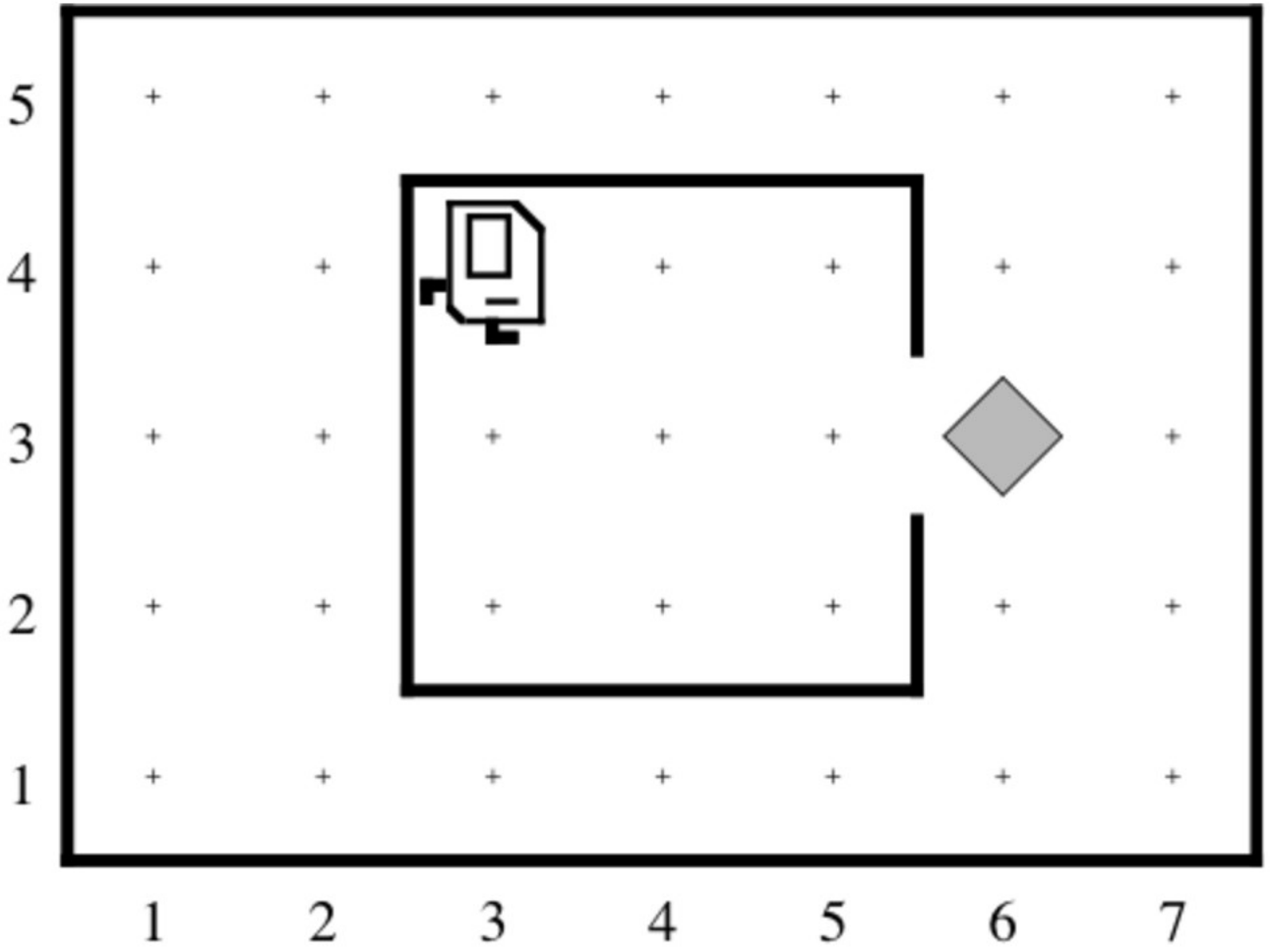
Control Flow

Before we start:

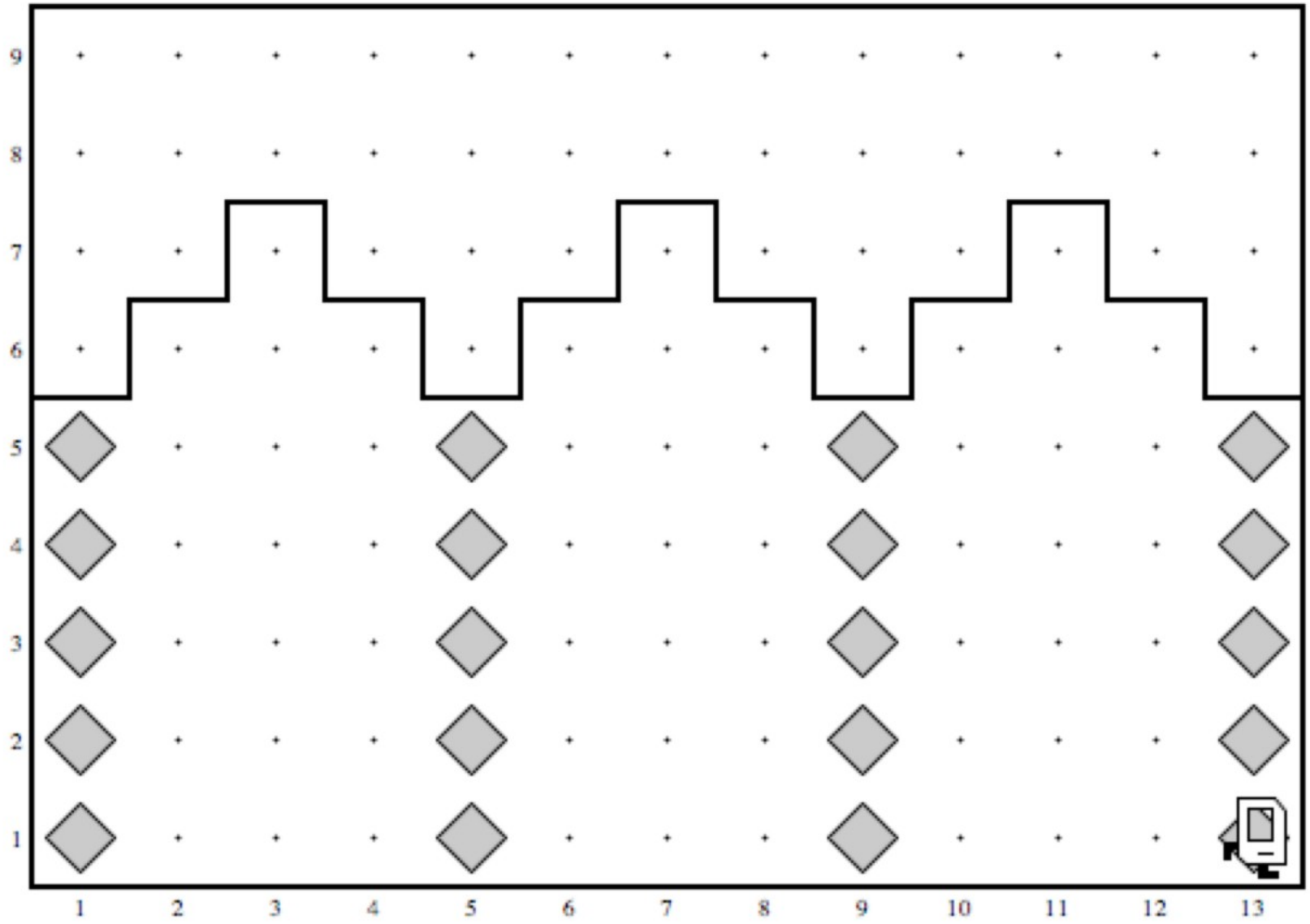
Any questions on Karel?

Collect Newspaper



Any trouble implementing this task?

Have an idea how to make Karel build Efes?



That’s fun!...

