**Assignment - 3. JavaScript**

*All the task codes have to be inserted into a doc file and uploaded to moodle.*

**Task 1-Strings.** Changing String Casing(**10 points**):

1. Go to Google Chrome Browser-> View -> Developers ->JavaScript Console then to snippets(*as was shown in the lecture*)
2. Create a prompt which will ask “What is your name?” and user can write his/her name;
3. Assign a name to a variable;
4. Create an alert which will show the message “Hello” + name written into the prompt. An example is “Hello Gulzhan”. *A user can write his/her name in different ways, he/she can write it starting with lowercase or can start the name with uppercase and use uppercase in the middle of the name as well. Examples:*

*aidana*

*AidAna*

*aiDana*

1. By using properties of JavaScript as .length(); .toLowerCase(); .toUpperCase(); .slice(a,b); and concatenation, write a program which standardizes the display of the name regardless of how the user writes their name.

Examples:

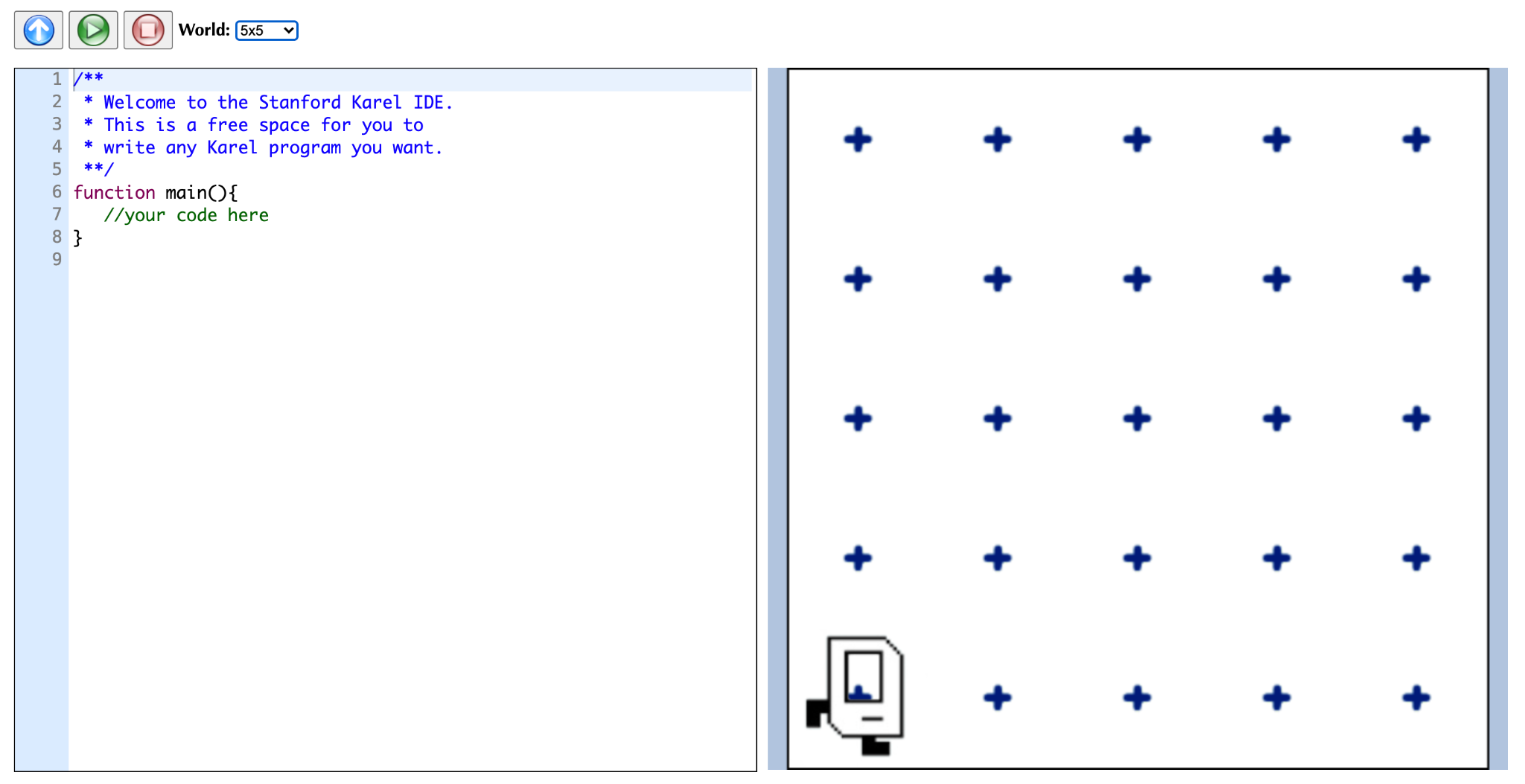
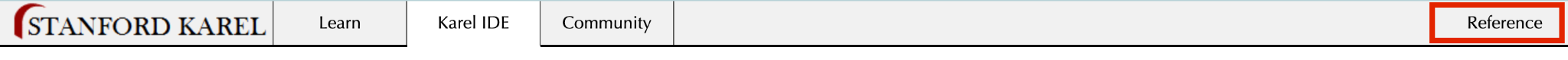
prompt: “What is your name?” - anuar

