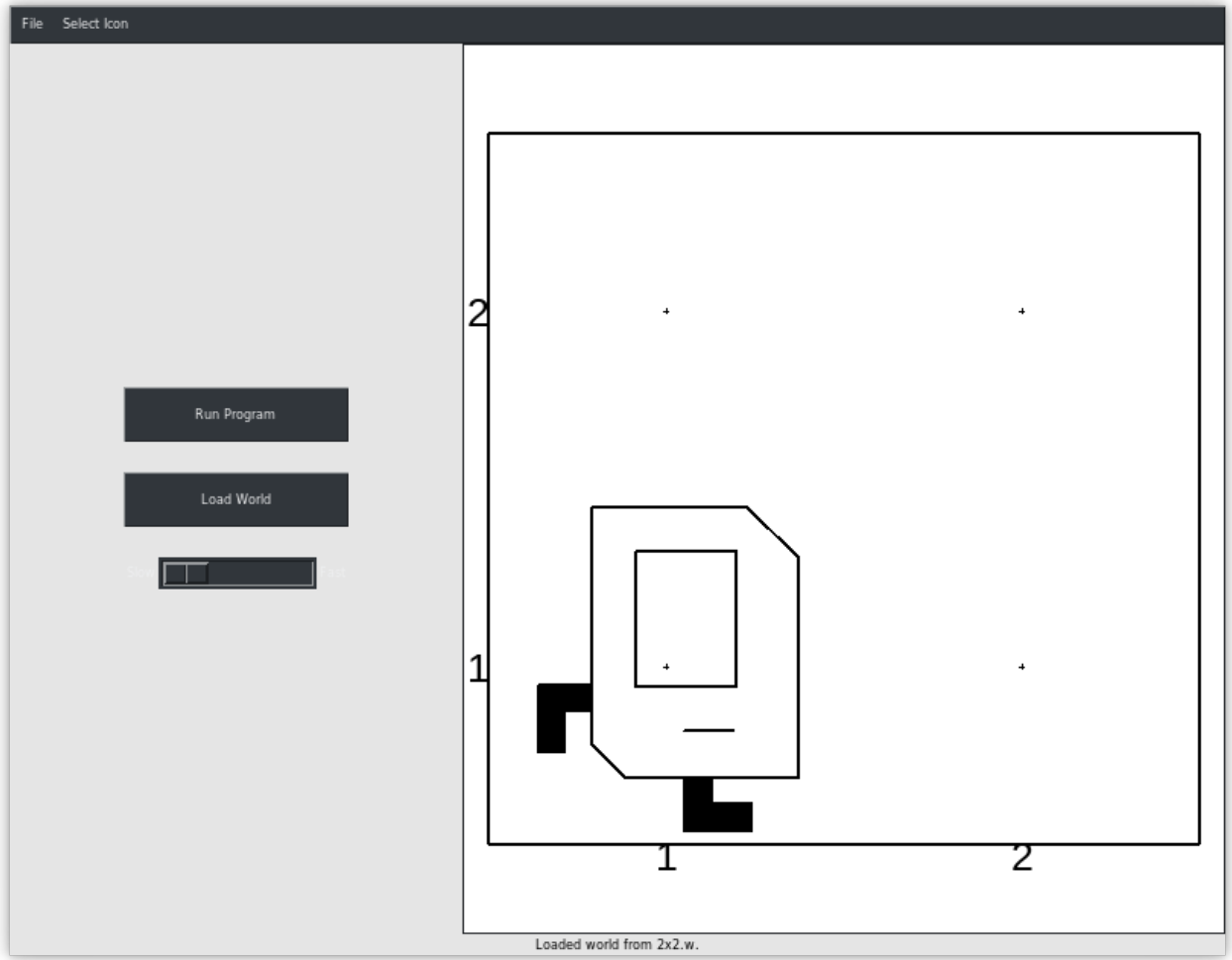
Karel can check a few things about the world

Test	*Opposite*	*What it checks*
`front_is_clear()`	`front_is_blocked()`	Is there a wall in front of Karel?
`left_is_clear()`	`left_is_blocked()`	Is there a wall to Karel's left?
`right_is_clear()`	`right_is_blocked()`	Is there a wall to Karel's right?
`beepers_present()`	`no_beepers_present()`	Are there beepers on this corner?
`beepers_in_bag()`	`no_beepers_in_bag()`	Any there beepers in Karel's bag?
`facing_north()`	`not_facing_north()`	Is Karel facing north?
`facing_east()`	`not_facing_east()`	Is Karel facing east?
`facing_south()`	`not_facing_south()`	Is Karel facing south?
`facing_west()`	`not_facing_west()`	Is Karel facing west?

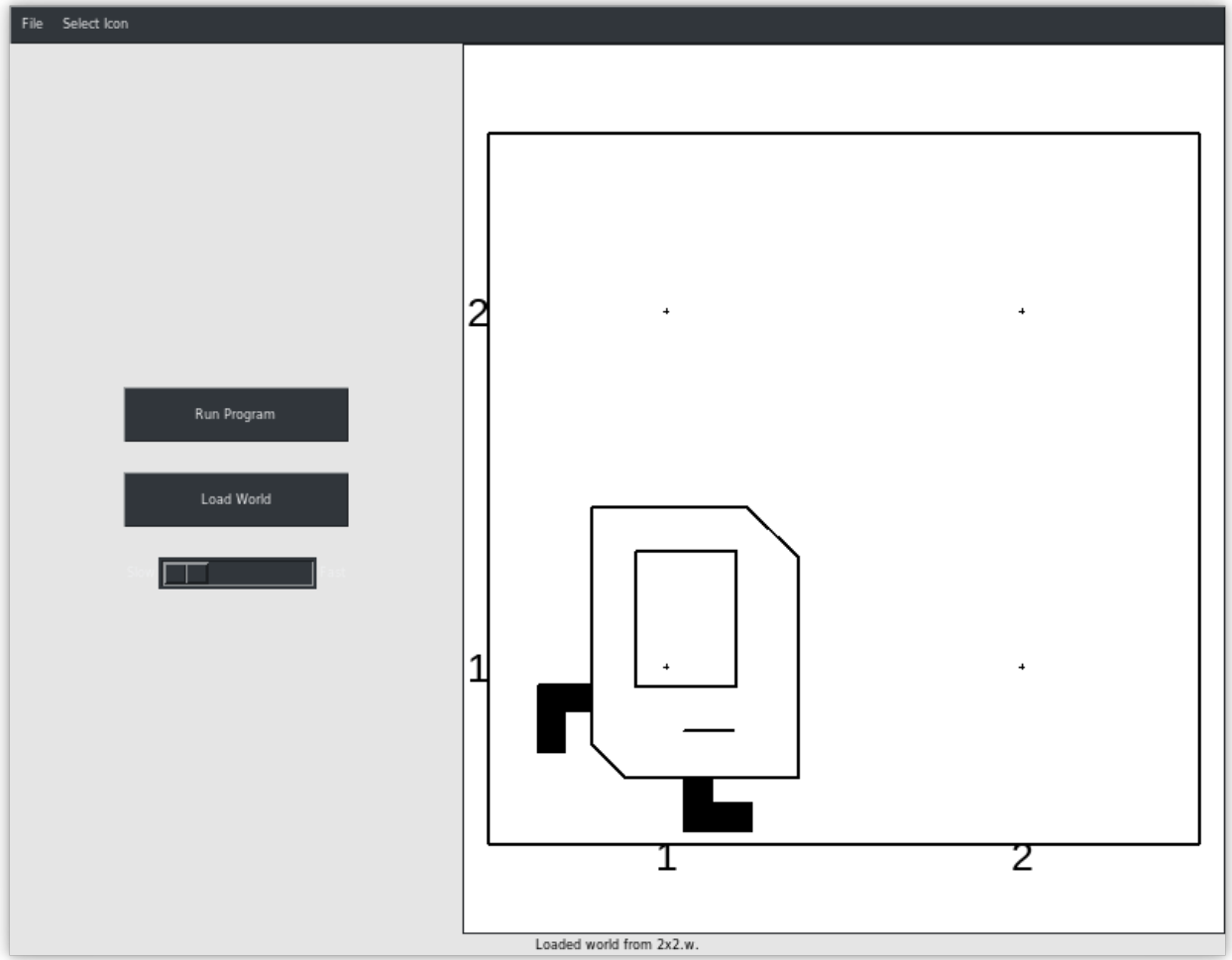
Maybe we can ask her to move as long as front is clear