…but a robot should get prepared for the dangers of the world


2
1

…awareness about the environment would save lives

…awareness about the environment would save lives

Karel should know how to:

…count the steps

…awareness about the environment would save lives

Karel should know how to :

…count the steps
…check what’s around

…awareness about the environment would save lives

Karel should know how to :

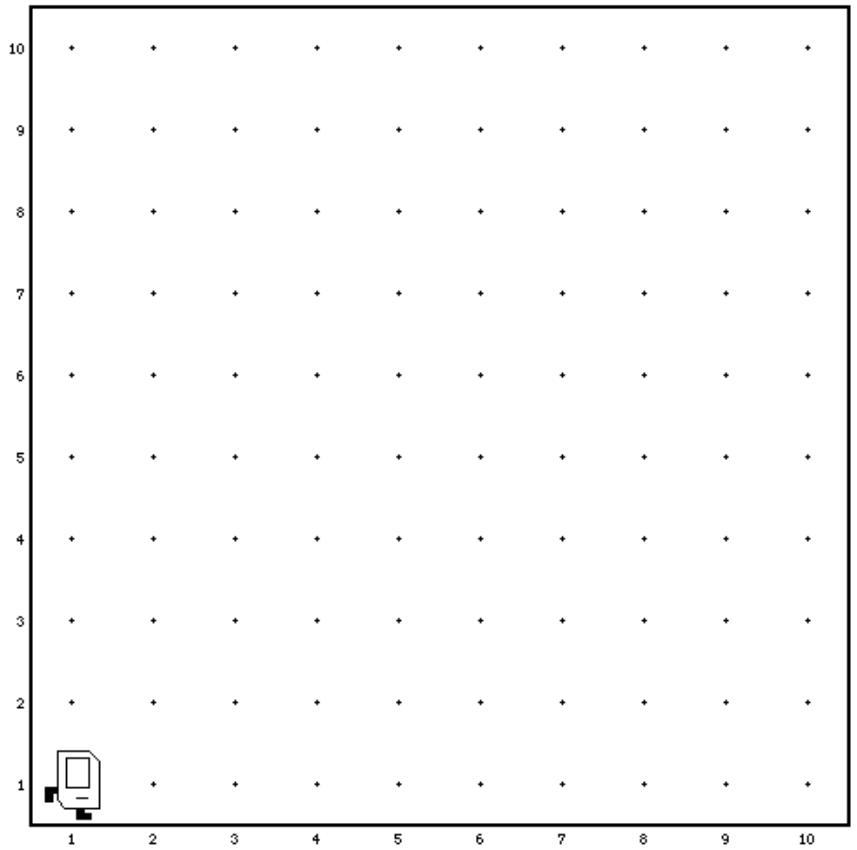
…count the steps
…check what’s around
…adjust to the environment

But first:

Who likes dancing here?

Yanar Döner Karel

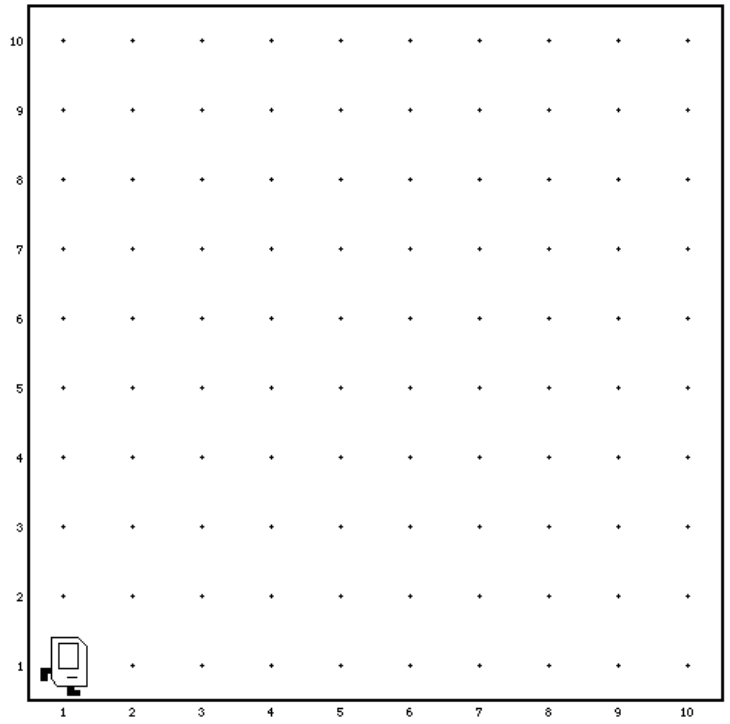
```
import stanford.karel.*;
public class YanarDoner extends SuperKarel {

    public void run() {
        //Dance
        move();move();turnLeft();
        move();move();turnLeft();
        move();move();turnLeft();
        move();move();turnLeft();
    }
}
```



Doing it once is not real dance, what about some more ‘loops’

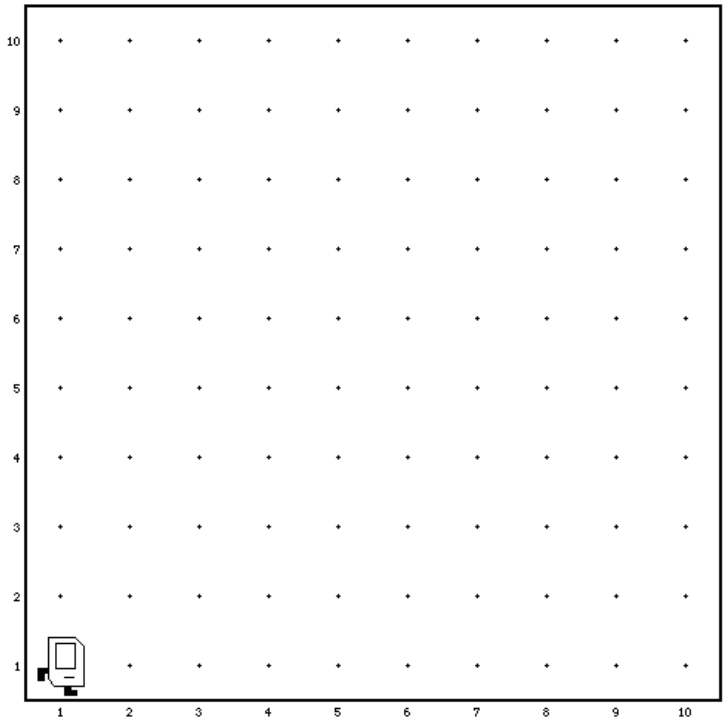
Yanar Döner Karel

```
import stanford.karel.*;
public class YanarDoner extends SuperKarel {

    public void run() {
        //Dance
        for(int i=0;i<4;i++) {
            move();move();turnLeft();
            move();move();turnLeft();
            move();move();turnLeft();
            move();move();turnLeft();
        }
    }
}
```



Yanar Döner Karel

```
import stanford.karel.*;
public class YanarDoner extends SuperKarel {

    public void run() {
        //Dance
        for(int i=0;i<4;i++) {
            for(int k=0;k<4;k++) {
                move();move();turnLeft();
            }
        }
    }
}
```



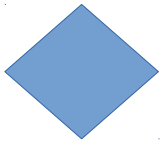
Understanding for loops

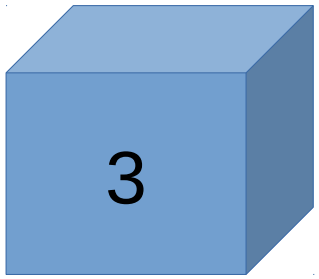
```
for(int numTickets = 3; numTickets > 0; numTickets--) {
      dance();
}
```

Initialization: `int numTickets = 3;`

Condition check: `numTickets > 0;`

The loop





Understanding for loops

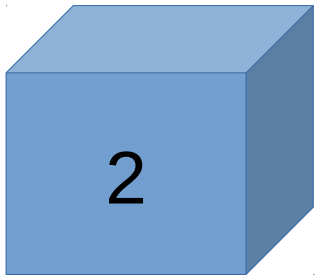
```
for(int numTickets = 3; numTickets > 0; numTickets--) {
      dance();
}
```