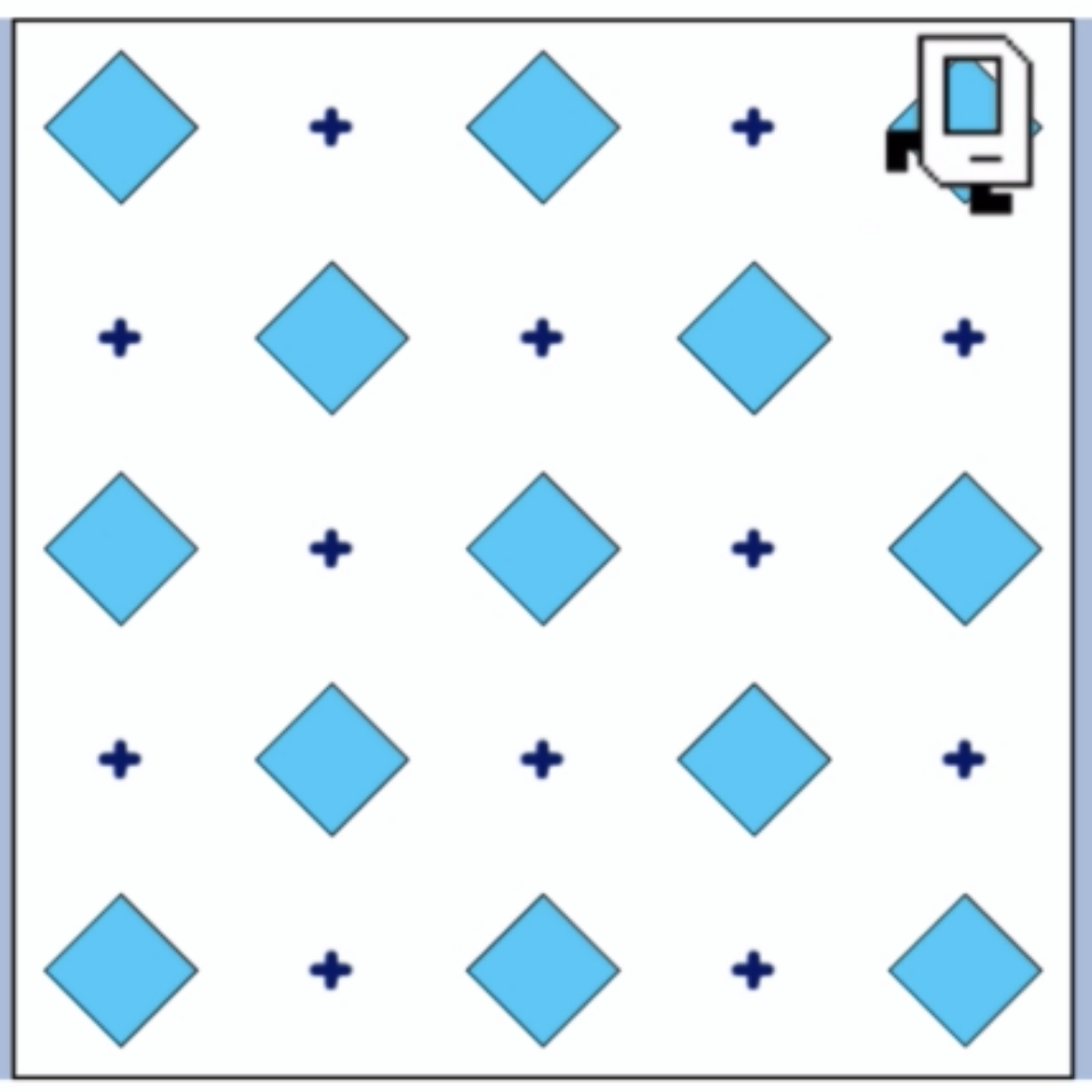
alert: “Hello Anuar”

