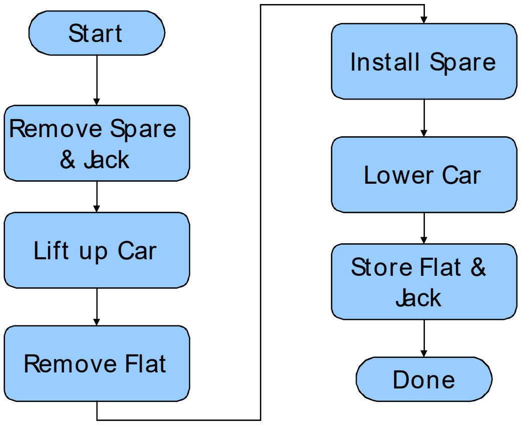
Spring 2014 CSC 150

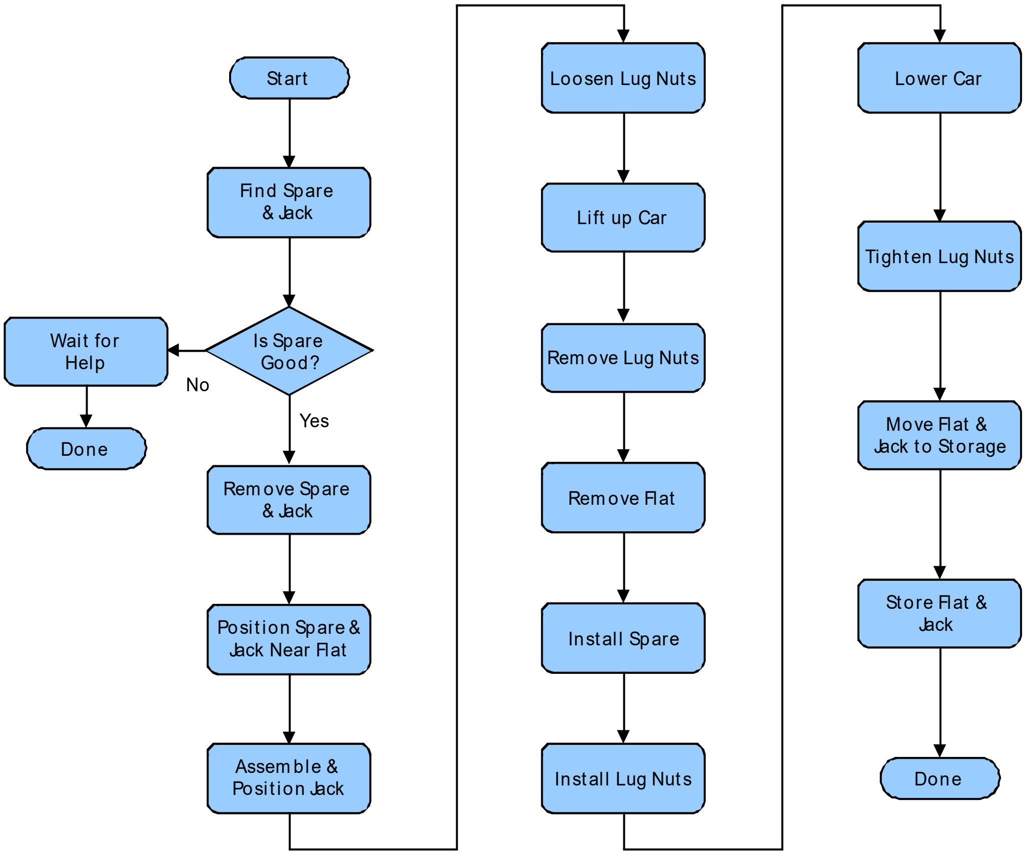
Prelab #2: Algorithm Development

- Programmers develop software by analyzing the problem or task, developing a solution, writing the program (code the solution), testing the program, debugging it, documenting it, and maintaining it.
    - That is, you must have an algorithm before you code it with a high level language (HLL)
- An algorithm can be represented by a flowchart or by pseudocode.
    - In either case, the algorithm may be refined to simplify its translation to source code.

The following flowcharts show an initial algorithm followed by a refined (more elaborated) algorithm for changing a flat tire:

