

EECS 1012: LAB 07 Tracing Algorithms; using Sub-Algorithms (Nov 4-8, 2019)

A. REMINDERS

- 1) You **must** attend your **assigned** lab session (we will be marking your submission in the lab).
- 2) You must arrive on time: anyone later than 15 minutes may not be admitted to the lab.
- 3) You must complete the pre-lab mini quiz posted on Moodle in the first 15 minutes of your lab time.
- 4) Each lab including the pre-lab mini quiz is about 2% of your overall grade.
- 5) TAs are in the lab to help you. They will also verify and mark your work at the end. Signal a TA for help if you stuck on any of the steps below. Yet, note that TAs would need to help other students too.

B. 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) Practice tracing algorithms with different sample inputs. There are several examples in Lect07.
- 3) In Slides 22 and 23 of Lect07 notes, there is an example of sub-algorithm. Review it and implement it in JavaScript once with and once without using sub-algorithms.

C. GOALS/OUTCOMES FOR LAB

- To practice **computational thinking** by first drawing flowcharts for basic computation problems, followed by implementation in JS

D. TASKS

- 1) Your first task in this lab is to verify 5 algorithms, in Exercises 1 to 5, by tracing them for some sample inputs. You also provide pre- and post- conditions for each algorithm. In Exercise 6, you draw a flowchart that includes two sub-algorithms. These tasks must be done in teams of two. (By the permission of the TA, only one team can be in team of three if the lab population is odd.) While you are done, you should show your trace tables, pre- and post- conditions for all flowcharts, as well as your solution to Exercise 6 to your TAs before you go to the next part. The TA may ask you to make minor modifications to demonstrate your computational thinking skills in those contexts.
 - Note you should NOT open VS-Code or browsers before finishing Task 1.
- 2) In Part 2, you implement all 6 exercises in JavaScript. You should create one html, one css, and one js file for all your work, and you demo it to the TA. See next pages for further instructions.

E. SUBMISSIONS

- 1) Manual verification by a TA

You will need to have one of the TAs of your lab to verify your work 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; You are required to sign the list to show your attendance and that you have been verified.

- 2) Moodle submission

You will see an assignment submission link on moodle. Create a **folder** named "**Lab07**" and copy all your lab materials inside (`img_{01,02,03,04,05,06}.jpg`, `Lab07<yourName>.html`, `Lab07<yourName>.css` and `Lab07<yourName>.js`). This folder should be compressed (or tar.gz) and the compressed file submitted.

F. COMPUTATIONAL THINKING

Part 1: This part must be done in teams of two. (By the permission of the TA, only one team can be in team of three if the lab population is odd.) If you have done it at home, you are required to discuss it with a peer from your lab before you show your final solution to your TA.

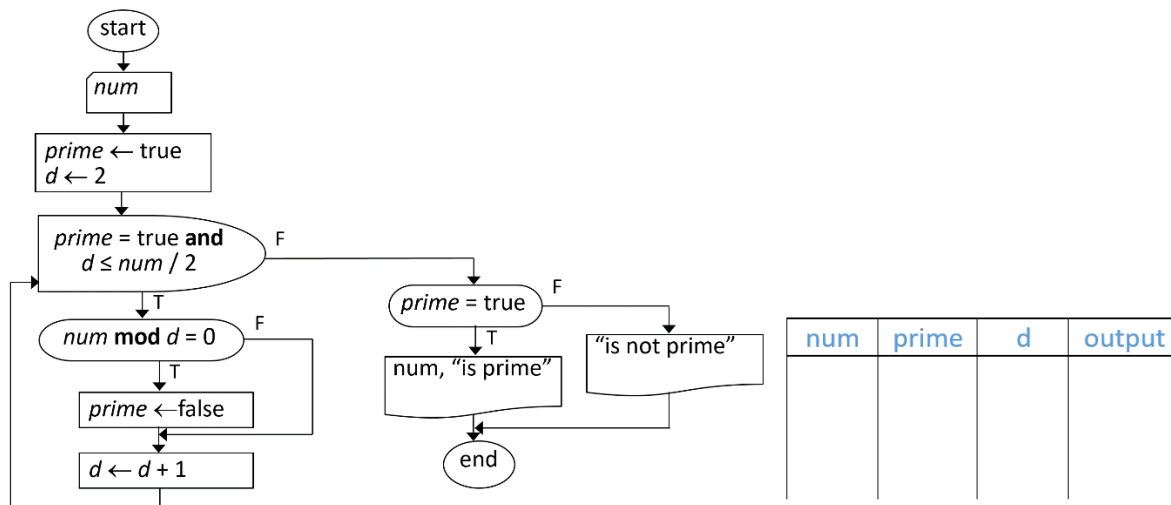
In Exercises 1 to 5, you verify the correctness of all algorithms by tracing them with some sample inputs. You also write pre- and post- conditions for each algorithm. In Exercise 6, you devise an algorithm that calls two sub-algorithms. In that Exercise you trace your new sub-algorithms for some sample inputs.