Initialization: `int numTickets = 3;`

Condition check: `numTickets > 0;`

The loop

Update/decrement: `numTickets--`



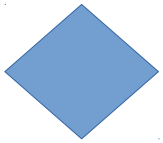
Understanding for loops

```
for(int numTickets = 3; numTickets > 0; numTickets--) {
      dance();
}
```

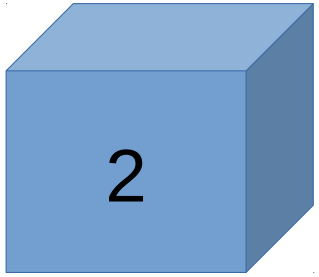
Initialization: `int numTickets = 3;`

Condition check: `numTickets > 0;`

The loop



numTickets



Understanding for loops

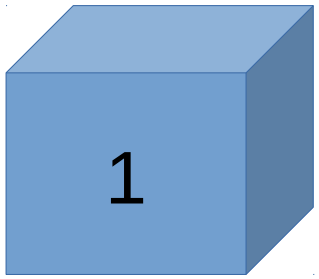
```
for(int numTickets = 3; numTickets > 0; numTickets--) {
      dance();
}
```

Initialization: `int numTickets = 3;`

Condition check: `numTickets > 0;`

The loop

Update/decrement: `numTickets--`

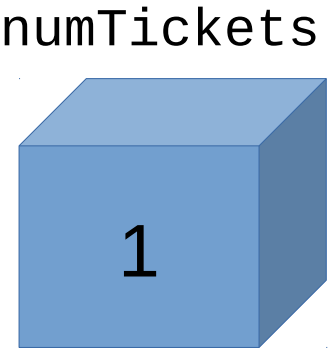


Understanding for loops

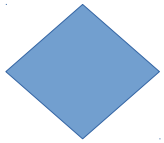
```
for(int numTickets = 3; numTickets > 0; numTickets--) {
      dance();
}
```

Initialization: `int numTickets = 3;`

Condition check: `numTickets > 0;`



The loop



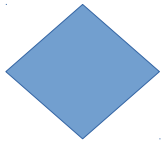
Understanding for loops

```
for(int numTickets = 3; numTickets > 0; numTickets--) {
      dance();
}
```

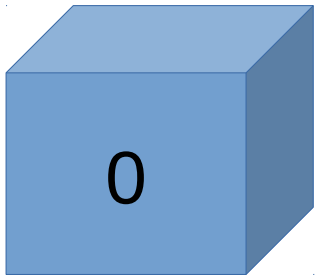
Initialization: `int numTickets = 3;`

Condition check: `numTickets > 0;`

The loop



Update/decrement: `numTickets--`

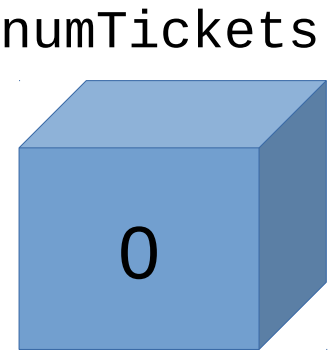


Understanding for loops

```
for(int numTickets = 3; numTickets > 0; numTickets--) {
      dance();
}
```

Initialization: `int numTickets = 3;`

Condition check: `numTickets > 0;`



END OF PARTY

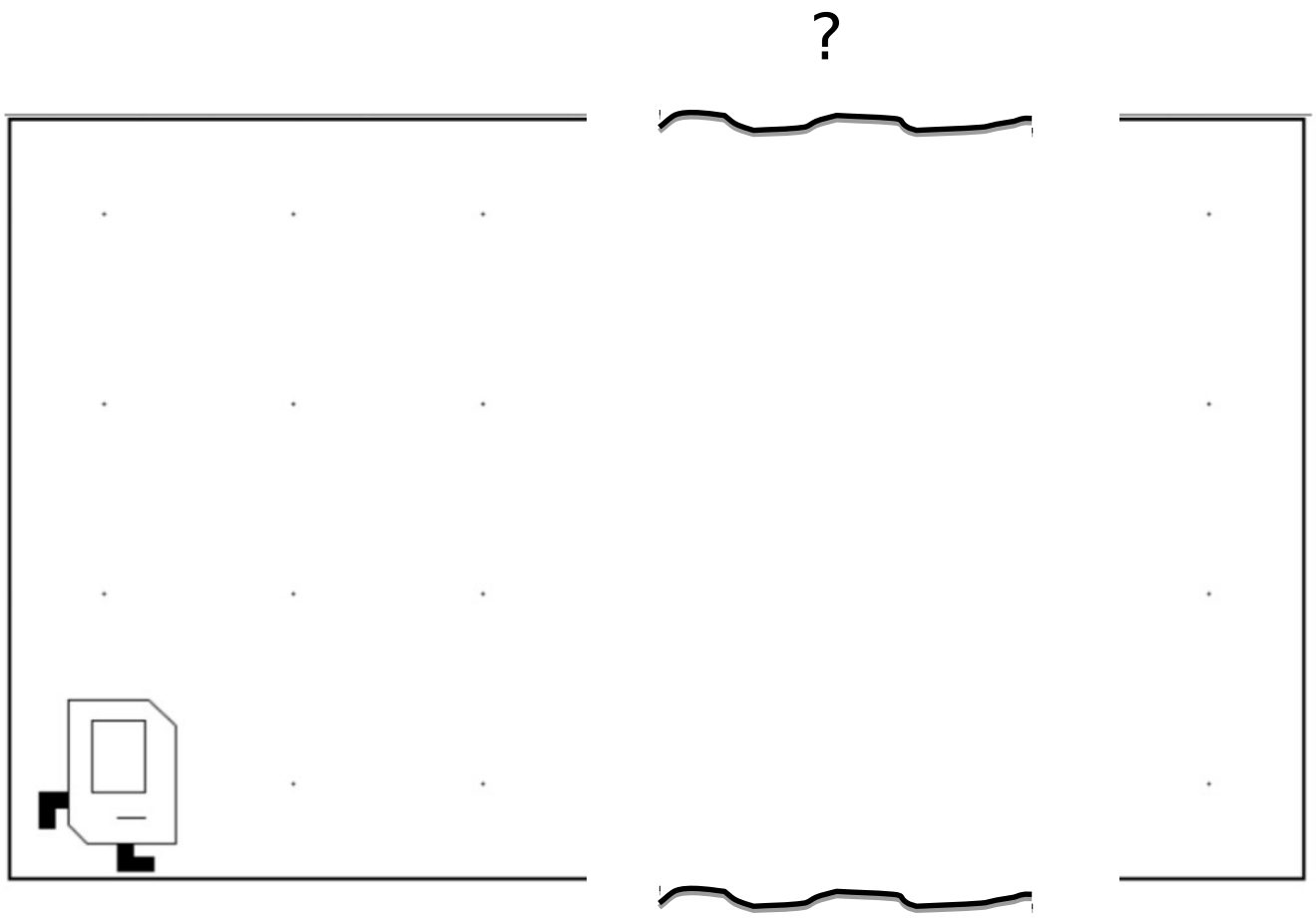
Understanding for loops

```
for(int numTickets = 3; numTickets > 0; numTickets--) {
     dance();
}
```

```
for(int i = 0; i < 3; i++) {
     dance();
}
```

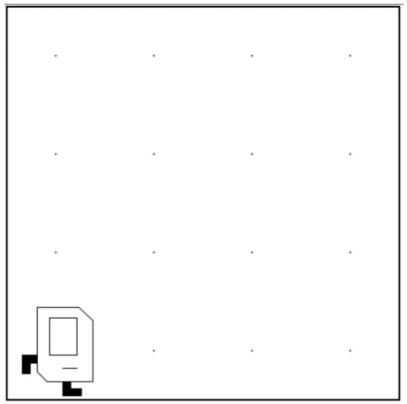
You can stick to the example until we talk about **int and ++**

