

Lab 09 Javascript 2

**Not to be redistributed
to Course Hero or any
other public websites**

General Information**What you should do**

- You should first review Lecture 4 (Part B) up to slide 13 and be familiar with the concepts on Javascript as well as the related code examples so that you can refer to the specific techniques from the corresponding slides when completing the tasks in this lab.
- You should try to come up with the code yourself as much as possible. Do not be afraid of making mistakes, since debugging (finding out where your code goes wrong and fixing it) is part of the learning process.
- We do not give out model programs to the exercises. There can be multiple ways to write the code that solves the same problem. It is important that you build up the program logic yourself instead of merely looking at some code that you do not understand. At any time if you are lost or if you have any questions, feel free to ask the instructor, tutor, or teaching assistant and we will be very happy to help you.

Self-Discovery

- Most lab tasks are designed to be relatively simple such that you can take the time to think about the related underlying concepts. Besides, we also encourage you to discover things on your own which may not be specified in the tasks.

Task 1.1 Create a New HTML File from Template in Komodo Edit

Create a new HTML file from Komodo Edit in the same way as described in Lab 08 Task 1.1:

1. Open Komodo Edit and select "New File from Template" (or press the shortcut keys Ctrl+Shift+N)
2. A dialog box will pop up on the right. Select "HTML 5" from the list

Remember to save the file after you make changes before opening it on a browser to view it.

Note that you should NOT copy and paste any code from this document to your code editor. Type up the code instead, otherwise it may not work properly, especially with symbols such as quotation marks "".

Task 1.2 Check if an integer is divisible by another integer

In Lab 04 Task 1.2, you have written a Scratch program to check if an integer is divisible by another integer. In this task, we would like to do the same thing in Javascript. Here are the requirements for the program:

1. When the webpage is opened on the browser, a pop-up window will ask the user to enter an integer.
2. It is assumed that the user has indeed entered an integer so there is no need to validate the input.
3. The webpage will then display " x is divisible by 2" or " x is NOT divisible by 2" depending on the actual case, where x is the integer entered by the user.

Now try to compose the program by yourself. If you have trouble writing it, then refer to the following hints:

- Use the `onload` event handler to call a function so that the function will be called after the webpage is opened on the browser
- Use the `prompt` function to get the input from the user. The input will be returned as a string so you need to use the `Number` function to convert it to the number type before processing it as a number
- Use the Javascript modulus operator `%` to get the remainder from a division, e.g., `6 % 2` means 6 mod 2, the remainder when 6 is divided by 2. If x is divisible by y , then $x \% y$ would be equal to 0
- Use the `if-else` statement to implement the 2-way conditional
- Create a `div` element in the HTML and assign an ID to it. Then set `document.getElementById([ID]).innerHTML` to the corresponding message depending on whether the input integer is divisible by 2

A sample HTML and Javascript code corresponding to an implementation according to the above requirements is shown below for your reference:

```

1  <!DOCTYPE html>
2
3  <html>
4  <head>
5      <title></title>
6      <script>
7          function init() {
8              var x, p, s;
9              p = prompt("Enter an integer");
10             x = Number(p);
11             if (x % 2 == 0)
12                 s = x + " is divisible by 2.";
13             else
14                 s = x + " is NOT divisible by 2.";
15             document.getElementById("result").innerHTML = s;
16         }
17     </script>
18 </head>
19 <body onload="init();">
20     <div id="result"></div>
21 </body>
22 </html>

```

Exercise 1.2: Modify your program so that it satisfies the following additional specifications. You should focus on one specification at a time, test the program each time after you finish with one specification to make sure that it is working correctly before moving to work on the next specification:

- Replace the divisor 2 by a variable y , whose value should be obtained from the user input. This means that your program should first ask the user to input the dividend, convert the corresponding input string to number and store it to the variable x . Your program should then ask the user to input the divisor, convert the corresponding input string to number and store it to the variable y . You can assume that the user always follows the instructions to enter an integer for the input so you do not need to validate it. Your program should output whether x is divisible by y on the webpage.
- Check if the divisor is 0, then display the message "Divide by 0 error!" on the webpage.
- Instead of assuming that x is always the dividend and y is always the divisor, your program should just ask the user to input the integers and convert the input strings to numbers to be stored as x and y . Assume that the user input is always an integer so you do not need to validate it. Your program should find the bigger number between x and y and store it by the variable m . Similarly, your program should find the smaller number between x and y and store it by the variable n . Afterwards, your program should check and display if m is divisible by n on the webpage.