Note that terms algorithm, sub-algorithm, function, method, programs are used interchangeably, yet they have subtle differences.

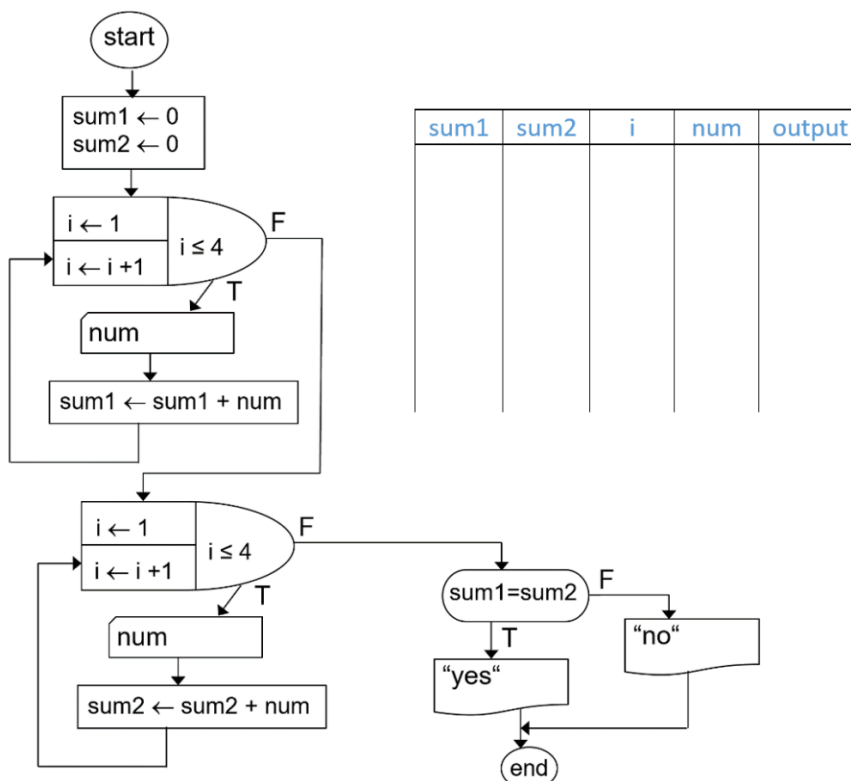
For each exercise, take one picture of all your trace tables and pre- post- conditions, including your name. By end of this lab, both you and your teammate should submit these pictures to moodle as **img_{01,02,03,04,05,06}.jpg** files, where **img_x** is the picture for exercise **x** below. Make sure the size of each image is less than 500KB, e.g. by reducing the resolution of your camera.

IMPORTANT: You are required to provide preconditions and postconditions for each solution you provide.

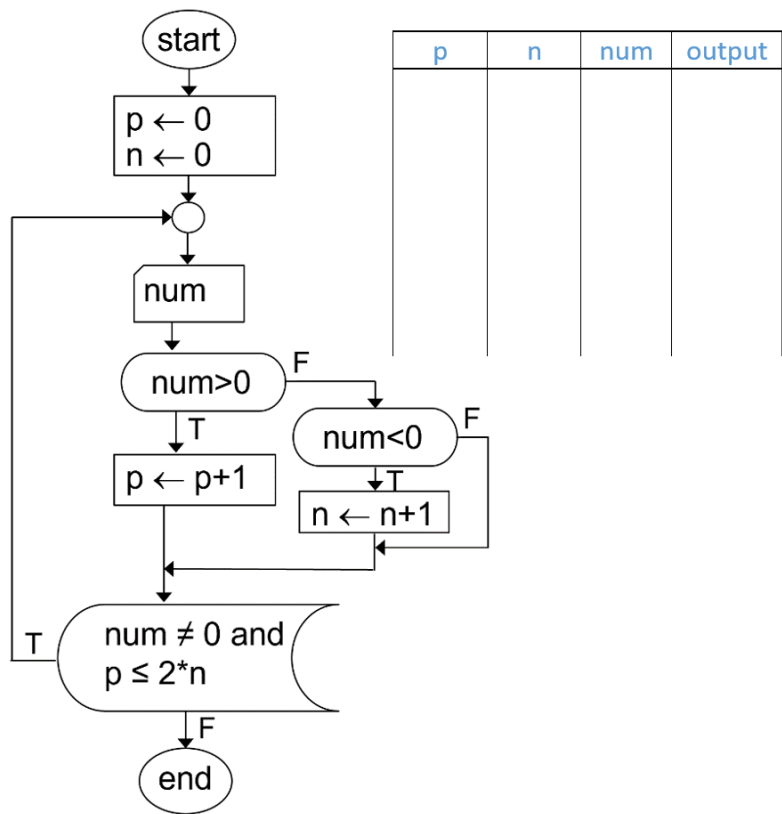
- Ex 1) Trace the following flowchart for when input is 9 and complete the trace table. Also, write pre- post-conditions for this flowchart.