Don’t Know World Size

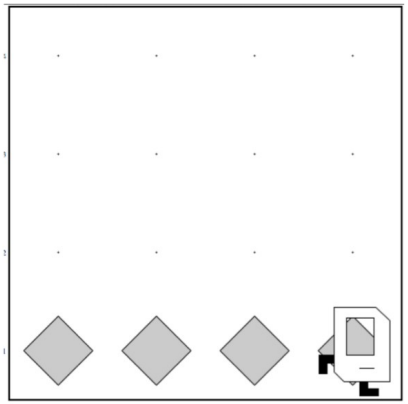


Work in Any World

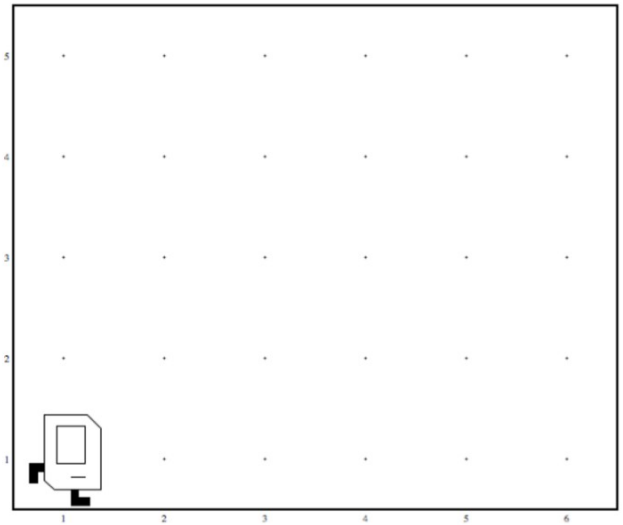
Before



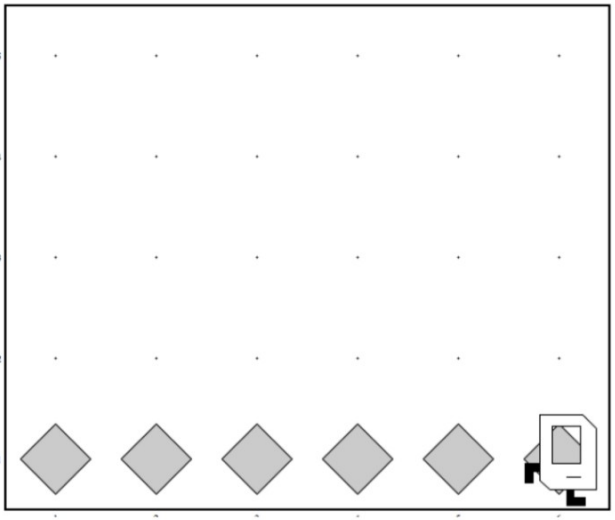
After



Before



After



While Loop

While Loop

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(condition) {
            code to repeat
        }

    }

}
```

Check the condition
True -> execute

Check the condition
True -> execute

....

Check the condition
False-> don't execute.

Continue the rest of the program

Conditions Karel can check

```
frontIsClear()     frontIsBlocked()
leftIsClear()      leftIsBlocked()
rightIsClear()     rightIsBlocked()
beepersPresent()   noBeepersPresent()
beepersInBag()     noBeepersInBag()
facingNorth()      notFacingNorth()
facingEast()       notFacingEast()
facingSouth()      notFacingSouth()
facingWest()       notFacingWest()
```

Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```

Any guess what this code will do?

How should we modify it to have beepers on the line?

Place Beeper Line

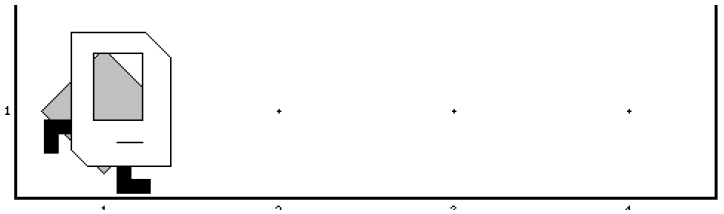
```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```

Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```

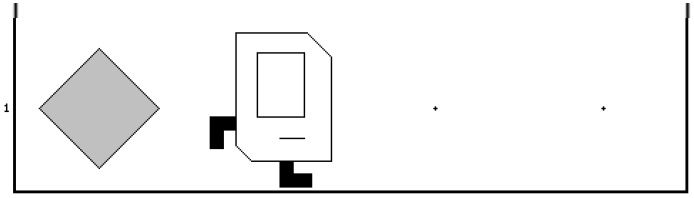
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```



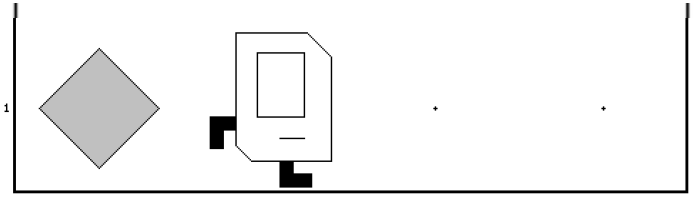
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```



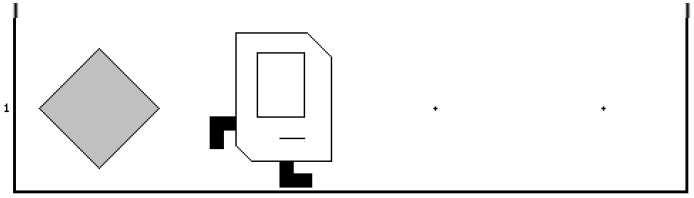
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```



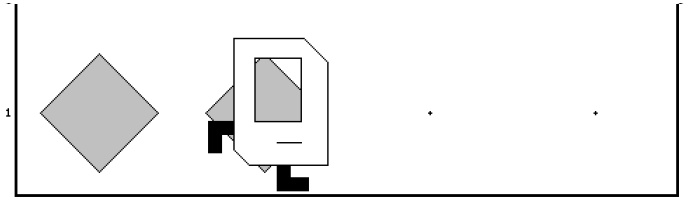
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```



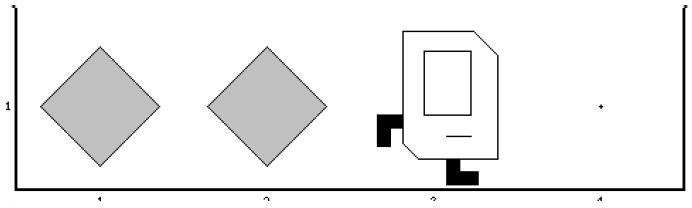
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```



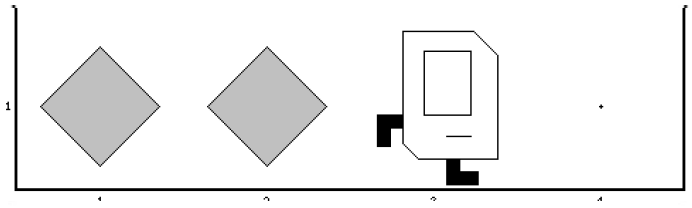
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```



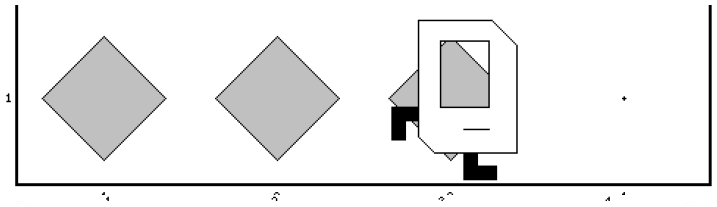
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```