Task 1.3 Check for valid floor numbers in a building

In Lab 04 Task 1.3, you have written a Scratch program to check whether the user input is a valid floor number. In this task, we will do the same thing using Javascript. The problem is stated again below:

A building has its top floor on the 50th floor and has the following floor numbers:

1,2,3,5,6,7,8,9,10,11,12,15,16,17,18,19,20,21,22,23,25,26,27,28,29,30,31,32,33,35,36,37,38,39,50

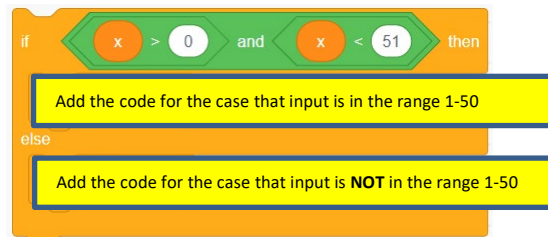
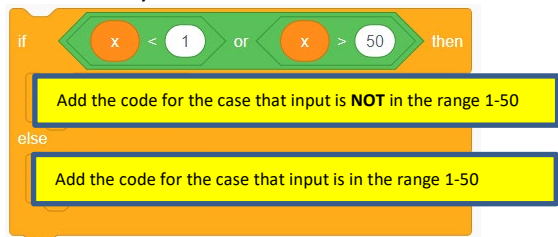
The other floor numbers within this range (1-50) are not included in this building because 1) the digit 4 is considered as unlucky; and 2) the number 13 is considered as unlucky.

When you try to enter this building, a robot receptionist will greet you and ask you which floor you would like to go. Write a program that simulates the robot receptionist such that

- It will first ask the user to input the floor number
- If the floor number is not valid, i.e., if it is not in the above floor number list, then display the message "Your input x is NOT a valid floor number!", where x is the input floor number
- If the floor number is valid, i.e., if it is in the above floor number list, then display the message "Welcome! Please enter! Take the elevator to go to floor x ", where x is the input floor number

Hints:

- A) You can assume that the input is an integer so you do not need to validate it
- B) Your program can first check if the input is in the range 1-50. From Lab 04 Task 1.2, you have learnt about 2 ways to form the conditional in Scratch:



Now try to form the same conditionals in Javascript. Note that the OR operator in Javascript is `||` and the AND operator in Javascript is `&&`

Also note that if there is only one statement after if or after else, then you can just put it there directly, i.e.,

```
if ([BooleanExpression])
    statement1;
else
    statement2;
```

However, if you would like to include more than one statement under the if-part or under the else-part, then you must enclose these statements with curly brackets `{ }` i.e.,

```
if ([BooleanExpression]) {
    statement1;
    statement2;
}
else {
    statement3;
    statement4;
}
```

- C) If the input is in the range 1-50, your program can then check if it is in the above floor number list. We have already mentioned that the numbers in the range 1-50 that are not in the above floor number list either contain a digit 4 or are equal to 13. Now we can implement the code to explore if the input has any of these patterns.

I) To check if the input is equal to 13, you can just use the Boolean expression `x == 13`

II) To check if the input contains a digit 4, we need to examine each digit of the input number. Since it has already been checked that the input is an integer in the range 1-50, the input is at most a 2-digit number. So we need to extract the tens' place and the ones' place from the input and check if each one is equal to 4.

a) Extracting the ones' place

From Lab 04 Task 1.2, you have learnt that you can use the modulus function to extract the remainder and then compare if it is equal to 4, so you can form the Boolean expression `x % 10 == 4`

b) Extracting the tens' place

From Lab 04 Task 1.2, you have learnt that we can obtain the ten's place of the input which is a 2-digit number by first dividing by 10 and keeping only the integer part. This can be implemented using the floor function in Scratch by forming the block



In Javascript, the syntax of the floor function is `Math.floor()`, e.g., the value of `Math.floor(3.4)` is 3 and the value of `Math.floor(5.9)` is 5. Essentially

`Math.floor()` means that the function `floor()` is a method of the `Math` object. You can check out many other useful mathematical functions of the `Math` object from here: https://www.w3schools.com/js/js_math.asp