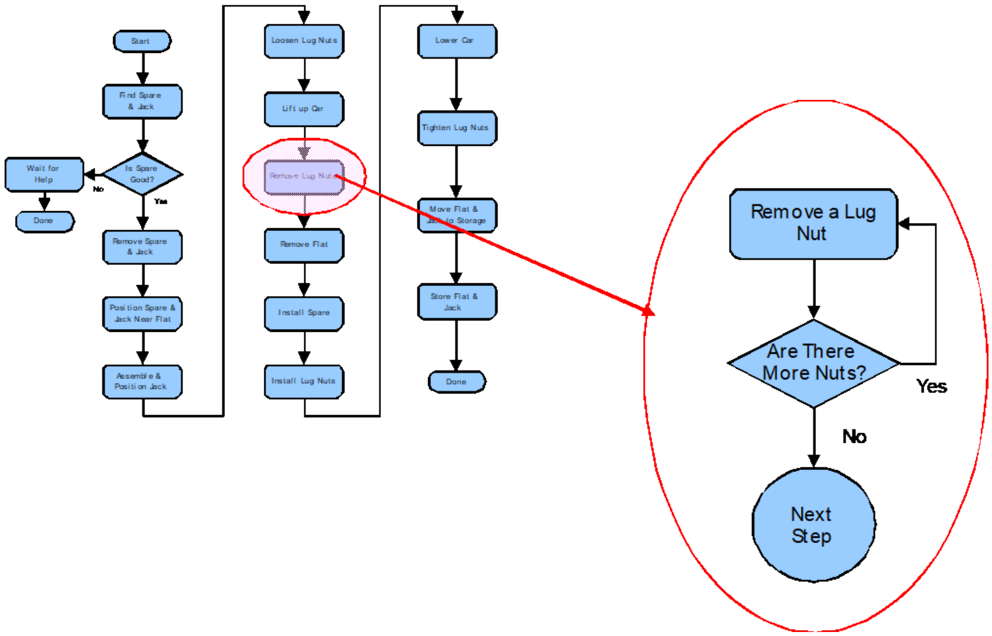


Initial flowchart for changing a flat tire



Refined flowchart for changing a flat tire

Detailed steps for a particular processing step on the refined flowchart are shown next.

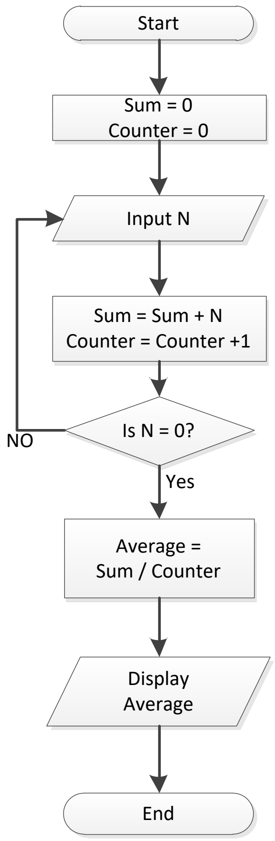


Detailed steps for removing Lug Nuts

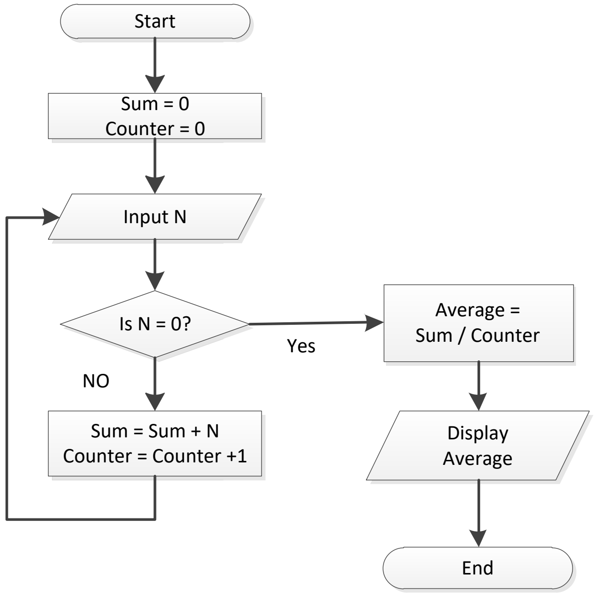
Exercise 1

The following flowcharts depict algorithms to find the average of nonzero numbers entered by a user. When a user enters a zero as a signal to stop, the program calculates and displays the average, then halts. Which one is correct? (Do not be concerned about a possible division by zero problem if the first entry is 0 – we'll discuss that at a later time.)

Solution A



Solution B



Exercise 2

Write the pseudocode for the correct algorithm of exercise 1.

Exercise 3

Show the flowchart and pseudocode for an algorithm that finds and displays the sum of the first five positive, non-zero integer numbers.

Exercise 4

A particle moves along a straight line. Its position, **x**, as a function of time is given by the following mathematical expression: **x** = $3t^3 – 2t^2 + 5$ where **x** represents the distance in meters traveled at time **t** in seconds. The velocity is computed as **v** = $9t^2 – 4t$ .Write an algorithm that computes and displays the position and velocity of the particle at time 1 second, 2 seconds,… up to **n** seconds, where **n** is the time when the velocity, **v**, of the particle has just reached or exceeded **Y** m/s. The algorithm must get the value of **Y** from the user. If **Y** is not positive the algorithm displays an error message and ends the program. Show the **pseudocode and flowchart** of the algorithm. For this exercise, the computations can be represented by simple statements "compute x" and "compute v".