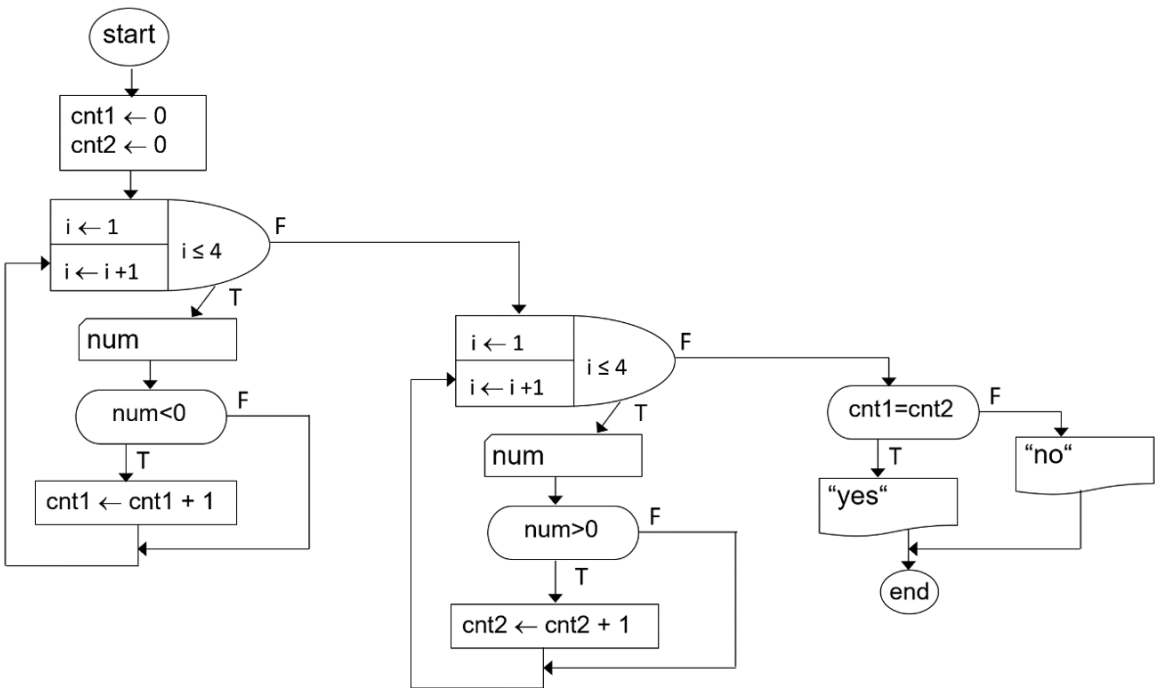
- Ex 2) Trace the following flowchart for when input values are 12, 3, 3, 6, 11, 14, 1, 1, and complete the trace table. Also, write pre- post-conditions for this flowchart.