condition
def main():
while front_is_clear():
put_beeper()
move()
put_beeper()
body
indentation
command outside body

```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

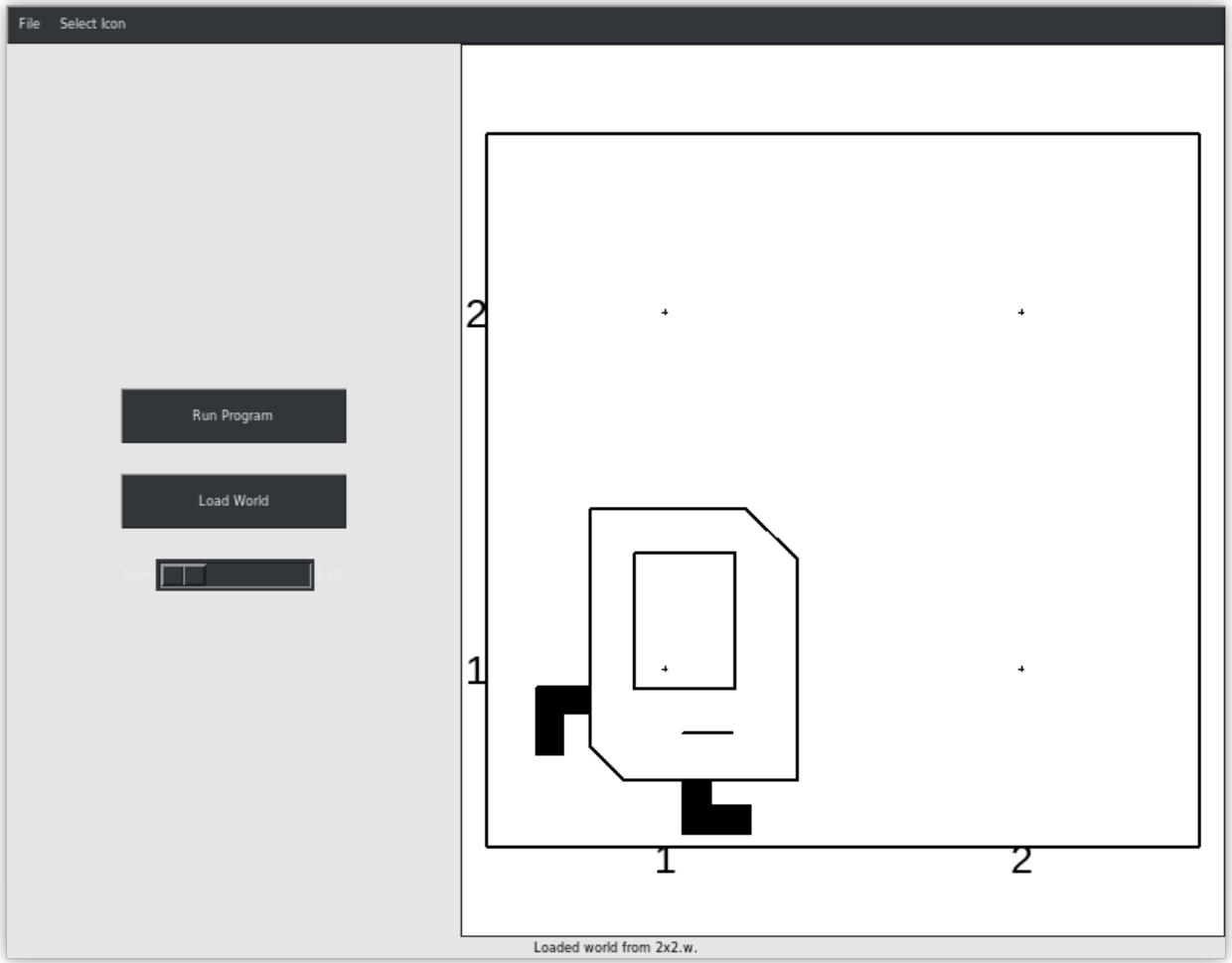


```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

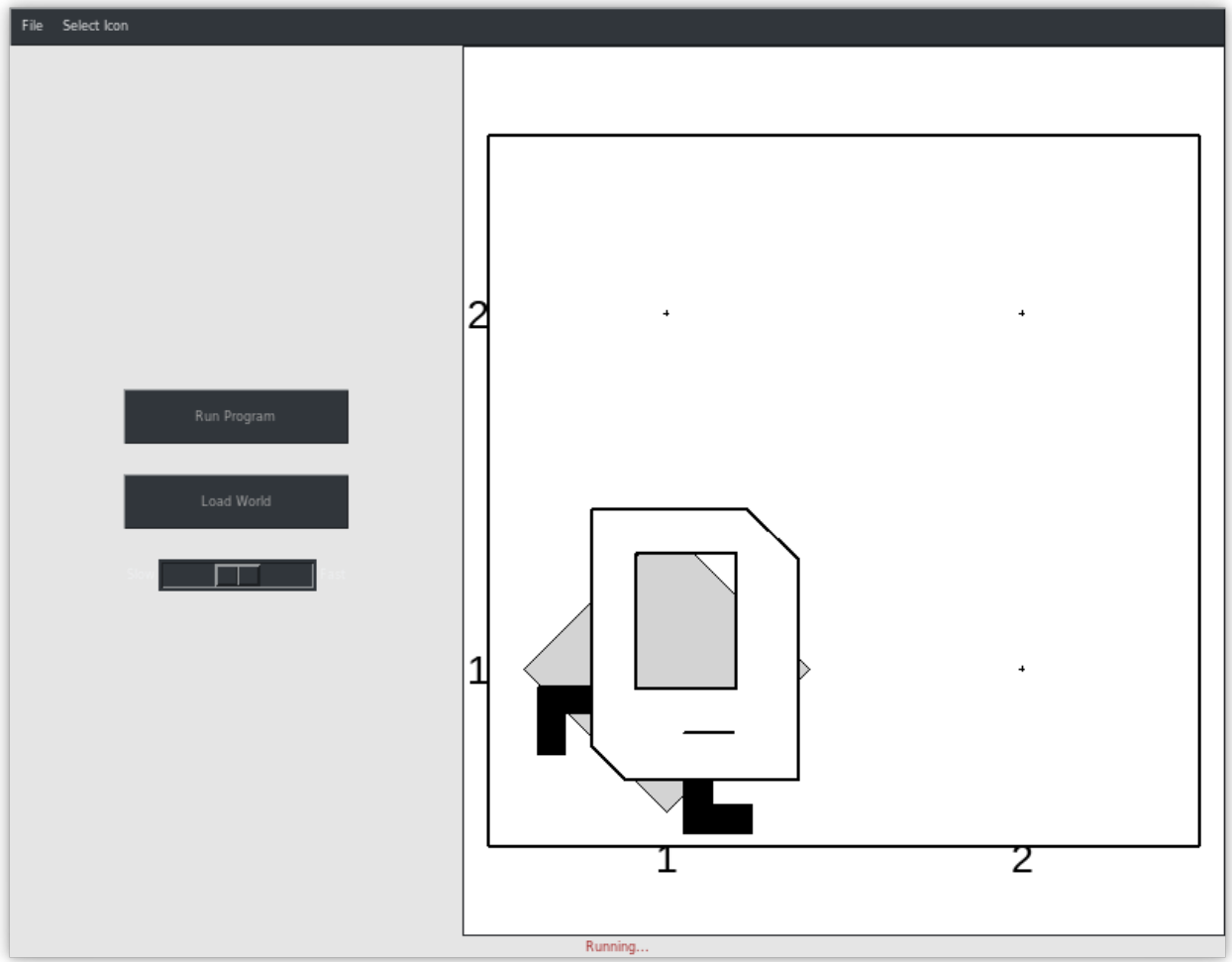


```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

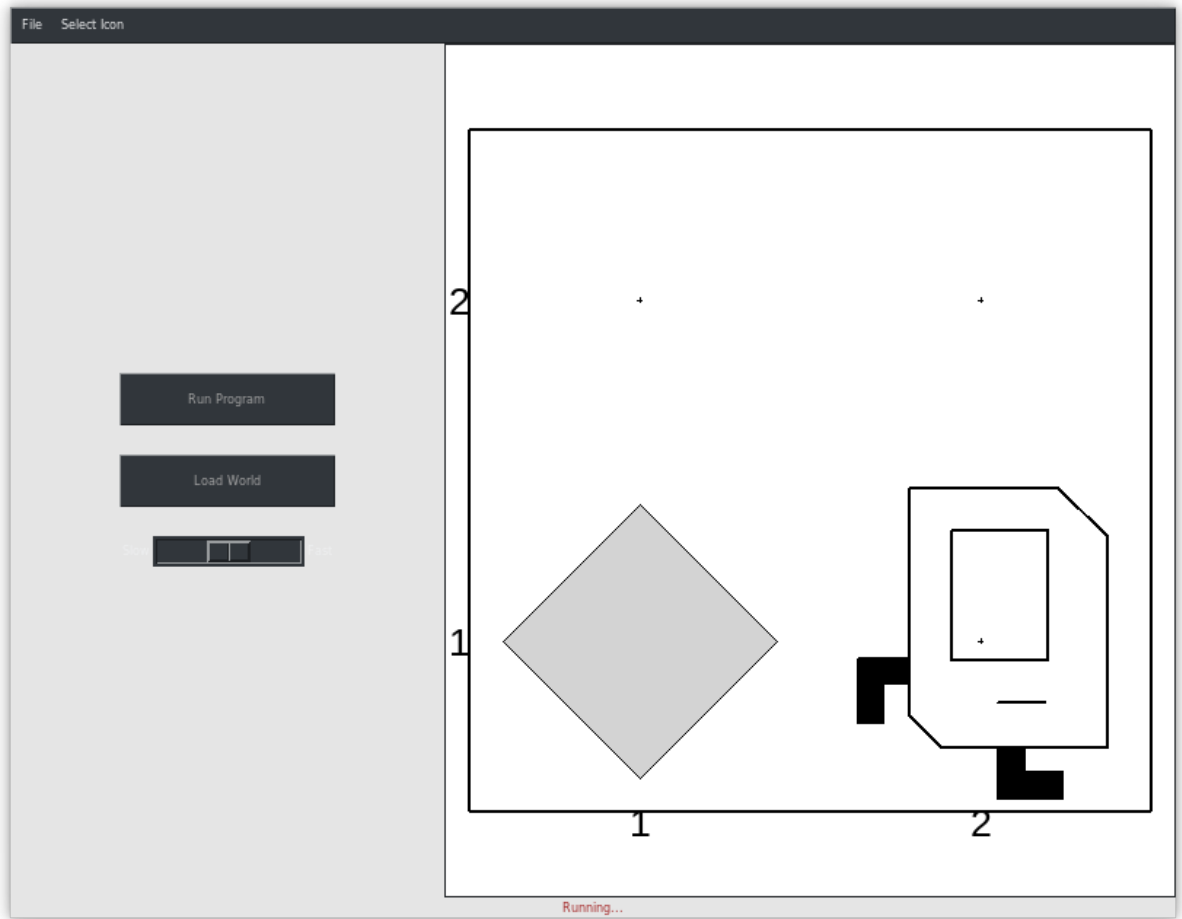
YES