prompt: “What is your name?” - AkeRKe

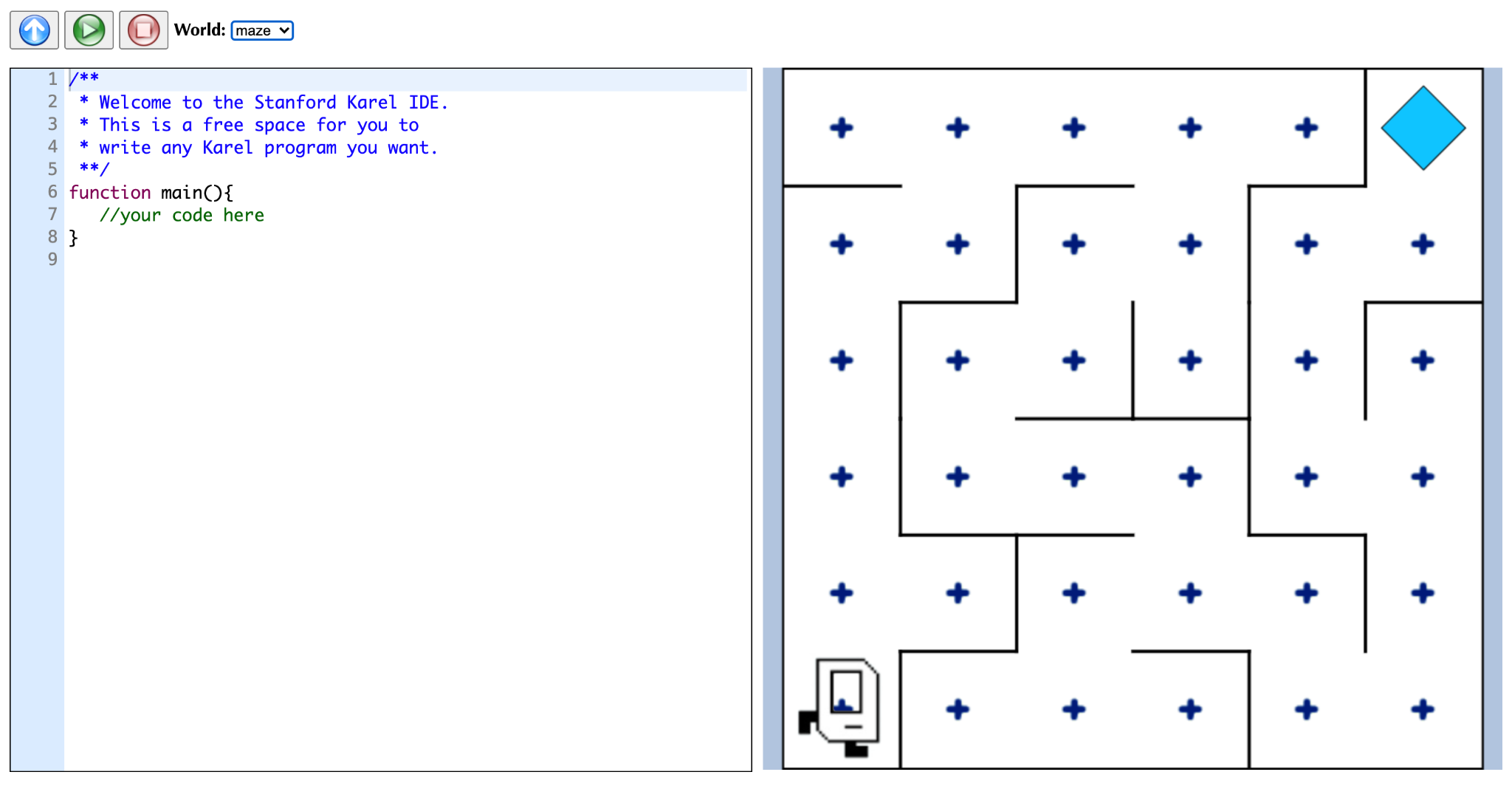
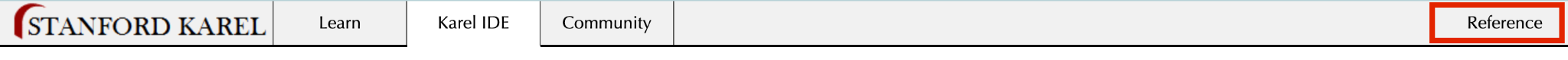
alert: “Hello Akerke”

