EECS 1012: LAB #7 – More Computational Thinking (Feb 25 — March 3, 2019)

#Important reminders

1) You **must** attend **your assigned** lab session (we will be marking your submission in the lab).
2) Do the mini-quiz prelab quiz. It will open on Wednesday by 13:00 and will expire on Thursday by 12:59. It's part of your lab grade. Each lab counts to 2% of your overall grade.
3) You must arrive on time – anyone later than 15 minutes may not be admitted to the lab.
4) Submitting your lab work to Moodle is required. Otherwise, you receive 0 even if you show the results to your TA.

#Important pre-lab works you need to do before going to the lab

1) Download this lab files and read them carefully to the end.
2) You should, by now, be comfortable with understanding/reading basic flowcharts.
3) don't forget to do the mini-quiz by Thursday at 12:59pm.

1. GOALS/OUTCOMES FOR LAB

- To practice more on computational thinking and implement it in js

2. LAB – TASK

1) You first task is to create a `lab7<yourname>.html` file (supported by a `lab7<yourname>.css` file),in which there are 5 buttons with captions Problem 1, Problem 2, …, Problem 5.
2) Then, you implement 5 flowcharts in your lab7<yourname>.js

3. SUBMISSIONS

1) Manual verification by TA

You will need to have one of the TAs verify your lab before submission. The TA will look at your various files in their progression. The TA may ask you to make minor modifications to the lab to demonstrate your knowledge of the materials.

The TA will mark your name off a list and ask you to sign that you have been verified.

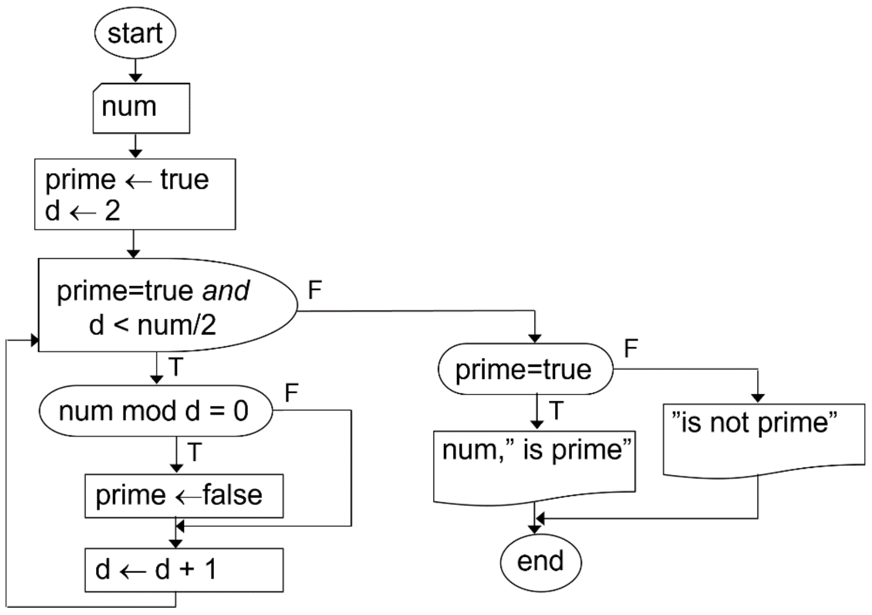
2) Moodle submission

You will see an assignment submission link on Moodle. Create a **folder** named "**Lab7**" and copy all of your 3 files (html, css, and js) inside. This folder should be compressed (or tar.gz on the VirtualBox machines) and the compressed file submitted.

Implementing 5 flowcharts:

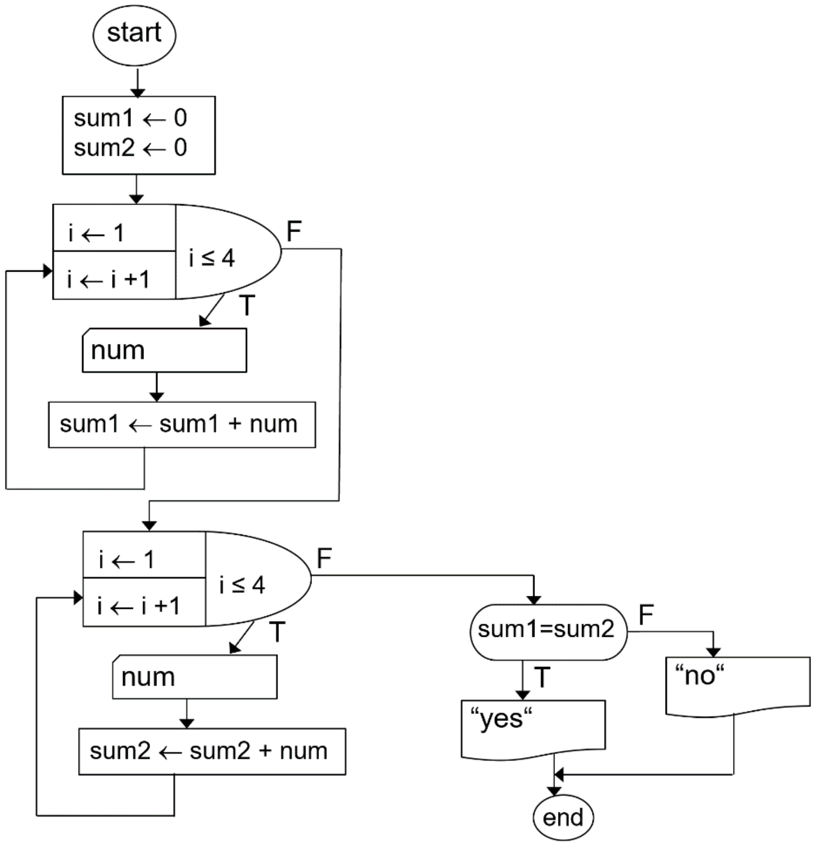
Note. it's a good practice to have comments in all your coding, such as in HTML, CSS, and JS.