```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

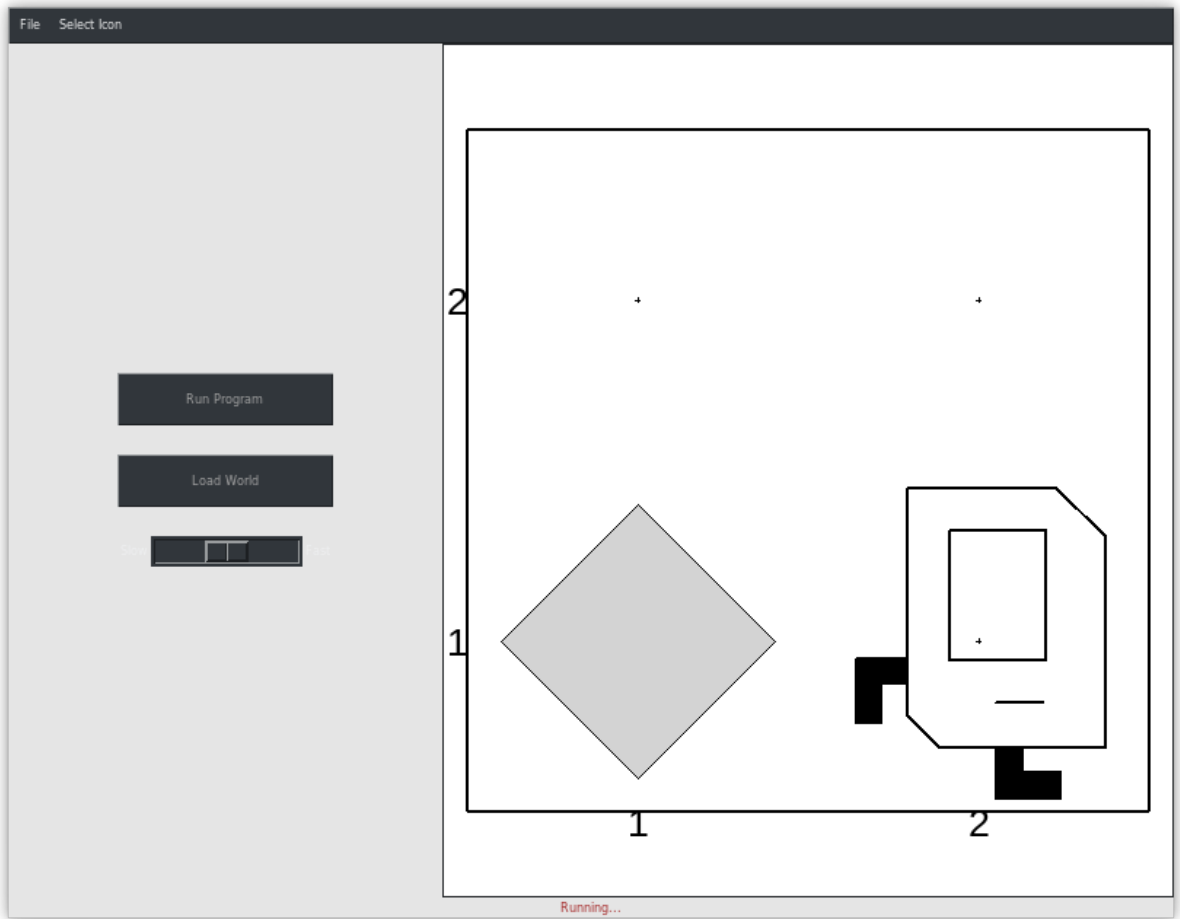


```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

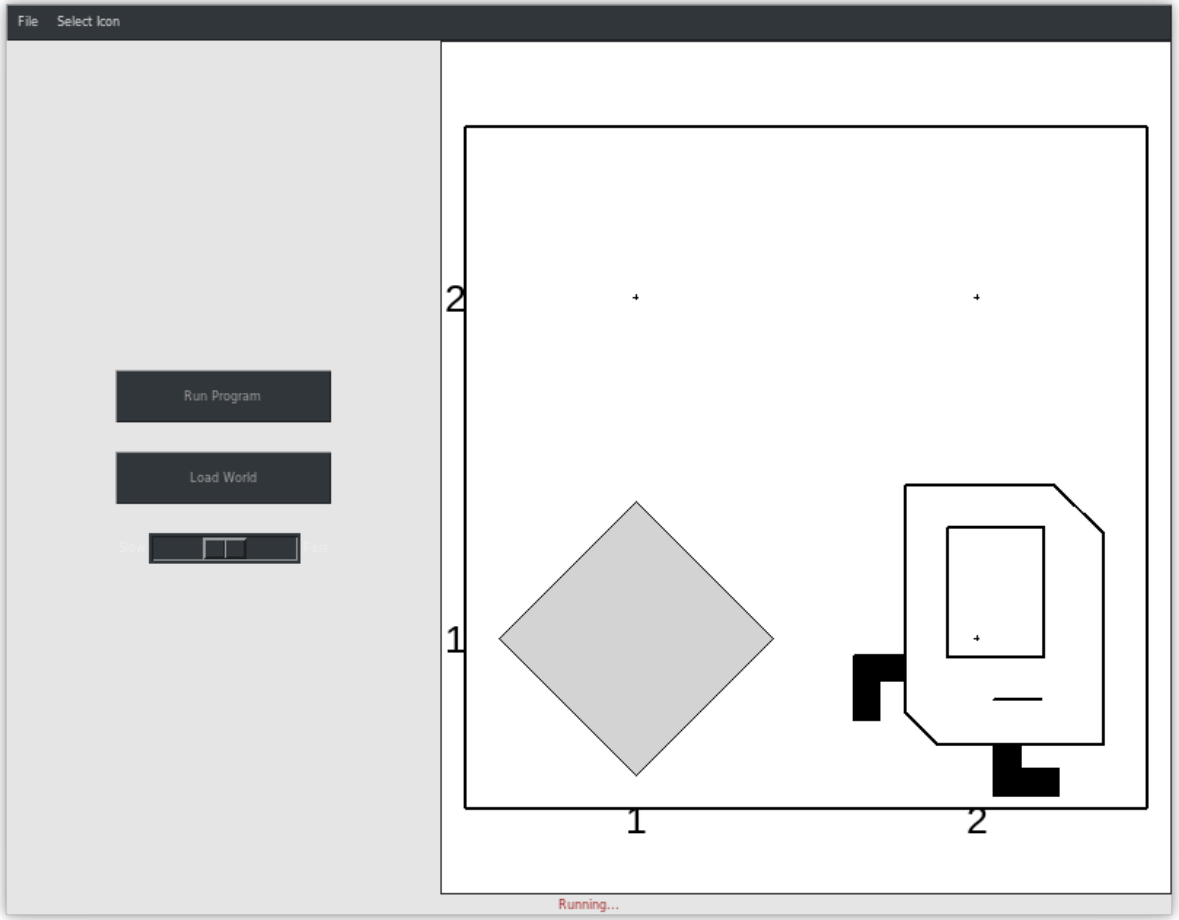