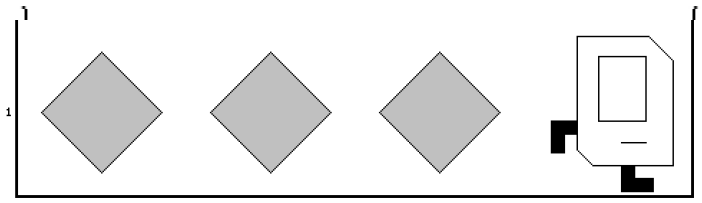
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```



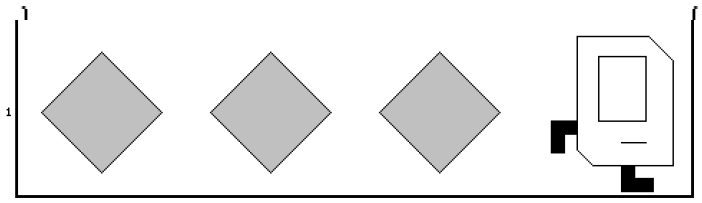
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```



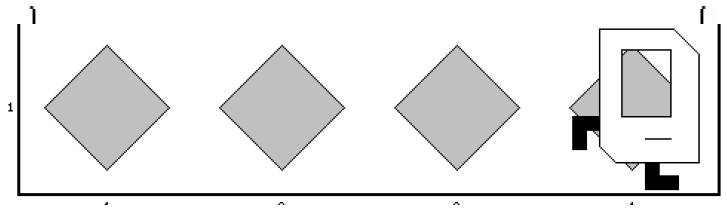
Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```

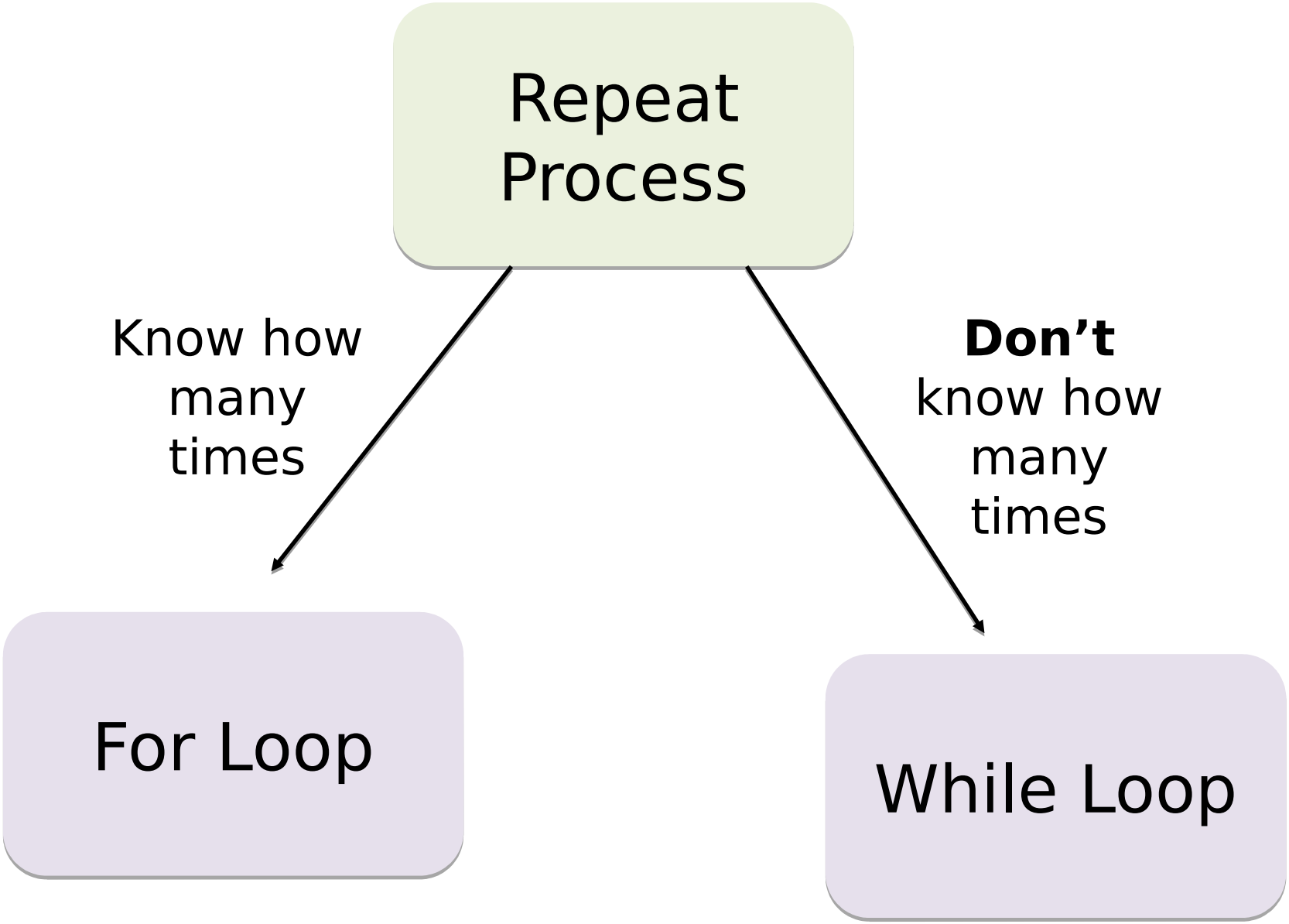


Place Beeper Line

```
import stanford.karel.*;

public class BeeperLine extends SuperKarel {

    public void run() {

        // example while loop
        while(frontIsClear()) {
            putBeeper();
            move();
        }
        // extra put beeper
        putBeeper();
    }

}
```



Which Loop


Repeat Process
Know how many times
Don't know how many times
For Loop
While Loop

What if you only want to perform a single loop based on a condition?

If statement

If Statement

```
import stanford.karel.*;

public class IfExample extends SuperKarel {

    public void run() {

        // example of an if statement
        if(condition) {
            code to run if condition is true
        }

    }

}
```

If Statement

```
import stanford.karel.*;

public class IfExample extends CSBridgeStudent{

    public void run() {

        // example of an if statement
        if(youAreInCSBridge()) {
            raiseYourHand();
        }

    }

}
```

Assume yourself as Karel and execute this program!

Let’s teach Karel to be more careful and stop killing itself trying to go through the walls.