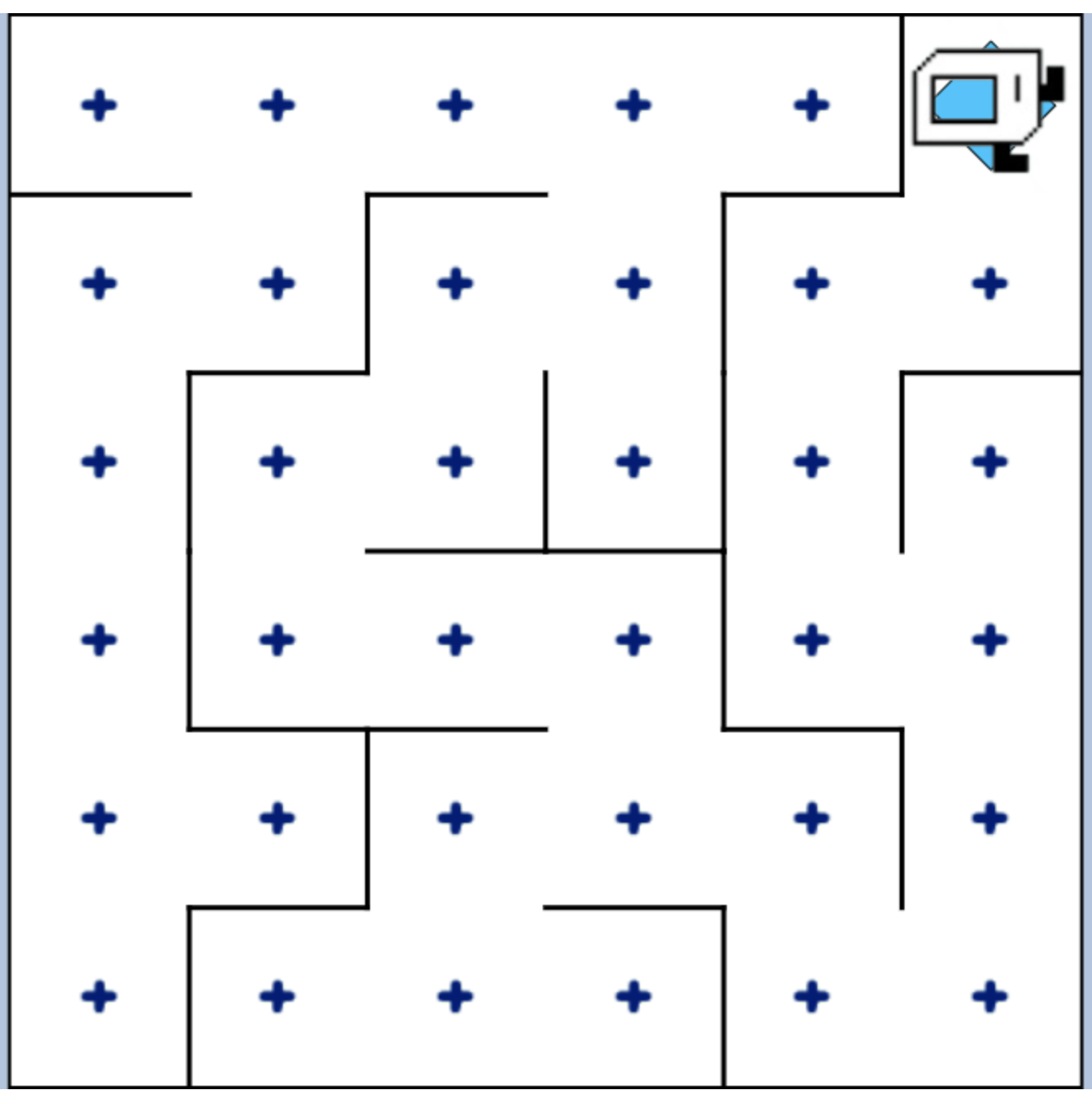
**Task 2- Functions**. The Karel Robot(**5 points**):