```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

NO

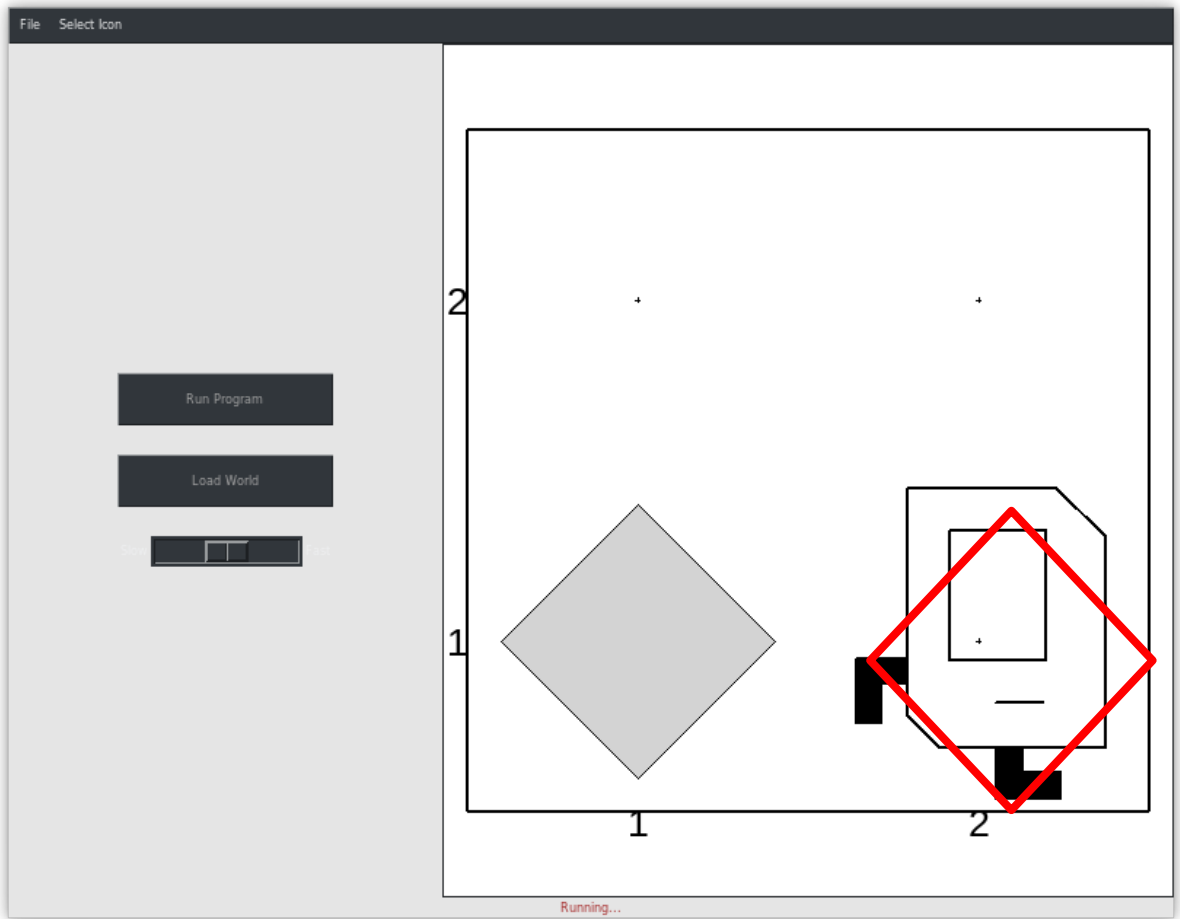


```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

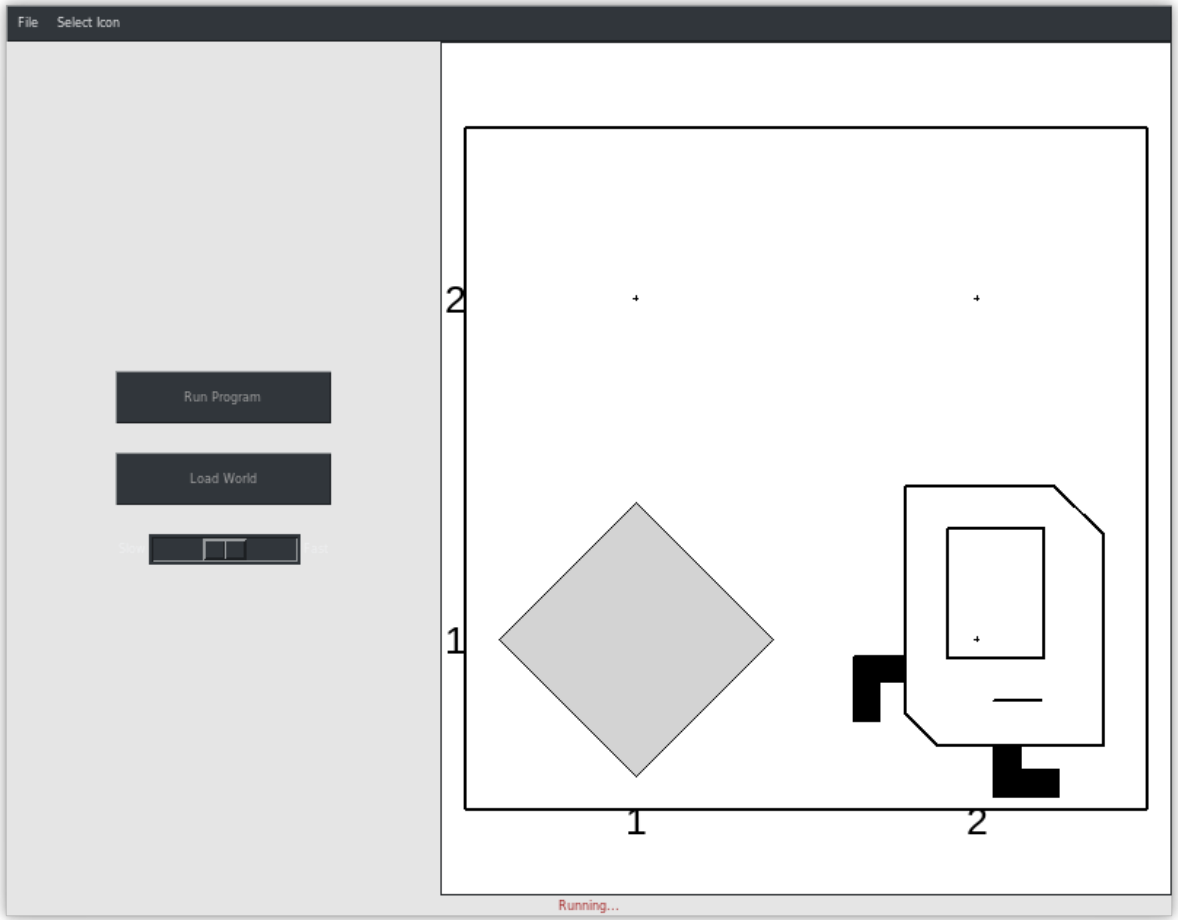


What if we run the loop once more?

```
def main():
    while front_is_clear():
        put_beeper()
        move()
```



```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

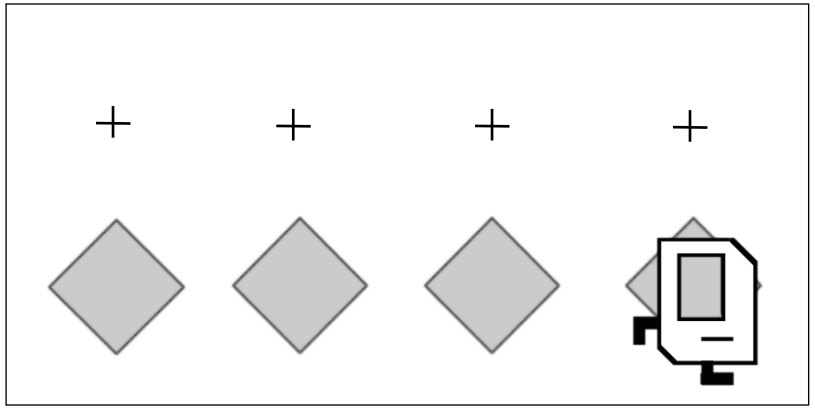


N beepers

N−1 moves

Fencepost

I want Karel to put down a row of beepers until it reaches a wall. How do I do this?