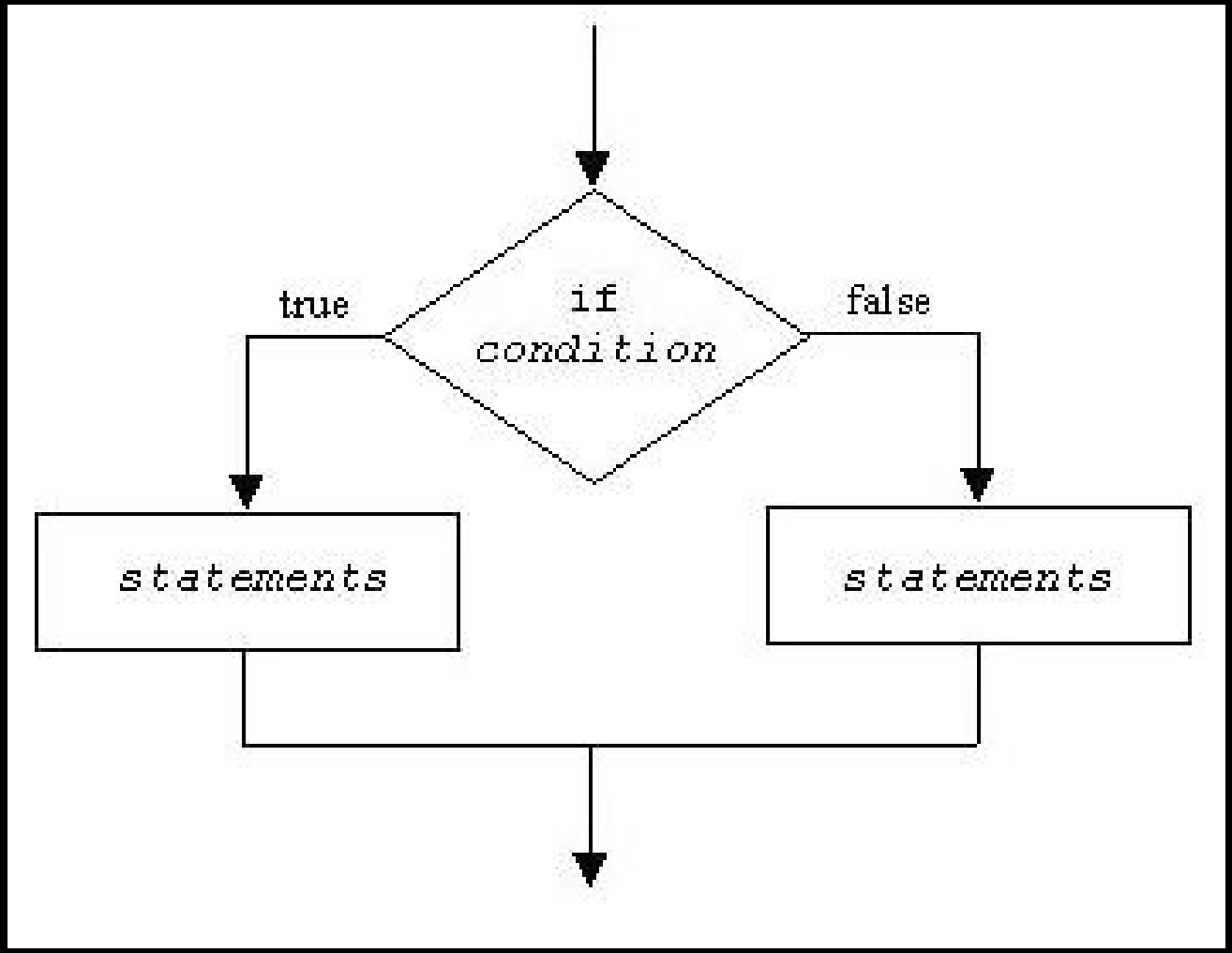
Could there be a safer way to move?

If Statement

```
import stanford.karel.*;

public class IfExample extends SuperKarel{

    public void run() {
        safeMove();
    }

    private void safeMove() {
        if(frontIsClear()) {
            move();
        }
    }

}
```

We can also specify what will happens if the condition is false

If else statement



Karel goes to the army:

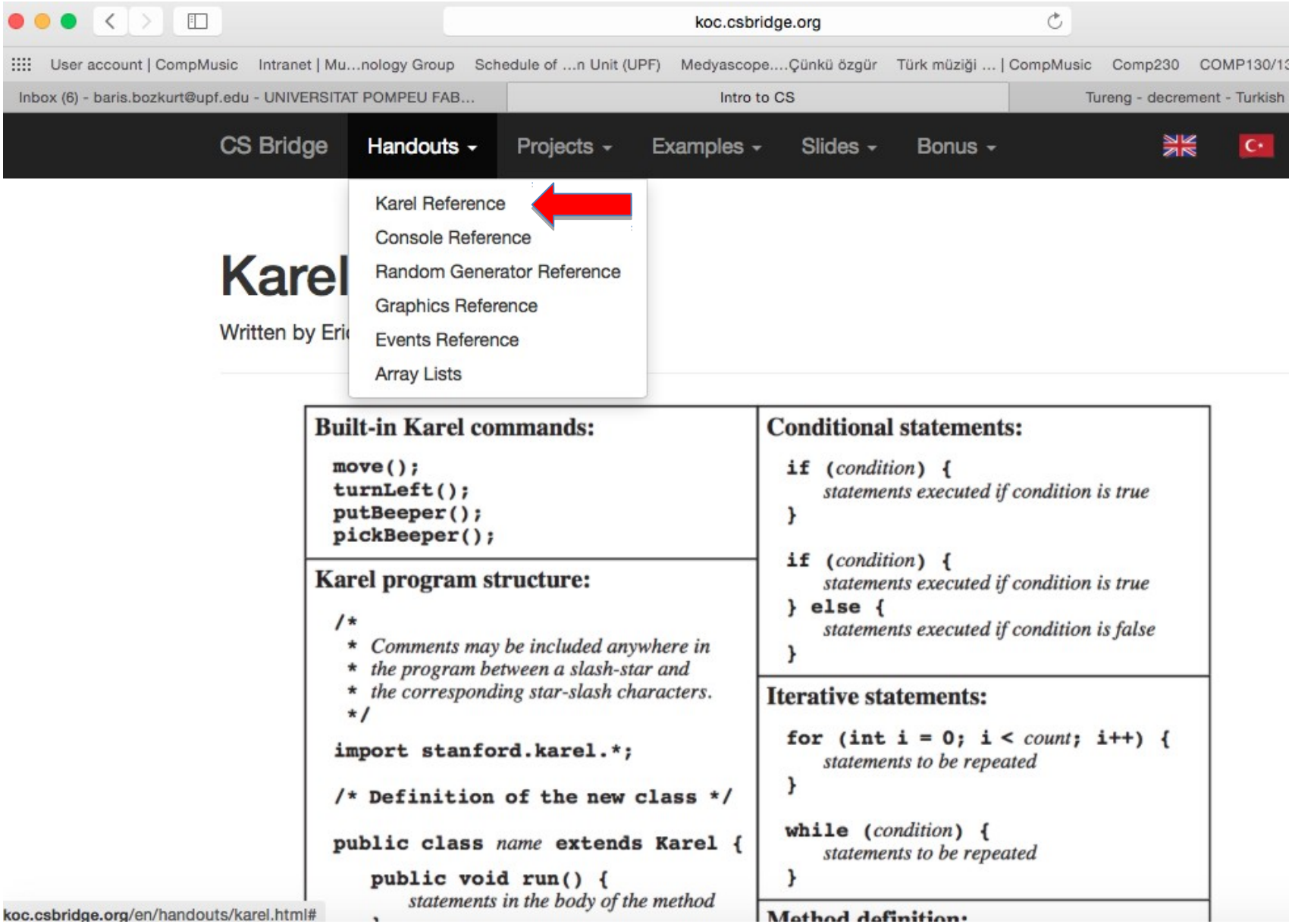
- If there is beeper take it
- If there is no beeper put one

… silly? That’s the point!

