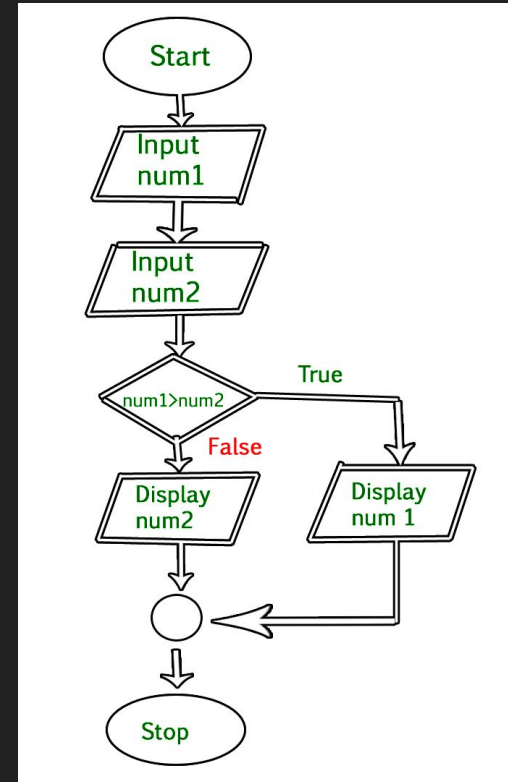


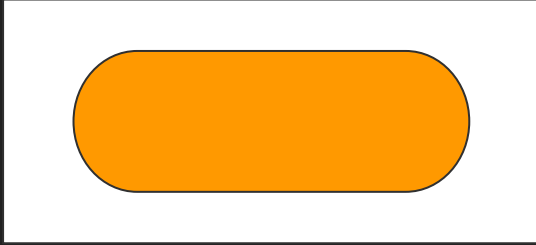
# Flowcharts and Pseudocode

# What are Flowcharts?

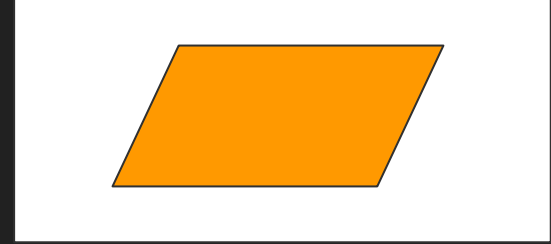
- Flowcharts act as a visualization of the coding design process
- They also help establish the logical flow of the code before the actual code is written



# How to build a flowchart



Terminator: The start  
and end of the code

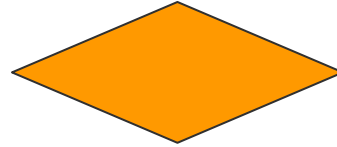


I/O: handles  
input/output

# How to build a flowchart



Processing: Arithmetic operations

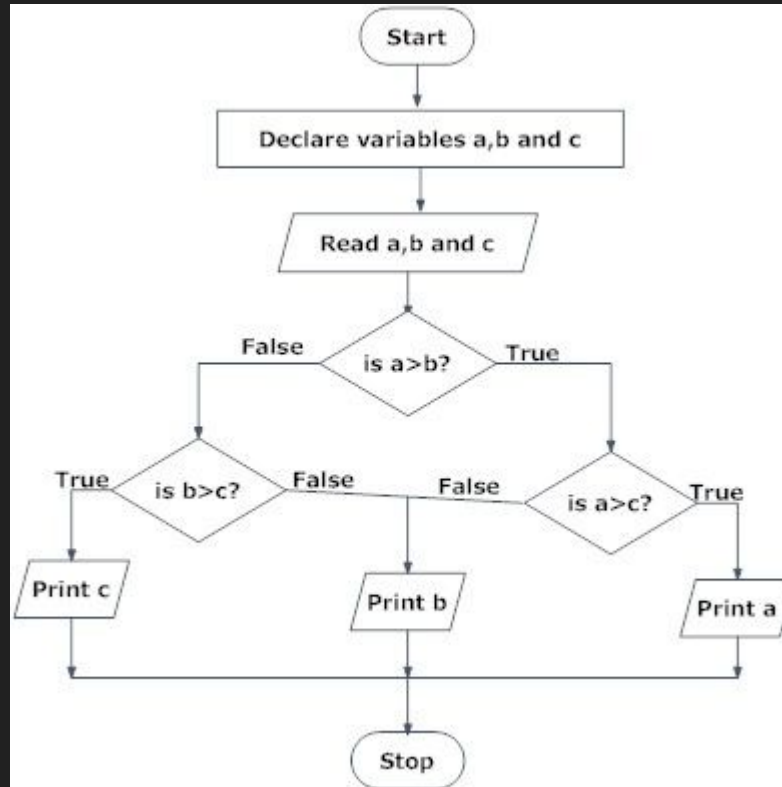


Decision: Boolean condition; has two outputs, T/F

## Example: Flowchart for the largest of 3 integers

Let's practice! Let's try to draw the flowchart for a program that would output the largest of three integers entered by the user.

# Example: Flowchart for the largest of 3 integers



# Example: Highest Exponent

- Let's now design a flowchart for another problem.
- Given a base and a limit, return the highest power the base can be raised to before crossing the limit.

# Pseudocode

- Pseudocode is a method of “coding” by writing the code in structured English.
- The goal of pseudocode is to write something that is in between English and computer code.
- This means that coding structures, such as functions, loops, and if-else statements should be present in your pseudocode, but actual function calls or math should typically not.



# Why should you use Pseudocode?

- “If I’m making the effort to write my code in English, why should I not just code it up?”
- Pseudocode, much like block outlines of essays, helps organize your thoughts more fully, and can help point out areas of your code where you may want to research more, such as trying to find pre-built functions that perform certain actions.

# Pseudocode Example

- Ex1: If you are trying to find the highest exponent before crossing a limit the pseudocode for that problem might be:

“Given the base and the limit, set power to 0. While  $\text{base}^{\text{power}}$  is less than limit, add one to power. Return power.”

- Ex 2: If you are given a list and asked to return a list of the square roots of the perfect squares:

“Given an list, for each index in the list, compute the square root. If the result is not an integer, remove it. Return the resulting list.”