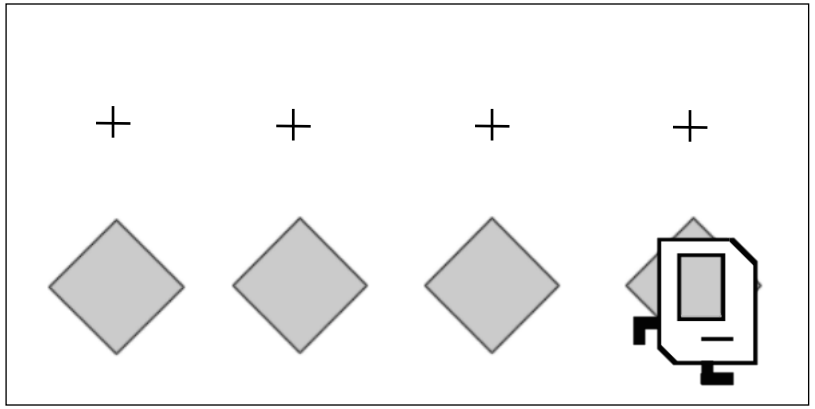


We must put 4 beepers but move 3 times!

```
put_beeper()
move()
put_beeper()
move()
put_beeper()
move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall. How do I do this?



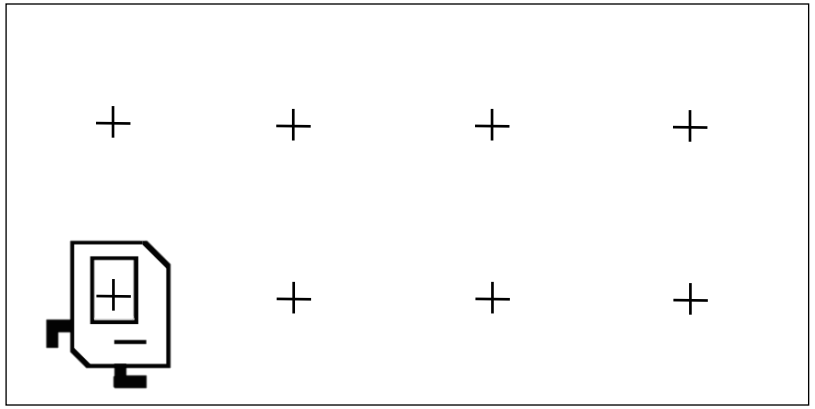
We must put 4 beepers but move 3 times!

```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

How do I do this?

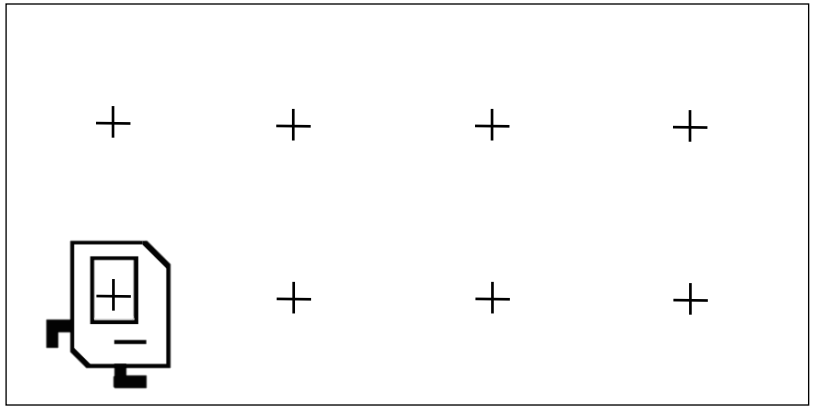


```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

How do I do this?

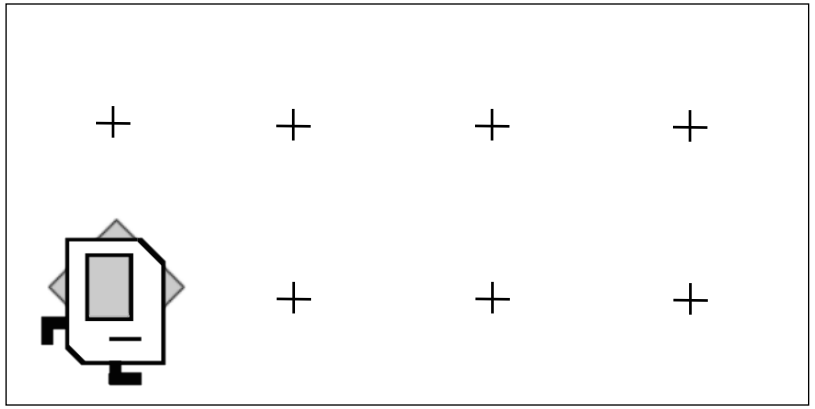


```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

How do I do this?

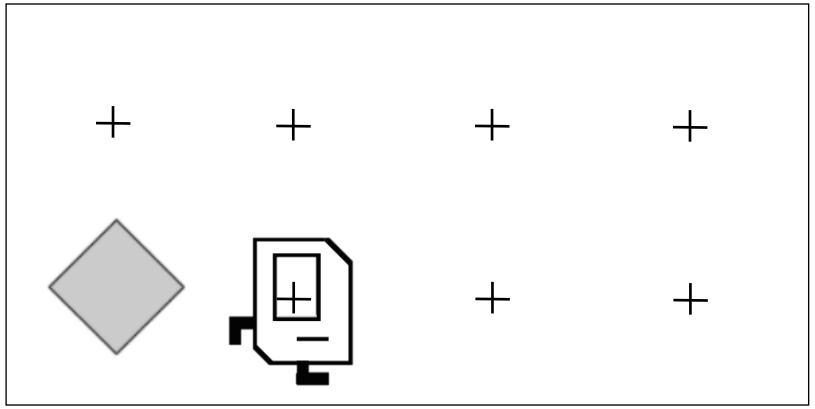


```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

How do I do this?

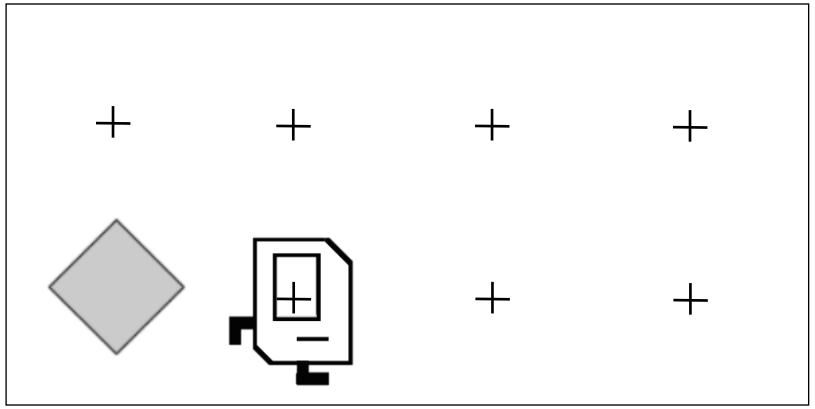


```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

How do I do this?

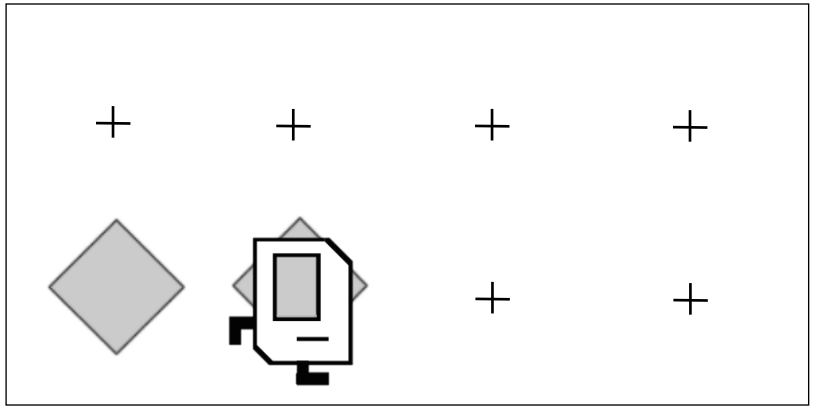


```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

How do I do this?

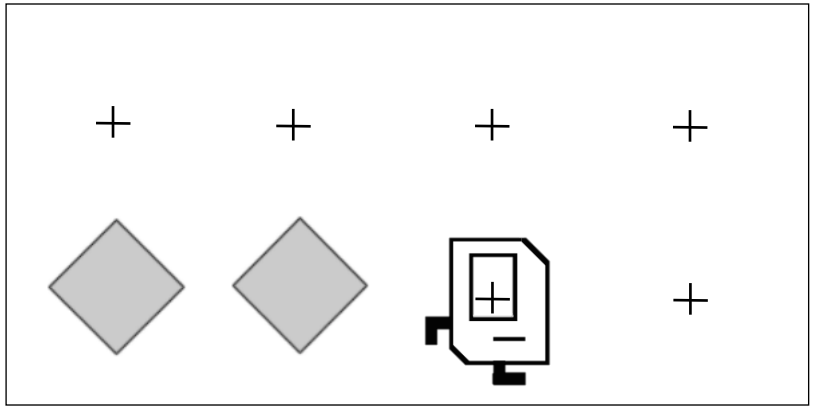


```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

How do I do this?