The resulting Boolean expression in Javascript to check the ten's place is `Math.floor(x/10) == 4`

- III) You can make use of the Boolean blocks derived in I), IIa), IIb) and check it one by one. If a Boolean block returns true, then you know that the input is not a valid floor number. Alternatively, you can use Boolean operators to join these 3 Boolean blocks and form a compound Boolean block so that you can use a single 2-way conditional to check for this. Should you use the OR operator or the AND operator to join these 3 Boolean blocks?
- 4) After verifying that the input is a valid floor number and displaying "Welcome! Please enter!", the robot receptionist will give you further directions:
- If your floor number is among 1,2,3,5,6,7,8,9,10,11,12,15, then the robot receptionist will say "Please proceed to take the lift on the left to go to floor x ", where x is the input floor number.
 - If your floor number is among 16,17,18,19,20,21,22,23,25,26,27,28, then the robot receptionist will say "Please proceed to take the lift on the middle to go to floor x ", where x is the input floor number.
 - If your floor number is among 29,30,31,32,33,35,36,37,38,39,50, then the robot receptionist will say "Please proceed to take the lift on the right to go to floor x ", where x is the input floor number.

Task 1.4 Recommend a travel option

This is the same problem as in Lab 04 Task 1.4 and here you will write a Javascript code to solve it. You can assume that the input by the user is indeed a number so you do not need to worry about checking if the input is a number or text (we will learn about how to do this in the next lab). The problem is stated again below:

Suppose that you need to go from Location A to Location B, and you have the following travel options with the corresponding transportation costs:

	Travel Option	Transportation Cost (\$)
1	On foot	0
2	By tram	2.6
3	By subway	10.7
4	By bus	11.1
5	By taxi	130

Write a program with the following specifications:

- When the webpage is opened, it will ask the user how much money the user has
- The program will then check if the user input is a valid amount of money. A valid amount of money means that it is a non-negative number and contains at most 1 decimal place. It will display a message "Your input is NOT a valid amount of money" if it is the case and the program ends.
- If the user input is indeed a valid amount of money, then the program would recommend the most expensive travel option that the user can afford, by displaying a message "You can go from Location A to Location B [travel option], and it will cost you [transportation cost]" on the webpage.

Example 1:

User input: 5.65

Message displayed on the webpage: Your input is NOT a valid amount of money

Example 2:

User input: 5

Message displayed on the webpage: You can go from Location A to Location B by tram, and it will cost you \$2.6

Example 3:

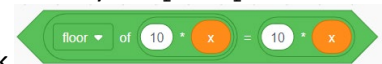
User input: 10.8

Message displayed on the webpage: You can go from Location A to Location B by subway, and it will cost you \$10.7

Test your program by using different input values. Make sure you include enough variations of the input such that each travel option appears at least once during your testing.

Hints:

- A) We assume that the user input is a number (instead of any arbitrary text). You still need to use the function `Number()` to convert the output from the `prompt` function from string to number type.
- B) Let `x` represents the resulting number converted from the string returned by the `prompt` function.



From Lab 04 Exercise 1.1, you have learnt that you can use the block to check if the number contains at most one decimal place. In Javascript, this Boolean expression can be written as `Math.floor(10*x) == 10*x`

Other than the above, you can further check if `x` is non-negative.

- C) Form a 5-way conditional by nesting 4 2-way conditionals

```

if (.....) {
    .....
}
else if (.....) {
    .....
}
else if (.....) {
    .....
}
else {
    .....
}

```

Task 1.5 List a series of numbers

Examine the following sample code and try to determine what the webpage will show when it is opened on the browser:

```

1  <!DOCTYPE html>
2
3  <html>
4  <head>
5      <title></title>
6      <script>
7          function init() {
8              var i, N, s;
9              N = 20;
10             s = "The integers from 1 to "+N+" are listed below:<br/>";
11             for (i=1; i<=N; i++) {
12                 s = s + i + "<br/>";
13             }
14             document.getElementById("result").innerHTML = s;
15         }
16     </script>
17 </head>
18 <body onload="init();">
19     <div id="result"></div>
20 </body>
21 </html>

```

Line 10: `s = "The integers from 1 to "+N+" are listed below:
"`;

It assigns a specific string to the variable `s`. Recall that the `+` operator concatenates the operands when at least one of them is a string.