If / Else Statement

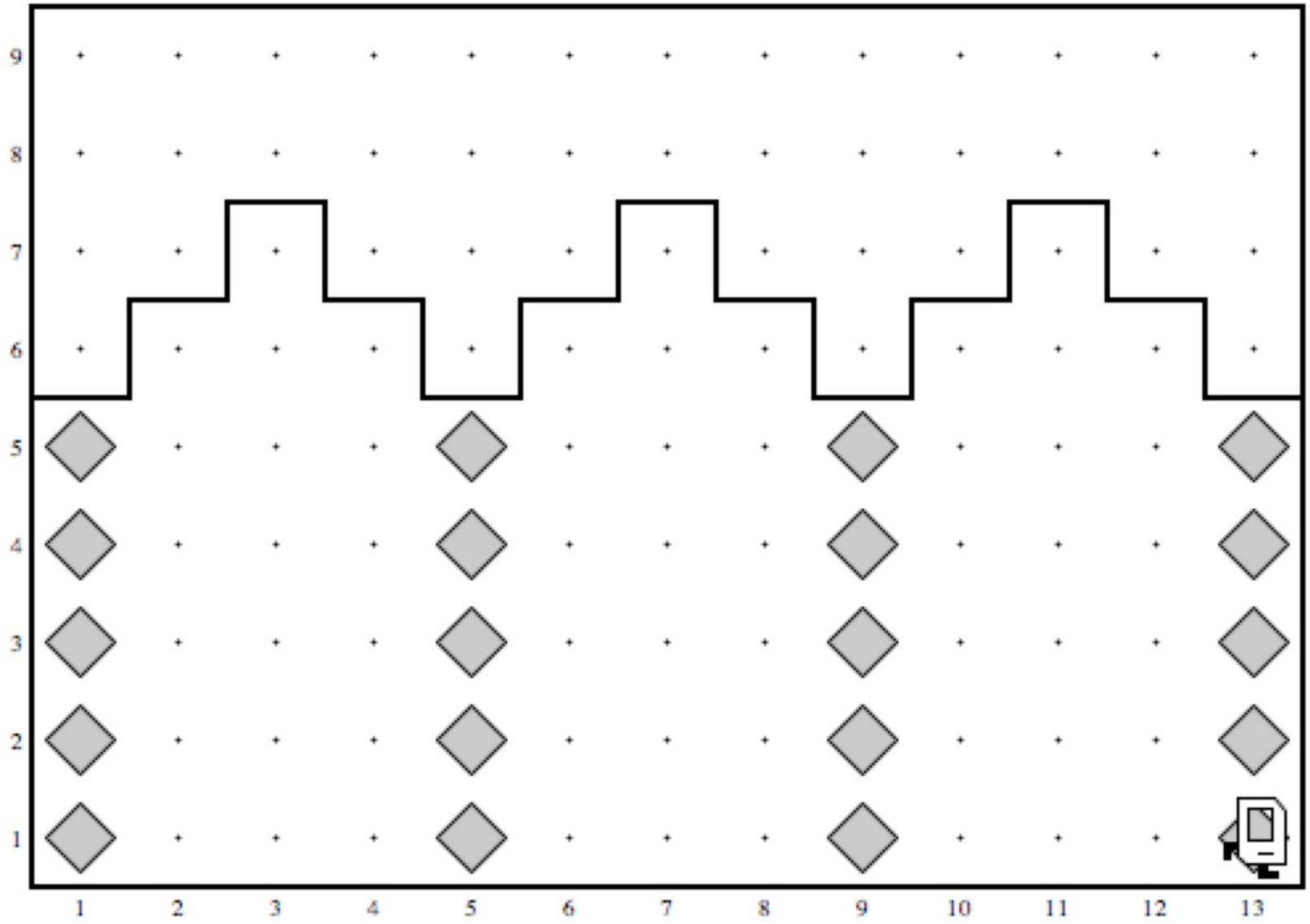
```
import stanford.karel.*;

public class IfExample extends SuperKarel{

    public void run() {
        invertBeeper();
    }

    private void invertBeeper() {
        if(beeperPresent()) {
            pickBeeper();
        } else {
            putBeeper();
        }
    }

}
```