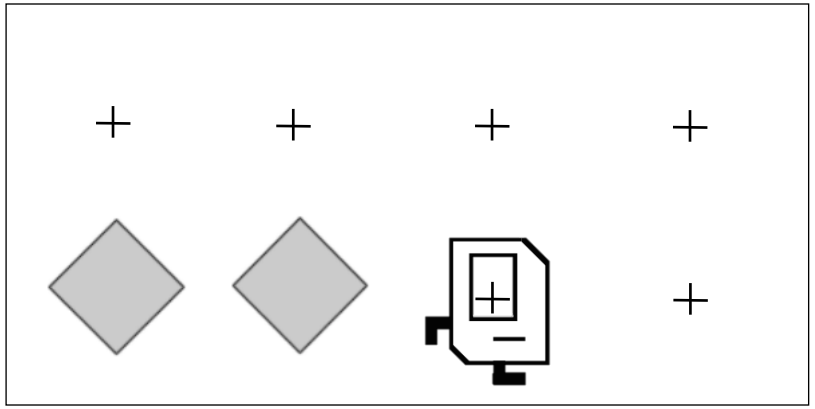


```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

How do I do this?

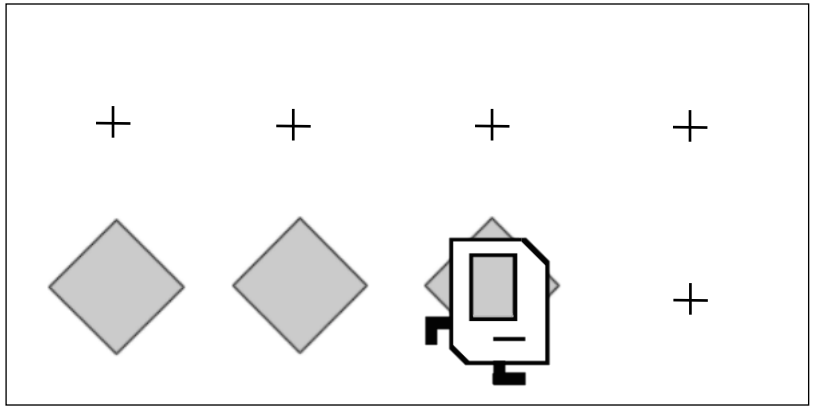


```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

How do I do this?

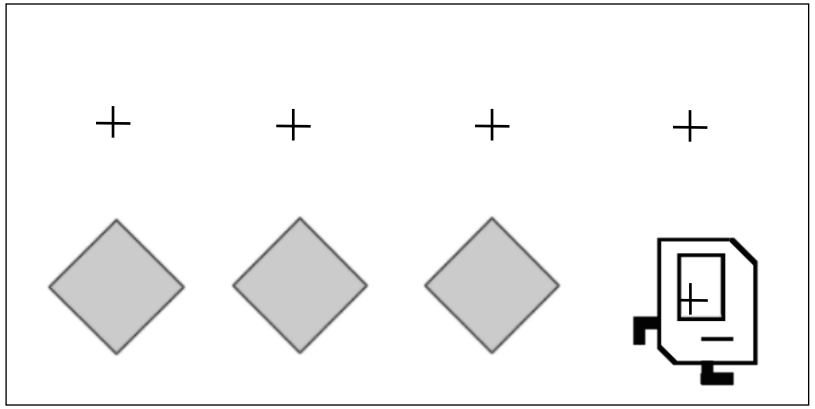


```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

How do I do this?

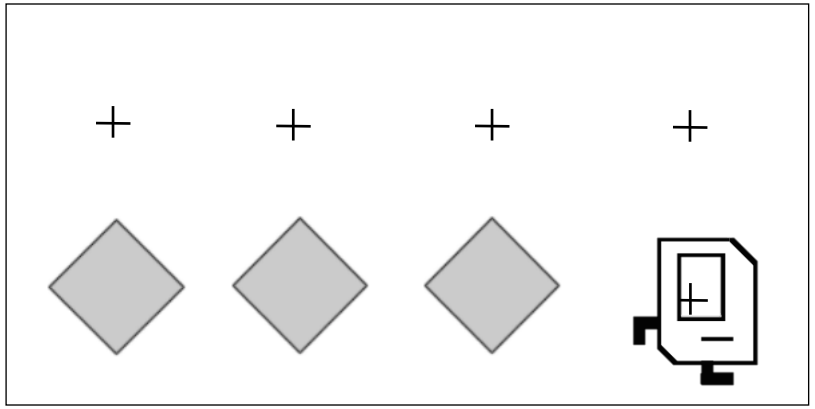


```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

How do I do this?

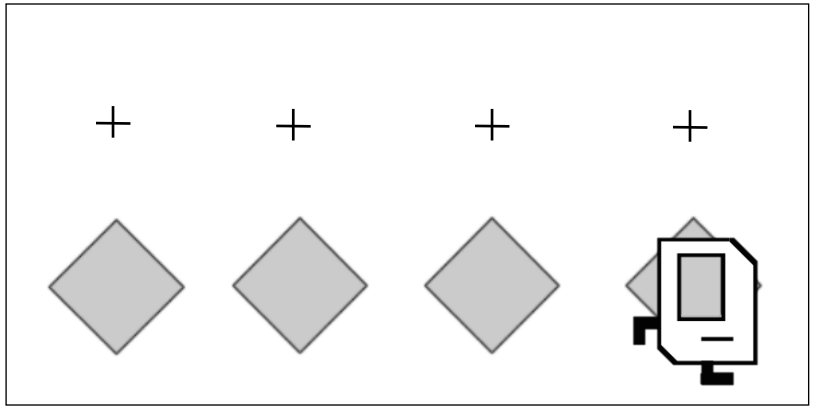


```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

How do I do this?

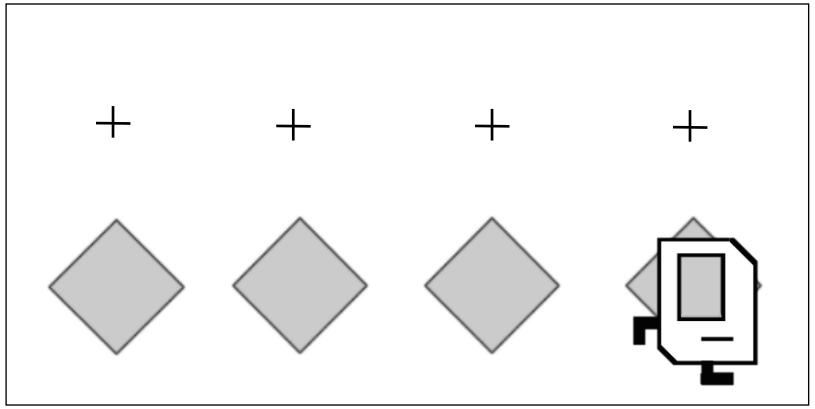


```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall.

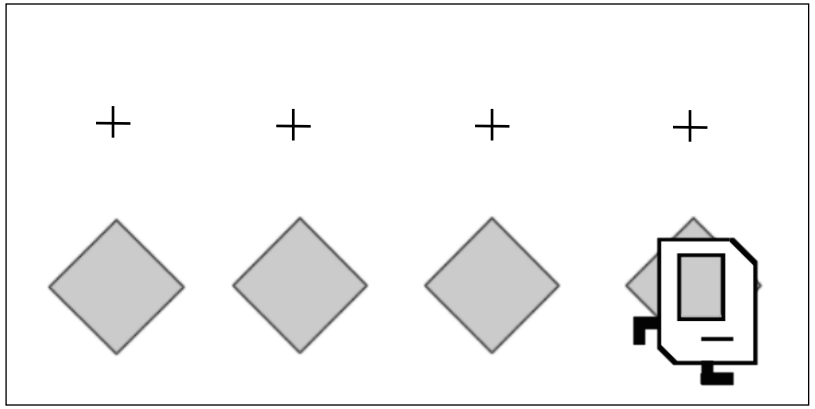
How do I do this?



```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall. How do I do this?

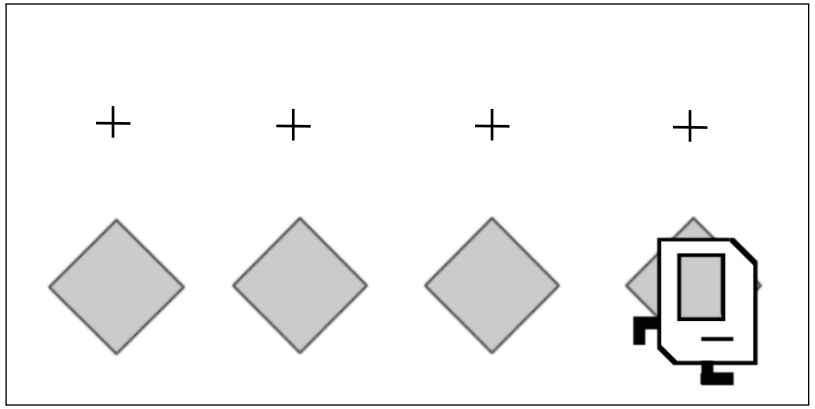


We must put N beepers but move N-1 times!