Line 11: `for (i=1; i<=N; i++) {`

From the lecture you have learnt that a for-statement has the following 3 parts:

- 1) **Initialization:** The initialization statement `i=1` is executed at the beginning of the for-loop.
- 2) **Continuation Condition:** the continuation condition should be a Boolean expression that is either true or false. The for-loop will check for this condition and if it is true, then it will execute the loop action for one iteration. In this case, the continuation condition is `i<=N`.
- 3) **Increment Statement:** The increment statement is executed at the end of each iteration of the loop action. In this case, the statement `i++` increases the value of the variable `i` by 1.

Line 12: `s = s + i + "
"`;

It will concatenate `s`, `i`, `"
"` together and assign the resulting string back to the variable `s`.

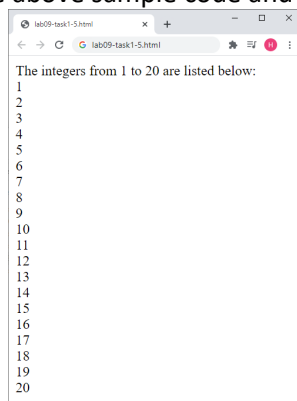
The above for-loop is running as follows:

- a) Set `i` to 1.
- b) Check the value of the continuation condition `i<=N`
If `i<=N` is true, then it will execute the loop action, i.e., the statement(s) inside the curly brackets `{ }`. In this case, there is only one statement (line 12).
After executing the loop action, the increment statement is executed, i.e., `i++`, to increase the variable `i` by 1.
Then it will go back to step b) and this process is repeated until the continuation condition is false.
- c) If `i<=N` is false, then it will not execute the loop action and this is the end of the for-loop. The program will then execute the next statement, i.e., line 14 in this case.

If you trace the for-loop, then you should understand that

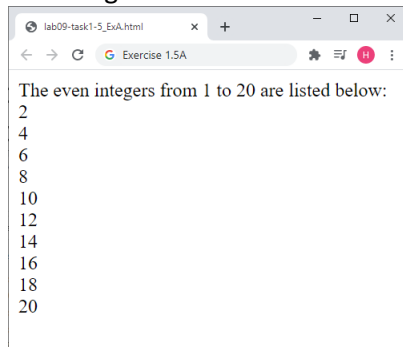
- A) The value of `s` becomes
`"The integers from 1 to 20 are listed below:
1
"`
at the end of the 1st iteration of the for-loop
- B) The value of `s` becomes
`"The integers from 1 to 20 are listed below:
1
2
"`
at the end of the 2nd iteration of the for-loop
You should be able to work out the pattern to see how `s` is changed by appending something to it at each iteration
- C) The continuation condition `i<=N` is true for the initial small values of `i`. This continuation condition will become false when `i>N`, i.e., when `i` is equal to 21 as `N` is equal to 20. In other words, the for-loop will run for a total of 20 iterations for `i=1, 2, 3, ..., 19, 20`.

Run the above sample code and verify that it shows the exact content as your anticipation:



Exercise 1.5:

- A) Modify your program so that it uses the for-loop to list the even integers from 1 to N as shown in the following screenshot:

**Approach 1:**

You can keep the for-loop statement `for(i=1; i<=N; i++)` unchanged, so that `i` still goes from 1 to N and increases by 1 after each iteration. For the loop action, you can add a one-way conditional to execute line 12 only if `i` is even.

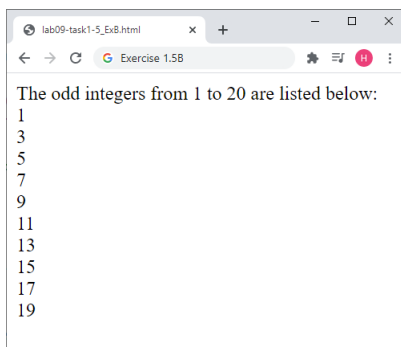
Approach 2:

You change the for-loop statement so that the initial value of `i` is changed to 2 which is the first even number. After each iteration the value of `i` should be increased by 2 (instead of 1) so the increment statement can be changed to `i=i+2`. So the resulting for-loop statement would look like `for(i=2; i<=N; i=i+2)`. In this case, you do not need to change the loop action.

Note: Another way of writing `i=i+2` is `i+=2` where both statements increase the value of `i` by 2.

Try both approaches and verify that you obtain the resulting webpage exactly as shown in the above screenshot.