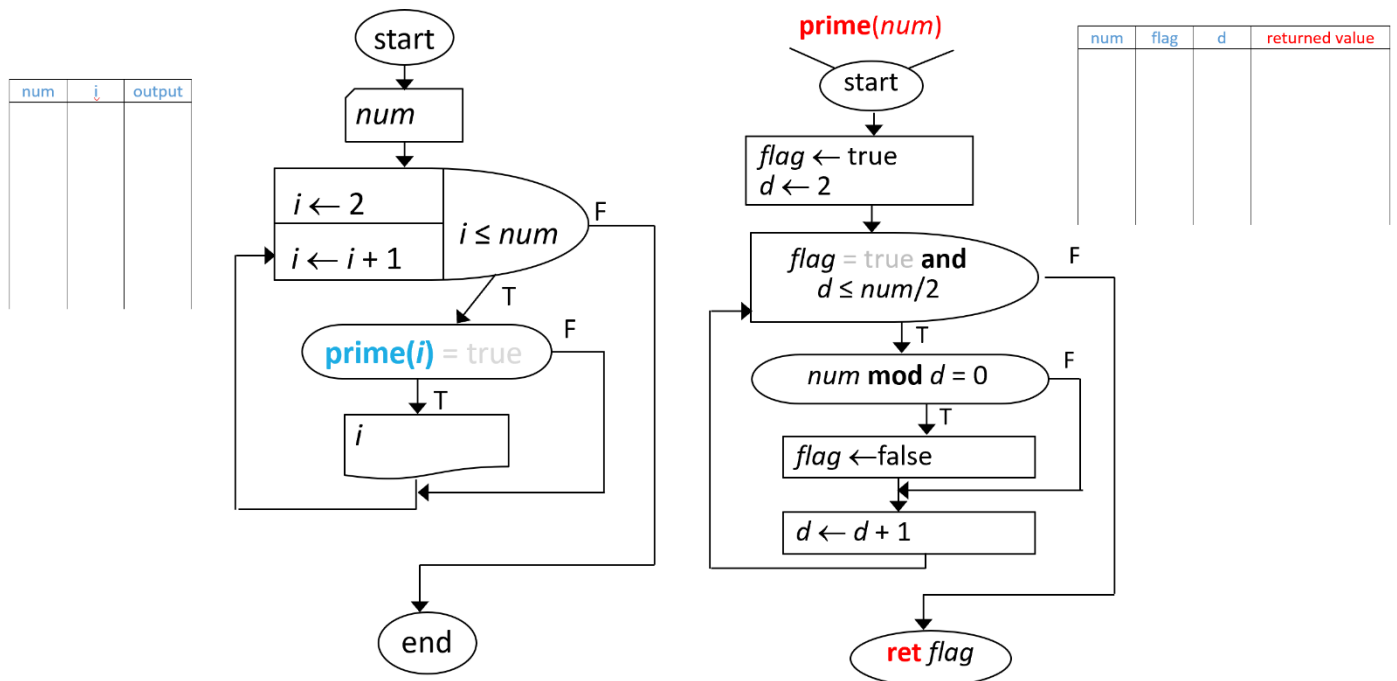
Ex 3) Trace the following flowchart for when input values are 12, 3, -3, 6, -11, -14, -1, -1, -3, 2, 0 and complete the trace table. Also, write pre- post-conditions for this flowchart.



Ex 4) Trace the following flowchart for when input values are 12, 3, -3, 6, 11, 3, -1, -1 and complete the trace table. Also, write pre- post-conditions for this flowchart.



Ex 5) This exercise uses sub-algorithms. Trace it for when input value is 6 and complete the trace tables. Also note that there are two variables name num, one in the main algorithm and another in the sub-algorithm. Write pre- post-conditions for each of these algorithms (the main one and the sub-algorithm).



Ex 6) Devise an algorithm to receive a positive number, n, and output all prime numbers that are smaller than n and have a digit 7. For example, if n is 100, the program should output 7, 17, 37, 47, 67, 71, 73, 79, and 97.

Your solution should have a main algorithm and two sub-algorithms, let's call them prime(num) and has7(num). You may reuse the prime sub-algorithm of Ex 5. You do not need to verify (e.g. by tracing) it though because you did it in Ex 5. You need to trace your has7 algorithm. You should provide pre- and post-conditions for all your 3 (sub)algorithms.

Show all your trace tables as well as pre- post- conditions and your solution of Exercise 6 to your TA before starting Part 2. The TAs may ask any of the teammates some questions to verify your level of knowledge and skills on each exercise, they may also ask you to verify the flowcharts for different sample inputs. They will provide you a grade accordingly.

Part 2: Create a lab07<yourname>.html file supported by a lab07<yourname>.css and a lab07<yourname>.js files. In your html file, you should have 6 buttons with captions Problem 1, Problem 2, ..., Problem 6. Then, you implement all 6 flowcharts in your lab07<yourname>.js . When each button is clicked, the event should execute the corresponding algorithm that you have in your js file.

G. AFTER-LAB WORKS (THIS PART WILL NOT BE GRADED)

In order to review what you have learned in this lab as well as expanding your skills further, we recommend the following questions and extra practices:

- 1) Revisit other flowcharts you have drawn/seen in the course, and trace them for some sample inputs.
- 2) Devise and algorithm to receive a positive number, n, and output all prime numbers that are smaller than n and all their digits are prime too. For example, if n is 100, the program should output 2, 3, 5, 7, 23, 37, 53, and 73.

Please feel free to discuss any of these questions in the course forum or see the TAs and/or Instructors for help.