1. Go to the website <http://stanford.edu/~cpiech/karel/ide.html>
2. Choose the world - 5x5;
3. Play around with References and learn what Karel Robot can do;
4. By the shortest and least repetitive way write functions list to put Beepers in the way showed below;



**Task 3 - Functions.** The Karel Robot Maze(**5 points**);

1. Go to the website <http://stanford.edu/~cpiech/karel/ide.html>
2. Choose the world - maze;
3. Play around with References and learn what Karel Robot can do;
4. By the shortest and least repetitive way write functions list to go through maze and reach the Beeper;



**Task 4 - Functions. Parameters and Arguments.** Life in Weeks. (**10 points**)

1. You can continue work inside of the Google Chrome JavaScript Console or in your text editor by creating a file with .js extension. Or you can try Repl.it playground as well.
2. In this task you are going to create a function that tells us how many days, weeks and months we have left if we live until 90 years old. It will take your current age as the input and console.logs a message with our time left in this format: **You have x days, y weeks, and z months left.** Where x, y and z are replaced with the actual calculated numbers. For this challenge, assume there are **365** days in a year, **52** weeks in a year and **12** months in a year.

**Example Input:**

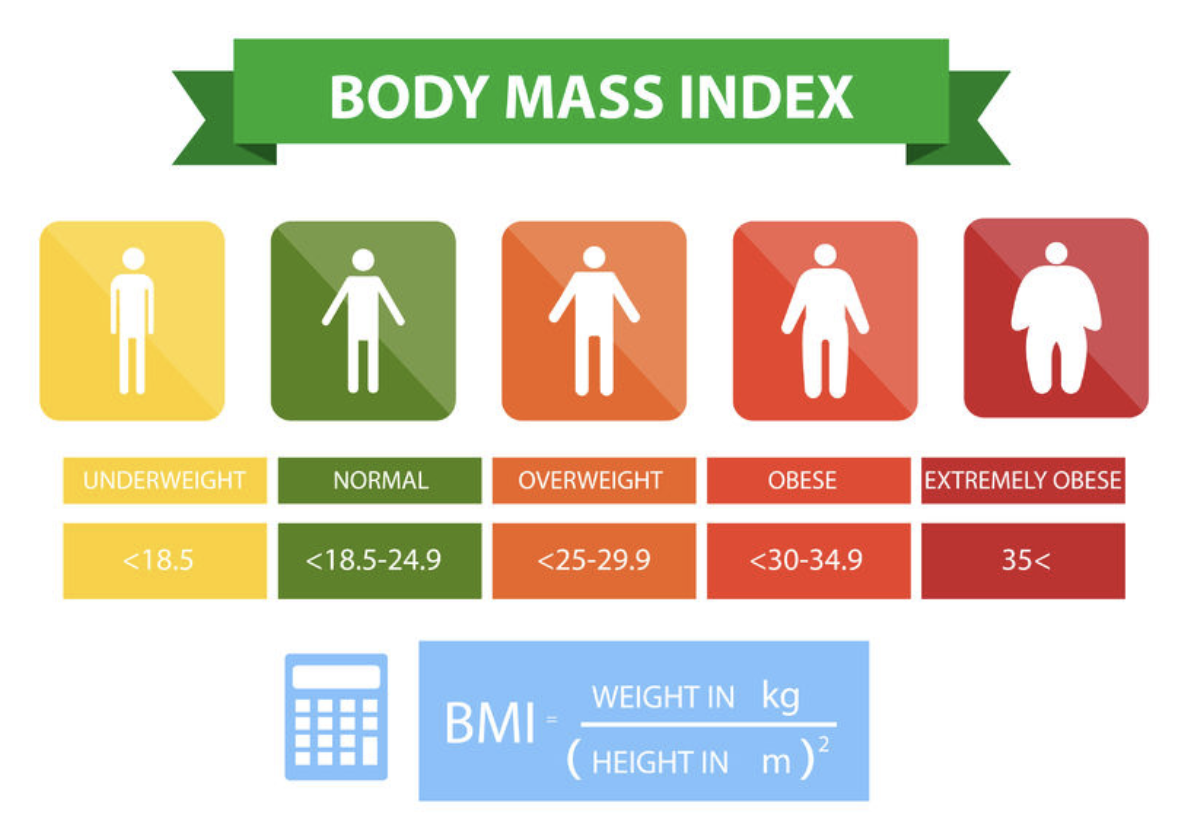
e.g. If you are 56 years old:

lifeInWeeks(56)

**Example Output:**

You have 12410 days, 1768 weeks, and 408 months left.