Function P1`<yourname>`. When button "problem 1" is clicked, a JavaScript function equivalent to the following flowchart should execute. In other words, your task, here, is to translate the following flowchart to JavaScript code in P1`<yourname>`:



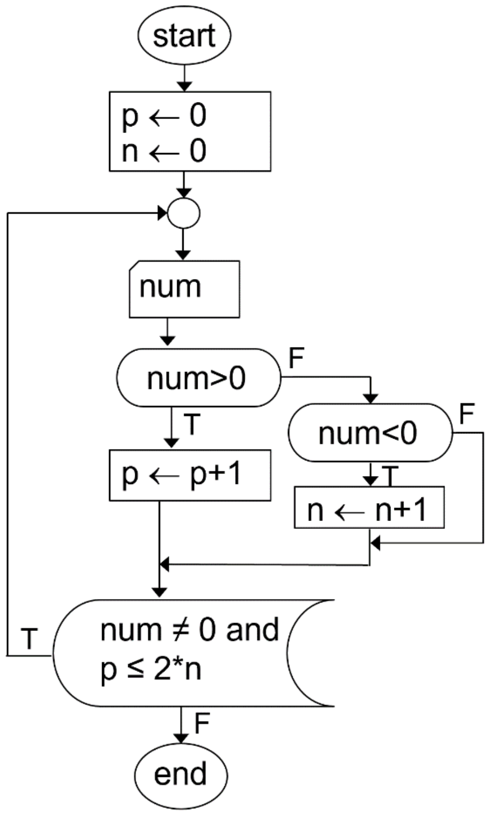
Hint: for input, use the `prompt()` function and for output, use the `alert()` function. Note: for translating this flowchart to JavaScript, you do not really need to know what the flowchart does. Yet, you know this flowchart receives a natural number greater than 1 and determines if it's a prime number or not.

Function P2<yourname>. When button "problem 2" is clicked, a JavaScript function equivalent to the following flowchart should execute. In other words, your task, here, is to translate the following flowchart to JavaScript code in P2<yourname>:



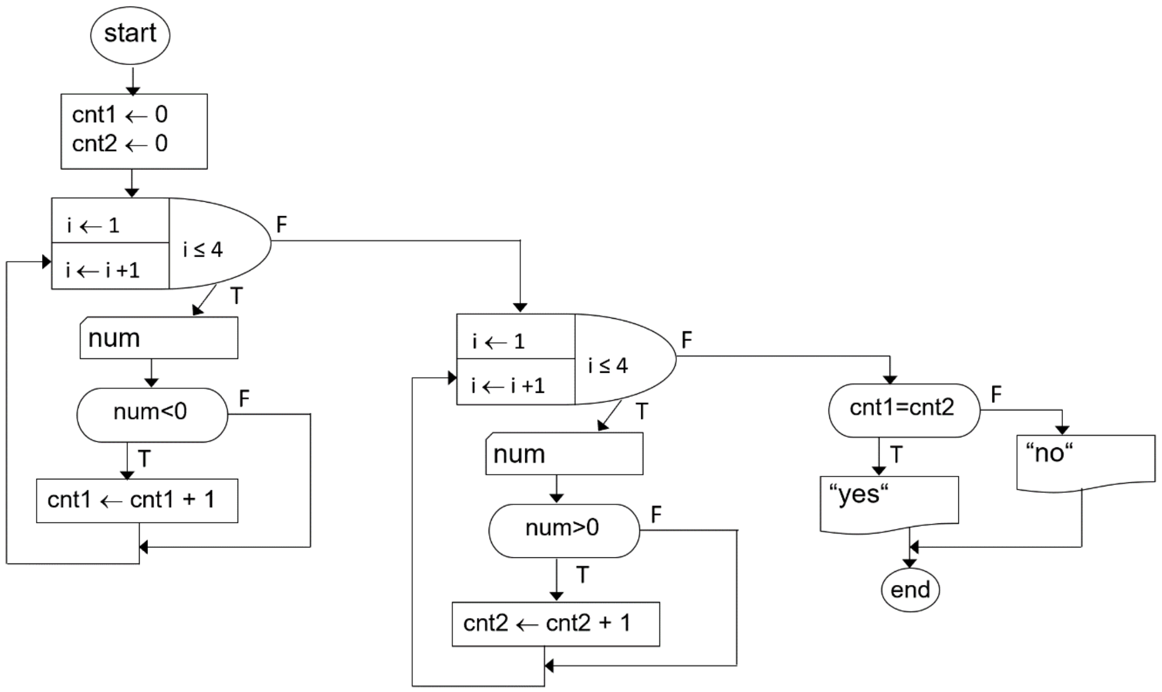
Hint: for input, use the `prompt()` function and for output, use the `alert()` function. Note: for translating this flowchart to JavaScript, you do not really need to know what the flowchart does. Yet, you perhaps know this flowchart receives 8 numbers and outputs "yes" if sum of the first 4 numbers is equal to sum of the last 4 numbers.