```
put_beeper()
move()
put_beeper()
move()
...
put_beeper()
```

Fencepost

I want Karel to put down a row of beepers until it reaches a wall. How do I do this?

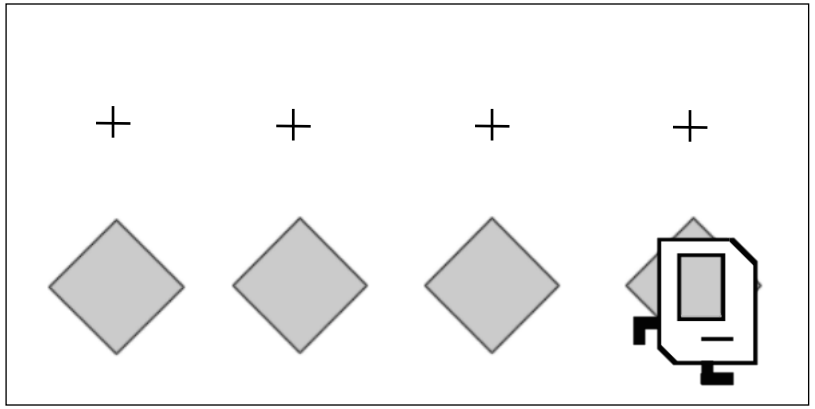


```
put_beeper()
move()
put_beeper()
move()
put_beeper()
move()
put_beeper()
```

Any suggestion for an alternative? A different way to group and loop

Fencepost

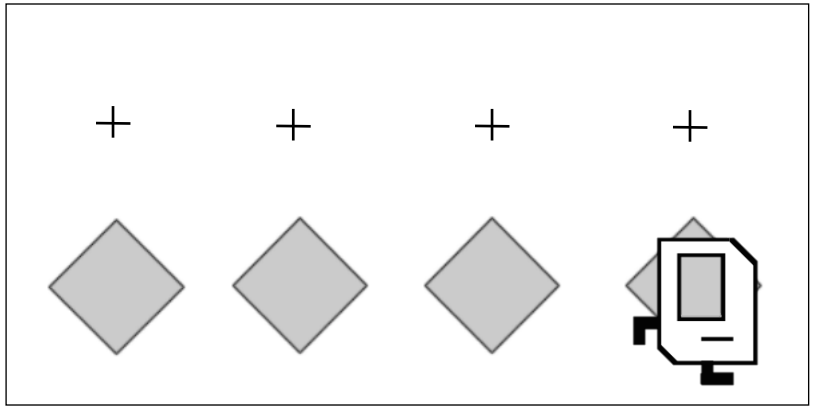
I want Karel to put down a row of beepers until it reaches a wall. How do I do this?



```
put_beeper()
move()
put_beeper()
move()
put_beeper()
move()
put_beeper()
```

Fencepost

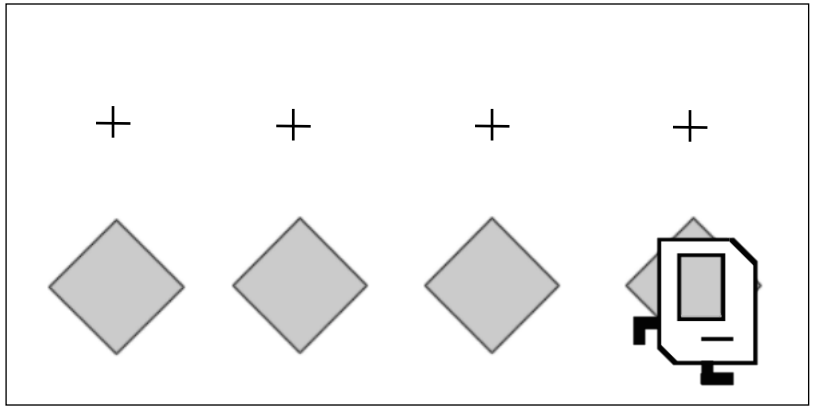
I want Karel to put down a row of beepers until it reaches a wall. How do I do this?



```
put_beeper()
while front_is_clear():
    move()
    put_beeper()
```

Fencepost

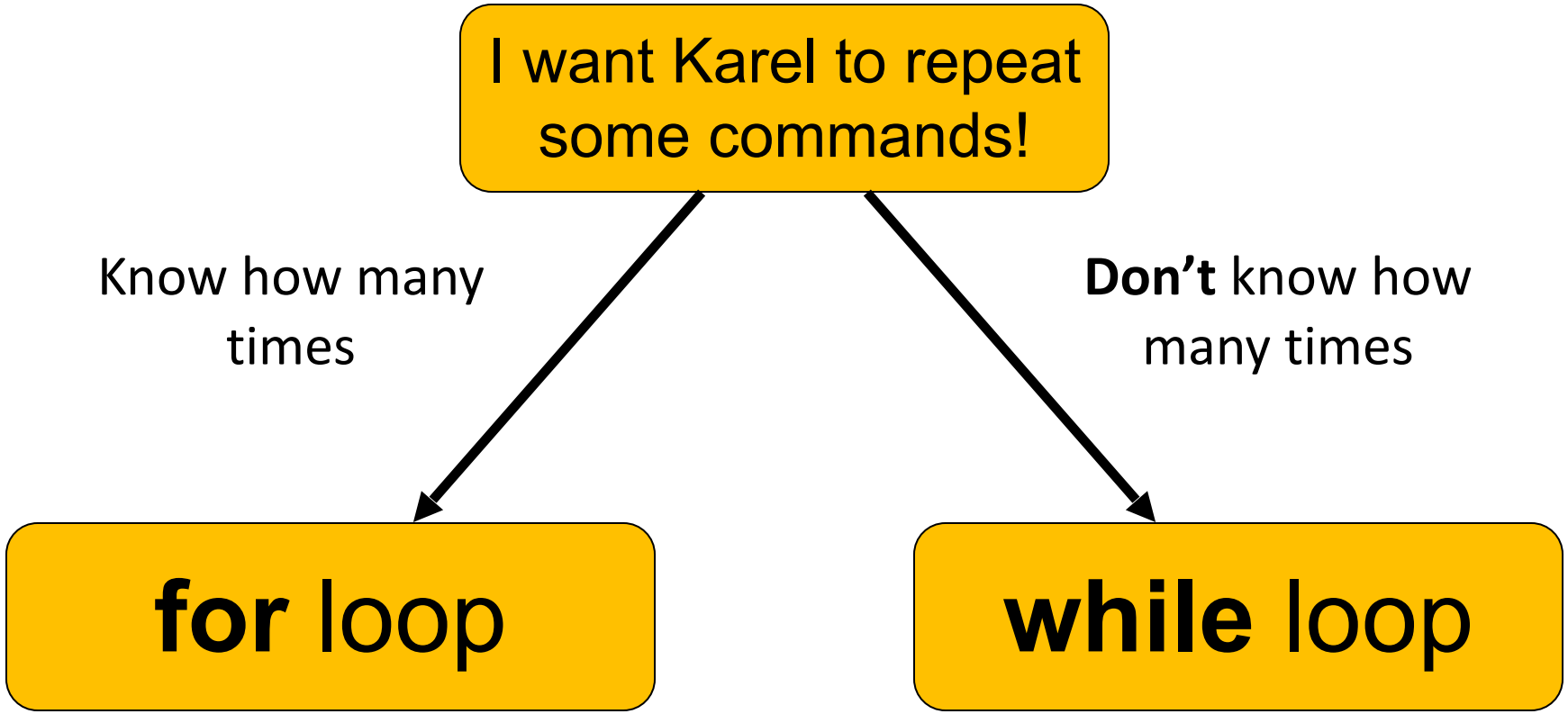
I want Karel to put down a row of beepers until it reaches a wall. How do I do this?



We must put N beepers but move N-1 times!

```
put_beeper()
move()
put_beeper()
move()
put_beeper()
...
move()
put_beeper()
```

Loops Overview



Fencepost Structure

The fencepost structure is useful when you want to loop a set of statements but do one part of that set 1 *additional* time.