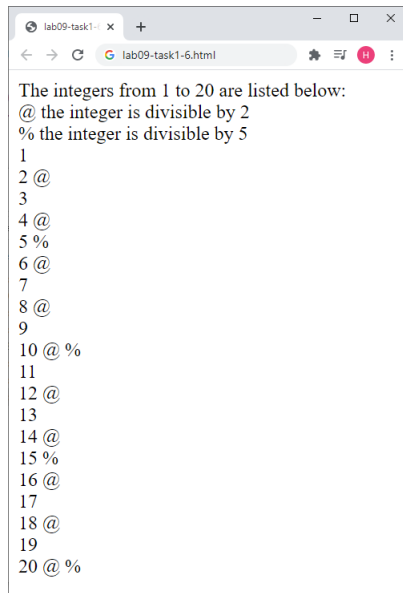
- B) Modify your program so that it uses the for-loop to list the odd integers from 1 to N as shown in the following screenshot:



Try implementing using both approaches introduced in A) and verify that you obtain the resulting webpage exactly as shown in the above screenshot.

Task 1.6 Label an integer if it is divisible by another integer

In Task 1.2 you have written a program to check if an input integer is divisible by another integer. In this task, we would like to list out a series of integers, and give a label to an integer if it is divisible by another integer as shown in the following screenshot:

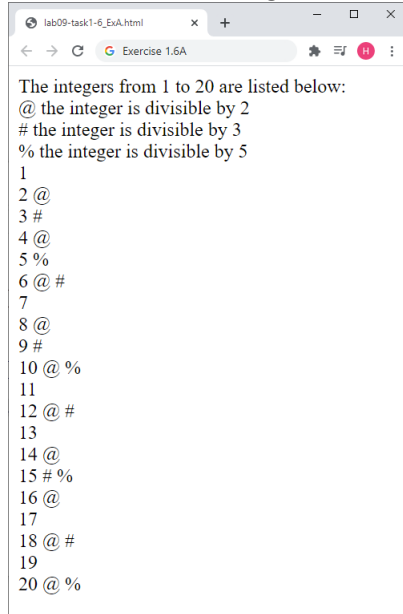


```
The integers from 1 to 20 are listed below:
@ the integer is divisible by 2
% the integer is divisible by 5
1
2 @
3
4 @
5 %
6 @
7
8 @
9
10 @ %
11
12 @
13
14 @
15 %
16 @
17
18 @
19
20 @ %
```

Now take a moment to think about how you can modify the program from Task 1.5 to perform this task. It is important for you to try to come up with the program logic by yourself and enhance your coding skill by trying to solve the problem without referring to any hints at first. A sample program is shown on p.10 for your reference.

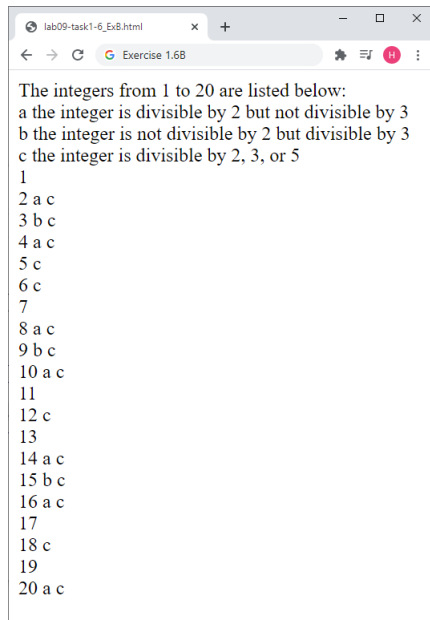
Exercise 1.6:

- A) Modify your program so that in addition to 2 and 5, it also checks if an integer is divisible by 3, as shown in the following screenshot:



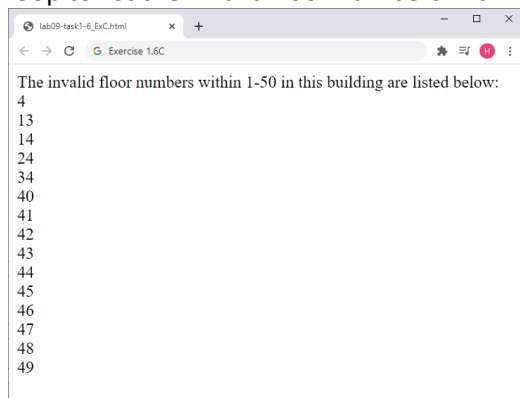
```
The integers from 1 to 20 are listed below:
@ the integer is divisible by 2
# the integer is divisible by 3
% the integer is divisible by 5
1
2 @
3 #
4 @
5 %
6 @ #
7
8 @
9 #
10 @ %
11
12 @ #
13
14 @
15 # %
16 @
17
18 @ #
19
20 @ %
```