Function P3<yourname>. When button "problem 3" is clicked, a JavaScript function equivalent to the following flowchart should execute. In other words, your task, here, is to translate the following flowchart to JavaScript code in P3<yourname>:



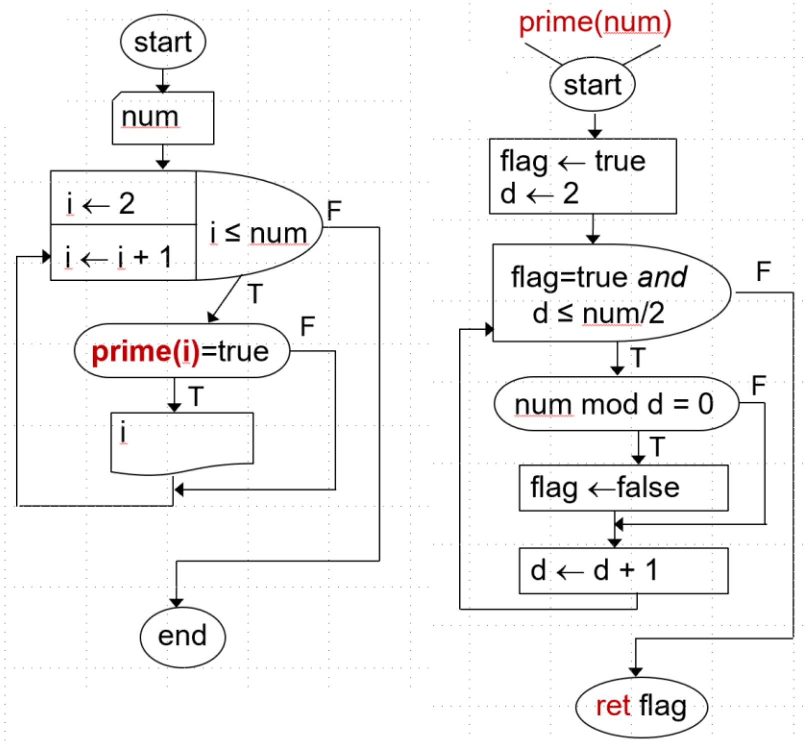
Hint: for input, use the `prompt()` function and for output, use the `alert()` function. Note: for translating this flowchart to JavaScript, you do not really need to know what the flowchart does. Yet, you know this flowchart receives natural numbers until a 0 is entered or number of posive values is more than twice negative ones.

Function P4<yourname>. When button "problem 4" is clicked, a JavaScript function equivalent to the following flowchart should execute. In other words, your task, here, is to translate the following flowchart to JavaScript code in P4<yourname>:



Hint: for input, use the `prompt()` function and for output, use the `alert()` function. Note: for translating this flowchart to JavaScript, you do not really need to know what the flowchart does. Yet, you perhaps know this flowchart receives 8 numbers and outputs "yes" if number of negative values in the first half is equal to the number of positive values in the second half..

Function P5<yourname>. When button "problem 5" is clicked, a JavaScript function equivalent to the flowcharts illustrated in Slids 49 and 50 of Computational Thinking should execute. (that's the flowchart that calls function prime to print all prime numbers less than or equal to a certain input.)


start
num
i ← 2
i ← i + 1
i ≤ num
F
T
prime(i)=true
F
T
i
end
prime(num)
start
flag ← true
d ← 2
flag=true and d ≤ num/2
F
T
num mod d = 0
F
T
flag ← false
d ← d + 1
ret flag