**Task 5 - IF/ELSE Statements.** BMI Calculator (**10 points**)



In this task create a BMI calculator using JavaScript functions. The Body Mass Index (BMI) is a way of estimating the amount of body fat. It's used in medicine to calculate risk of heart disease. You can calculate it using the formula above, where weight divided by height squared.

Write a function that outputs (returns) a different message depending on the BMI.

BMI **< 18.5** , the output should be: "Your BMI is <bmi>, **so you are underweight**.

BMI **18.5-24.9** , the output should be: "Your BMI is <bmi>, **so you have a normal weight**.

BMI **>24.9** , the output should be: "Your BMI is <bmi>, **so you are overweight**.

The message must be returned as an output from your function; you should **not need** to use **alert, prompt** or **console.log** in this task.

**Task 6 - IF/ELSE Statements.** Leap Year (**20 points**)

Write a program that works out whether if a given year is a leap year. A normal year has 365 days, leap years have 366, with an extra day in February. The reason why we have leap years is really fascinating, [this video](https://www.youtube.com/watch?v=xX96xng7sAE) goes into more detail.

This is how to work out whether if a particular year is a leap year:

A year is a leap year if it is evenly divisible by **4;**

**except** if that year is also evenly divisible by **100**;

**unless** that year is also evenly divisible by **400**;

e.g. Is the year 2000 a leap year?:

2000 ÷ 4 = 500 (Leap)

2000 ÷ 100 = 20 (Not Leap)

2000 ÷ 400 = 5 (Leap!)

So the year 2000 is a leap year.

But the year 2100 is not a leap year because:

2100 ÷ 4 = 525 (Leap)

2100 ÷ 100 = 21 (Not Leap)

2100 ÷ 400 = 5.25 (Not Leap)

Example Input 1:

2400

Example Output 1:

Leap year.

Example Input 2:

1989

Example Output 2:

Not Leap year.

**Hint:**

1. Remember that the modulo (%) operator gives you the remainder of a division.
2. Try to visualize the rules by creating a flow chart on [www.draw.io](http://www.draw.io)
3. If you really get stuck, you can see the [flow chart I created](https://bit.ly/36BjS2D)
4. Try to run your code in [this Repl.it playground](https://repl.it/@appbrewery/Leap-year-challenge) and check it against the [known leap years](https://www.mathsisfun.com/leap-years.html).

**Task 7 - Arrays.** Who’s buying lunch? (**20 points**)

You are going to write a function which will select a random name from a list of names. The person selected will have to pay for everybody's food bill.

**Important:** The output should be returned from the function and you do not need **alert, prompt** or **console.log**. The output should match the example output exactly, including capitalisation and punctuation.

**Example Input**

[“Aidana”, “Bekbolat”, “Mariya”, “Aleksandr”, “Diana”]

**Example Output**

Diana is going to buy lunch today.

**Hint:**

1. You might need to think about [Array.length](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/length)
2. Remember that Arrays start at position **0**.

**Task 8 - Loops.** Fibonacci (**20 points**)

Fibonacci was an Italian mathematician who came up with the [Fibonacci sequence](https://en.wikipedia.org/wiki/Fibonacci_number).

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 …

Where every number is the sum of the two previous ones.

e.g. 0, 1, 1, 2, 3, 5 comes from

0 + 1 = 1

1 + 1 = 2

2 + 1 = 3

3 + 2 = 5 etc.

Create a function where you can call it by writing the code:

fibonacciGenerator(n);

Where **n** is the number of items in the sequence.

So we should be able to call:

fibonacciGenerator(3) and get [0,1,1] as the output.

**IMPORTANT:** The solution checker is expecting an **array** as the correct output.

You don’t need any **alerts or prompts**, the result should be returned from the function as an output.

The first two numbers in the sequence **must** be **0 and 1**.

Also, if you decide to create a for loop, make sure you explicitly specify var i = 0 rather than simply writing i=0. This is a quirk of the testing suite.

e.g. for (var i = 0; i < 10; i ++)

**HINT:** Use [this flow chart](https://drive.google.com/file/d/1g8vVtqhSj44vcElfc-HK0nMbecteW8Yg/view?usp=sharing) to understand the logic if you get stuck.