- B) Modify your program so that it labels a) if an integer is divisible by 2 but not divisible by 3; b) if an integer is not divisible by 2 but divisible by 3; c) an integer is divisible by 2, 3 or 5, as shown in the following screenshot:



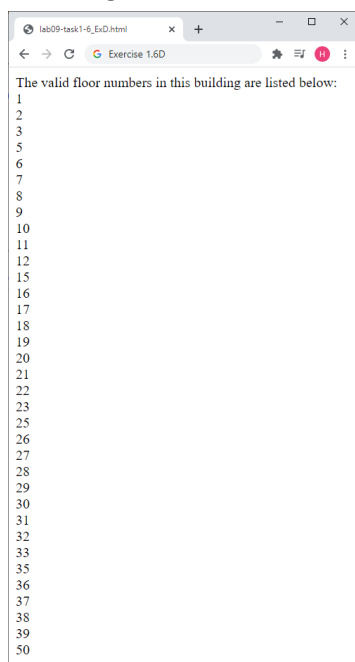
```
The integers from 1 to 20 are listed below:  
a the integer is divisible by 2 but not divisible by 3  
b the integer is not divisible by 2 but divisible by 3  
c the integer is divisible by 2, 3, or 5  
1  
2 a c  
3 b c  
4 a c  
5 c  
6 c  
7  
8 a c  
9 b c  
10 a c  
11  
12 c  
13  
14 a c  
15 b c  
16 a c  
17  
18 c  
19  
20 a c
```

- C) In Task 1.3 you have checked if the user input a valid floor number. Here write a program to use a for-loop to list the invalid floor numbers within 1-50, as shown in the following screenshot:



```
The invalid floor numbers within 1-50 in this building are listed below:  
4  
13  
14  
24  
34  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49
```

- D) Modify your program from C) such that it lists the valid floor numbers in the building, as shown in the following screenshot:



```
The valid floor numbers in this building are listed below:  
1  
2  
3  
5  
6  
7  
8  
9  
10  
11  
12  
15  
16  
17  
18  
19  
20  
21  
22  
23  
25  
26  
27  
28  
29  
30  
31  
32  
33  
35  
36  
37  
38  
39  
50
```

A sample HTML and Javascript code corresponding to an implementation according to the requirements of Task 1.6 is shown below for your reference:

```
1 <!DOCTYPE html>
2
3 <html>
4 <head>
5   <title></title>
6   <script>
7     function init() {
8       var i, N, s;
9       N = 20;
10      s = "The integers from 1 to "+N+" are listed below:<br/>";
11      s = s + "@ the integer is divisible by 2<br/>";
12      s = s + "% the integer is divisible by 5<br/>";
13      for (i=1; i<=N; i++) {
14        s = s + i;
15        if (i%2==0) {
16          s = s + " @";
17        }
18        if (i%5==0) {
19          s = s + " %";
20        }
21        s = s + "<br/>";
22      }
23      document.getElementById("result").innerHTML = s;
24    }
25  </script>
26 </head>
27 <body onload="init();">
28   <div id="result"></div>
29 </body>
30 </html>
```

Task 2 Review Previous Labs

You can review the previous labs by reading and following the lab instructions, and check that you are able to complete the lab tasks and exercises by yourself. If you have any questions, you can ask the lab tutor or TA during the lab session.

Task 3 Complete the assessment from the Canvas course page

You should complete the [Lab 09 Assessment](#) from the Canvas course page before the posted deadline.

Task 4 Challenge your classmates

You can first reflect on what you have learnt in this lab, and then come up with problems to challenge your classmates. You can post your problem on the Canvas course page, under this [Discussion page](#). One should be able to solve your problem by using what he/she learns in Lab 09. You will not get extra marks by posting a challenging problem or solving a challenging problem posted by another student, but you will earn your fame so that you can impress the course leader, the lab tutors, and your classmates.