Karel Conditions



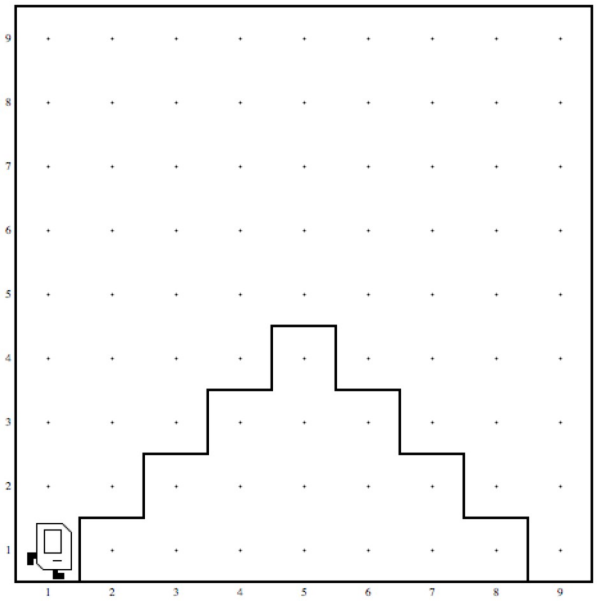
Your tasks this afternoon

1. Program Build Efes

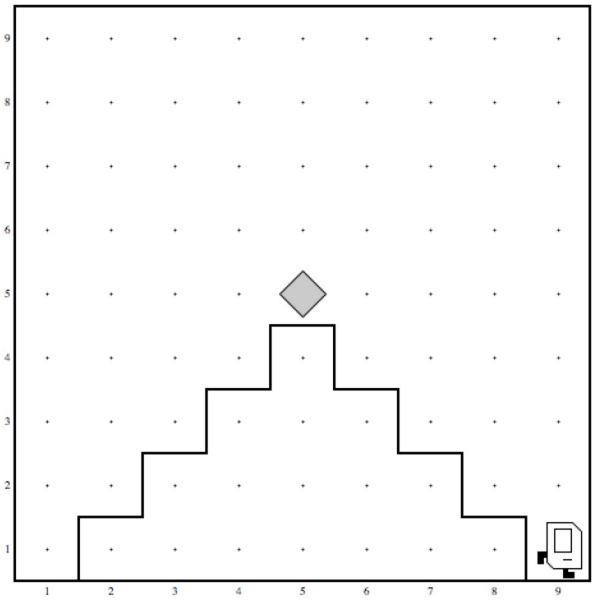


2. Mountain Karel

Before

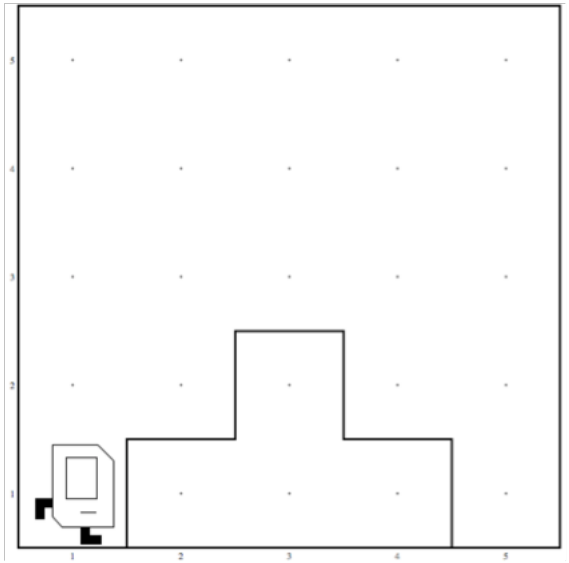


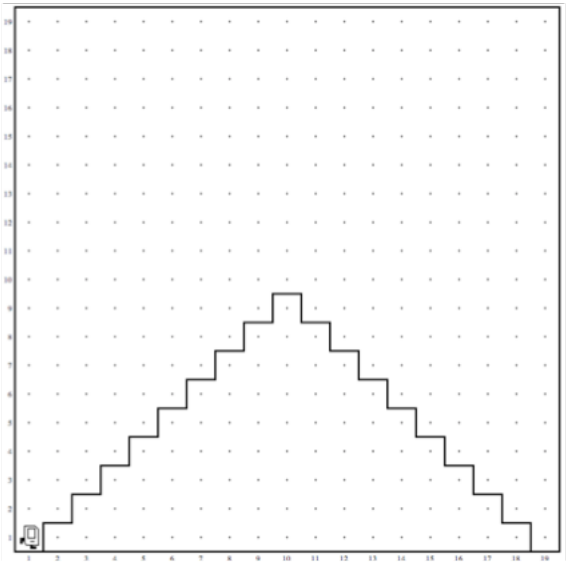
After

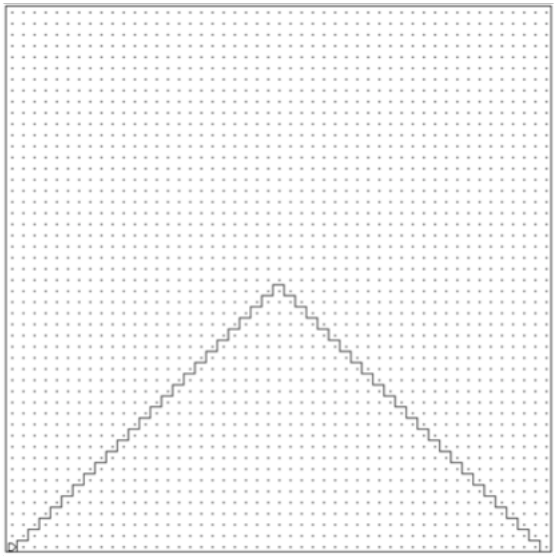


2. Mountain Karel

Should work on a world of any size

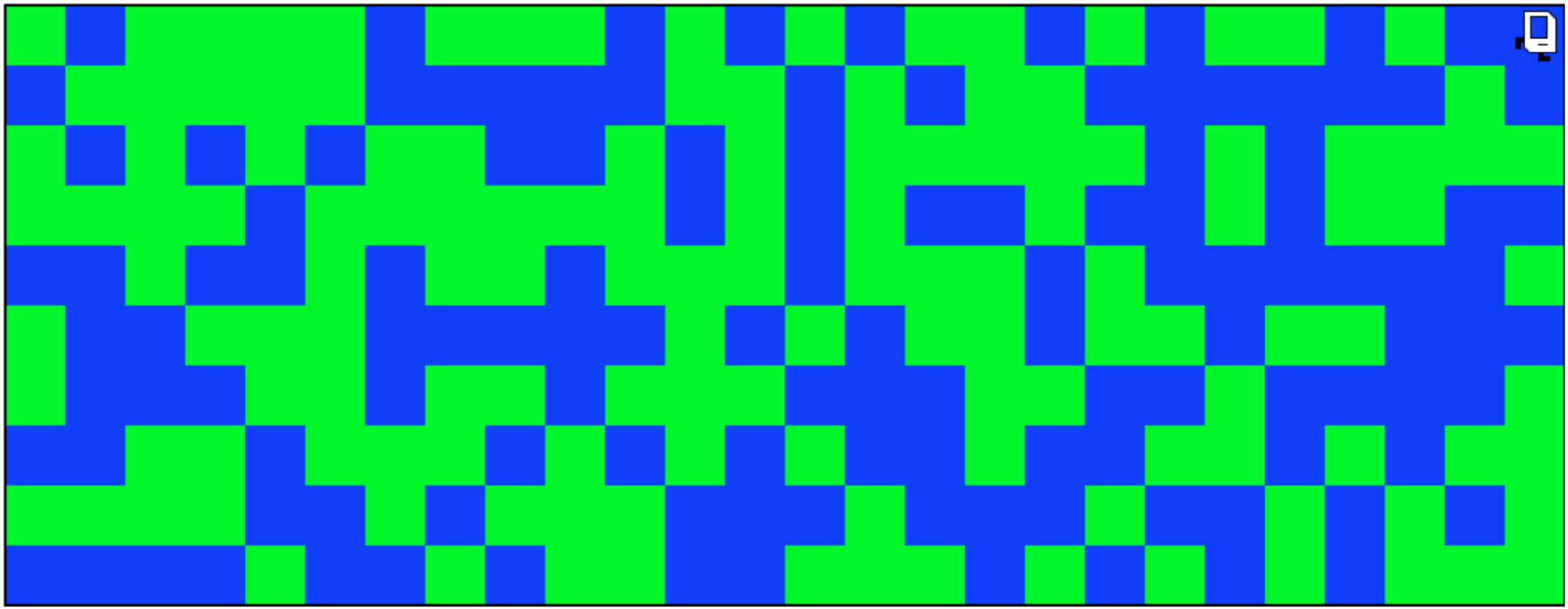






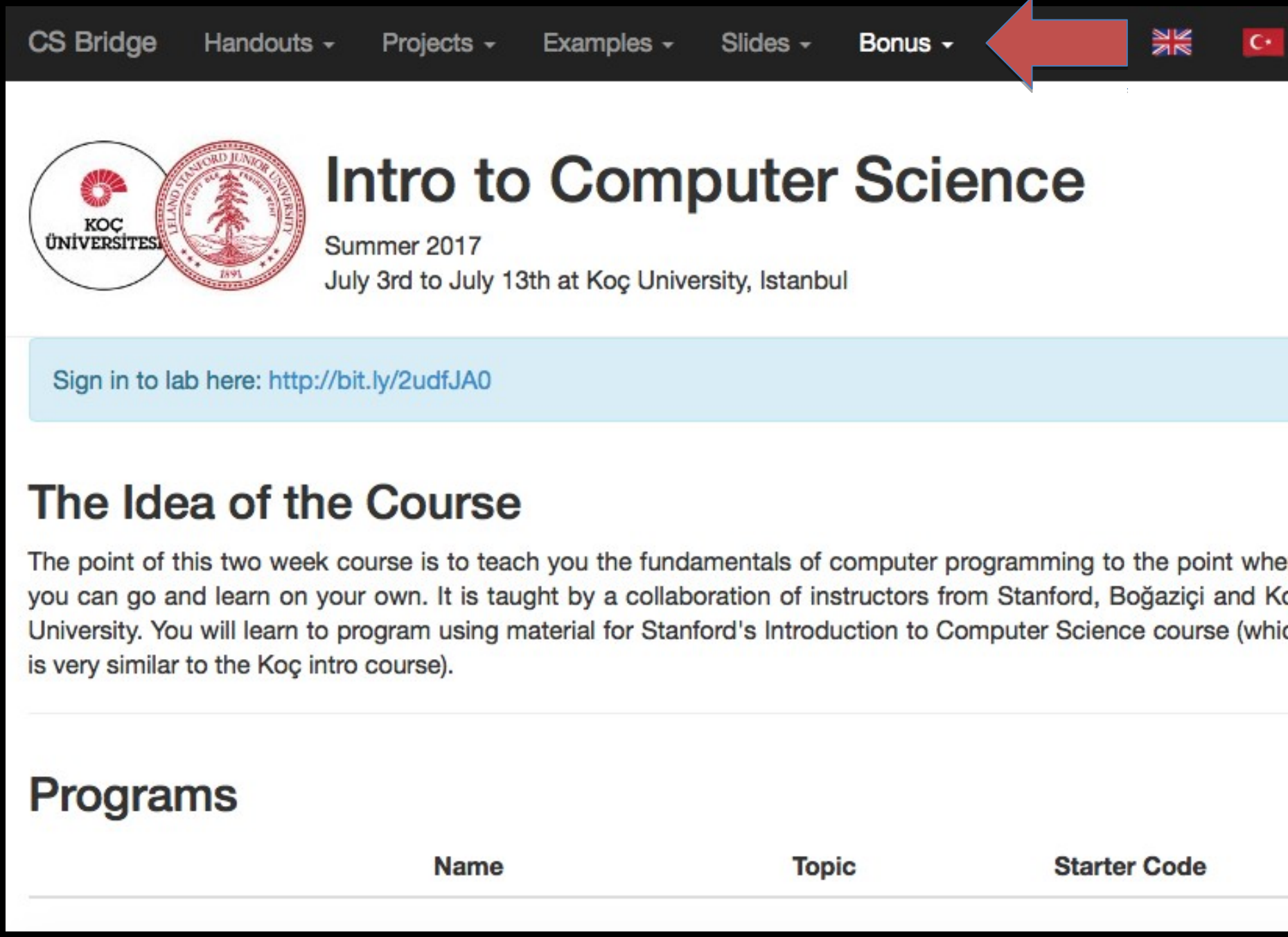
3. Random Painter

https://koc.csbridge.org/en/projects/randomPainter.html



Read the description and try to understand.

What if I finish early?


CS Bridge
Handouts
Projects
Examples
Slides
Bonus
KOÇ ÜNİVERSİTESİ
Intro to Computer Science
Summer 2017
July 3rd to July 13th at Koç University, Istanbul
Sign in to lab here: http://bit.ly/2udfJA0
The Idea of the Course
The point of this two week course is to teach you the fundamentals of computer programming to the point where you can go and learn on your own. It is taught by a collaboration of instructors from Stanford, Boğaziçi and Koç University. You will learn to program using material for Stanford's Introduction to Computer Science course (which is very similar to the Koç intro course).
Programs
Name
Topic
Starter Code