```
put_beeper()                # post
while front_is_clear():
    move()                  # fence
    put_beeper()            # post


# or...
while front_is_clear():
    put_beeper()            # post
    move()                  # fence
put_beeper()                # post
```

Champion’s dance

```
def main():
    while front_is_clear():
        move()
        turn_left()
```

Reminder!

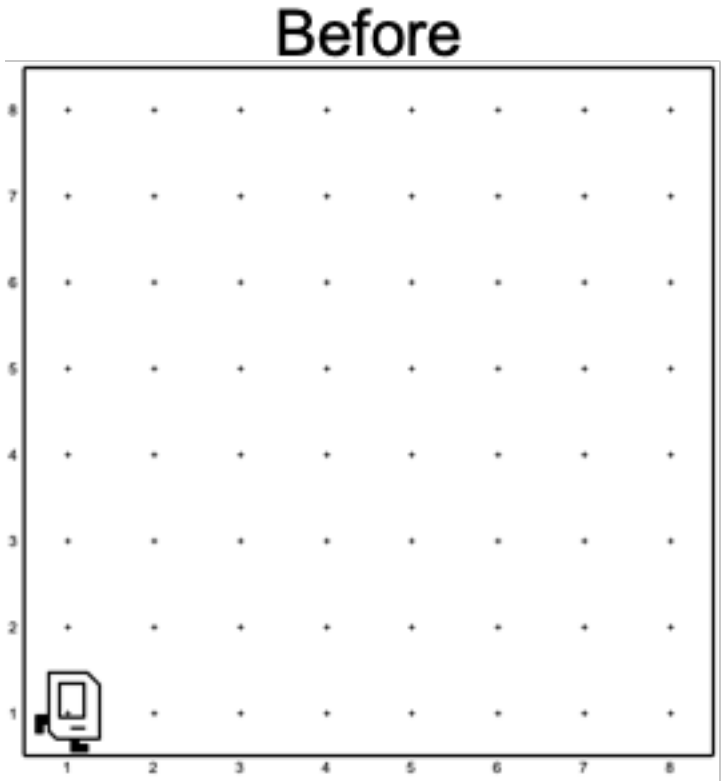
Karel is *very* picky about indentation.

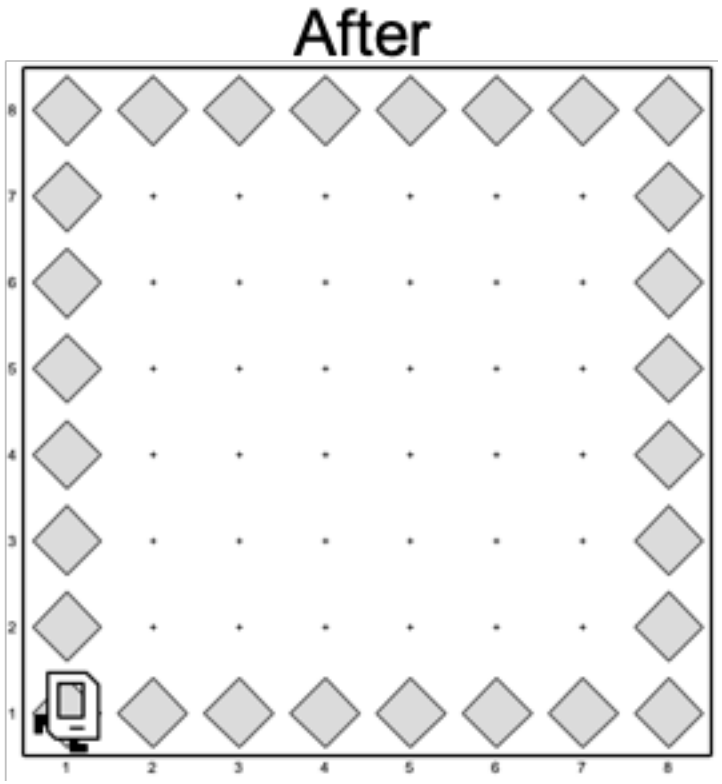
Make sure to indent a code block 1 level further when you:

- Define a new Karel command
- Write a for loop
- Write a while loop

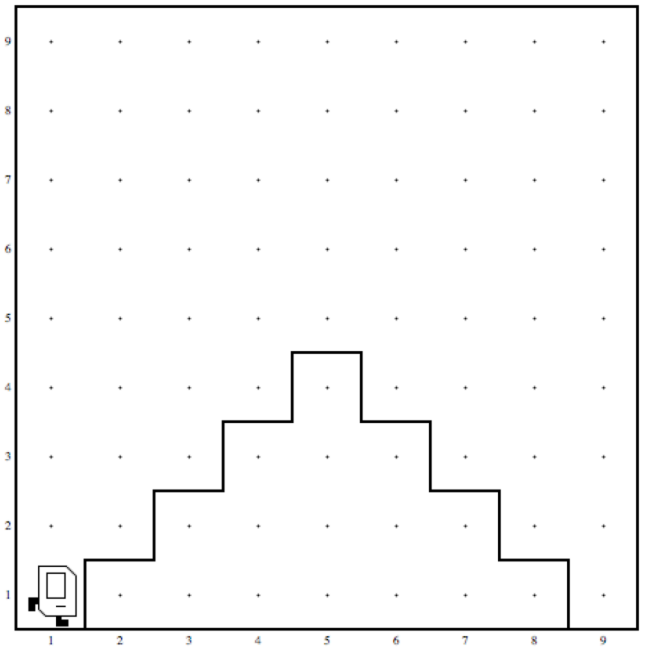
You may nest these. Make sure you keep track of your indentation!

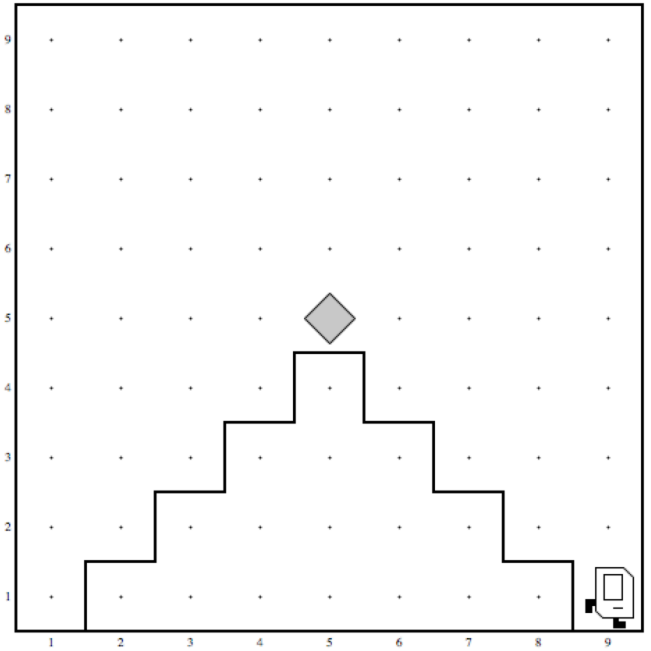
Projects for today: Place Square

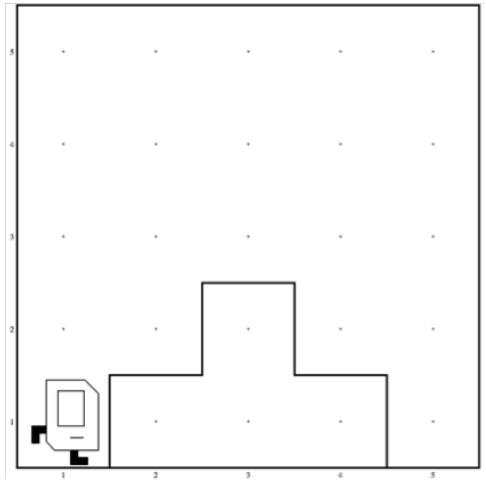


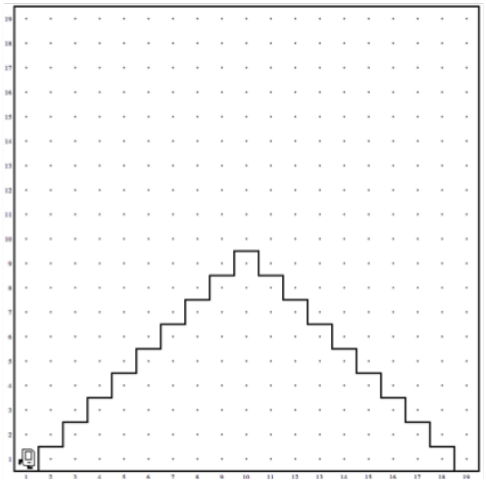


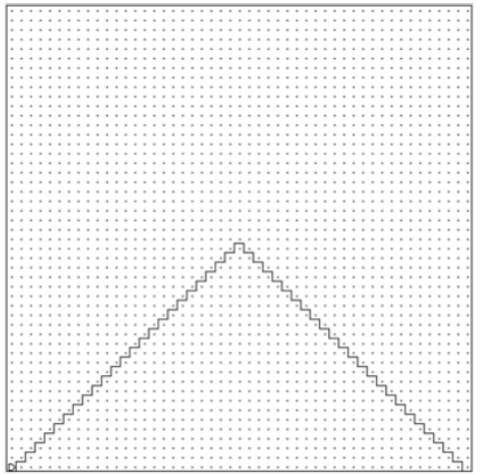
Projects for today: Mountain Karel











Schedule

Timezone: Turkey

No panic, first days may be a bit chaotic.