F.2 Computer Literacy
Coding with App Inventor 2 - Assignment 4

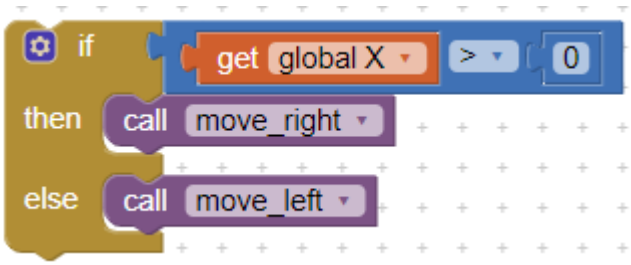
Flowcharts of conditional and looping blocks in App Inventor

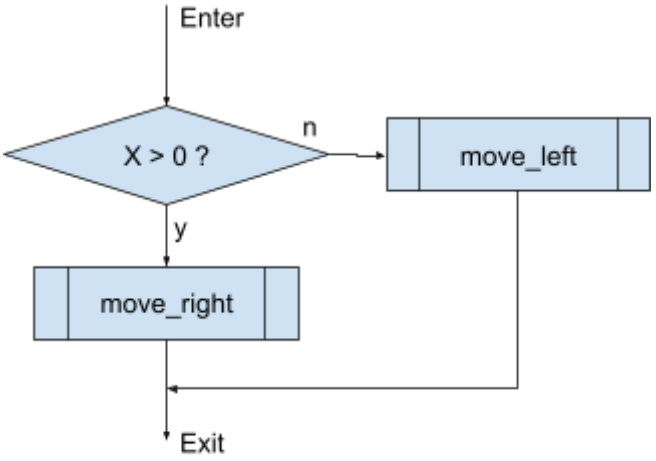
There are various looping blocks in App Inventor that are used to repeat actions. Some of them repeat the actions for a fixed number of times, others repeat until certain conditions are met.

When converted to flowcharts, they can look similar to conditional (if-then-else) blocks. Pay attention to the minor differences between them.

Conditional blocks

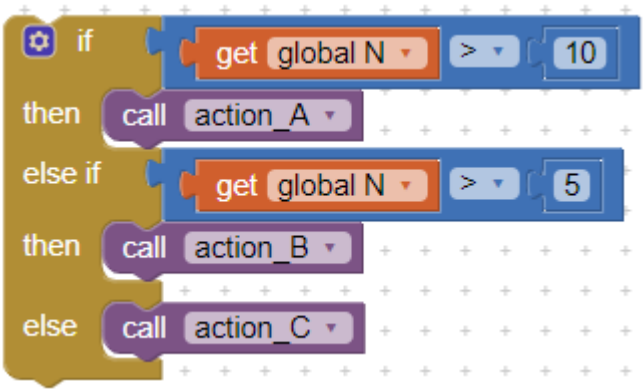
e.g.

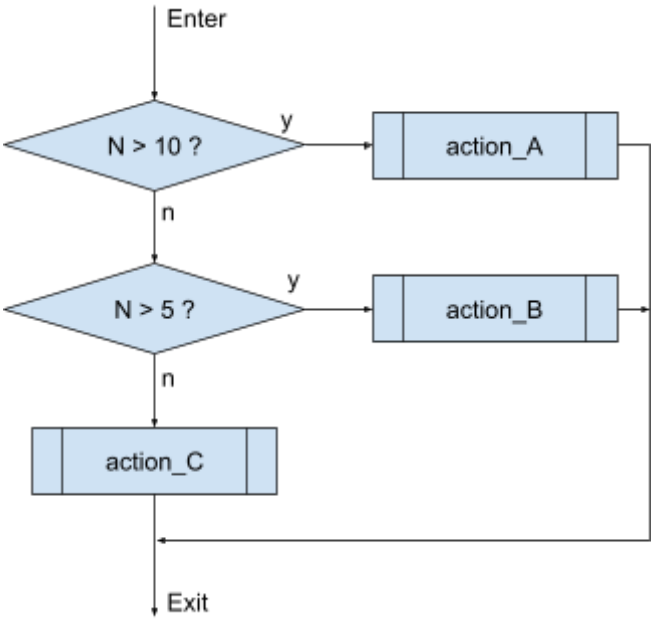




The program takes either one of the two paths (branches). It always goes from Enter to Exit without any repetition

A more complicated example:

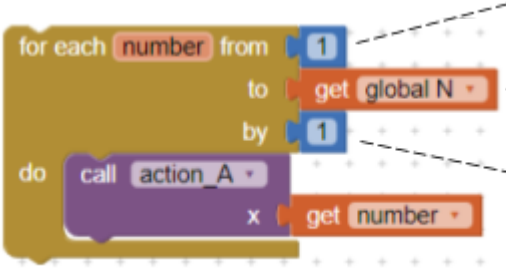


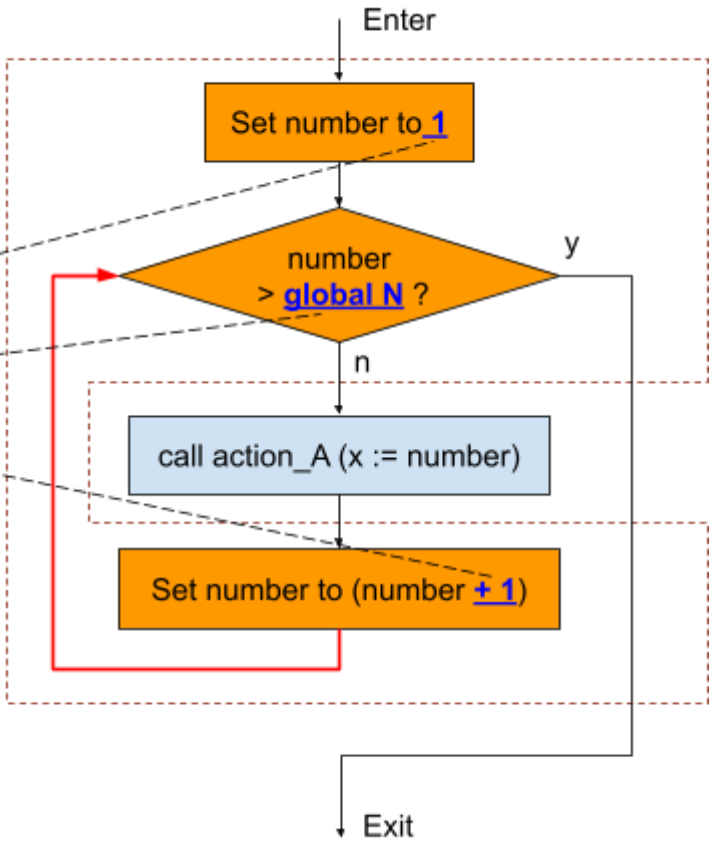


Looping (iteration)

Repeat **for a fixed number of times**

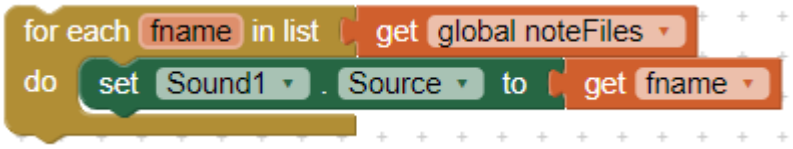
e.g.

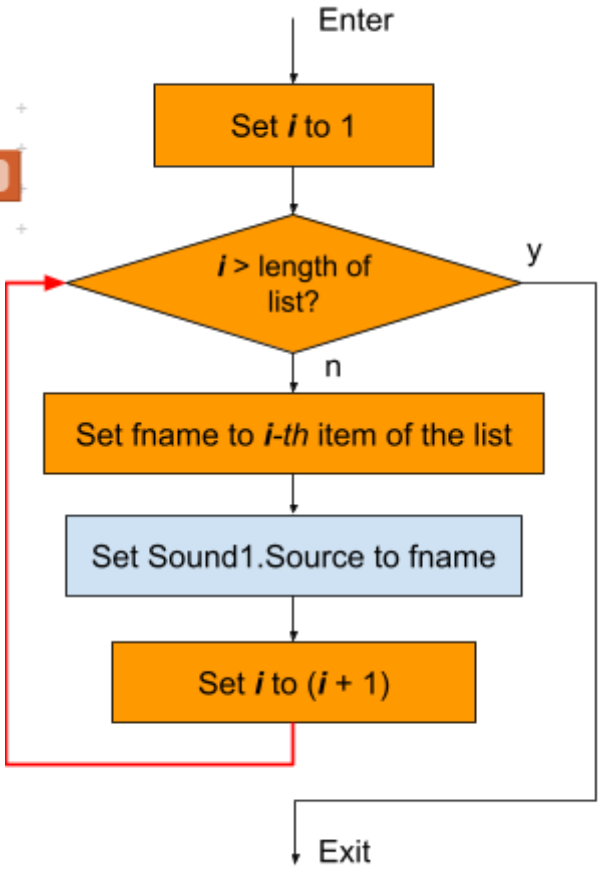




Before it reaches Exit, the [call action_A] block is repeated for *N* times.

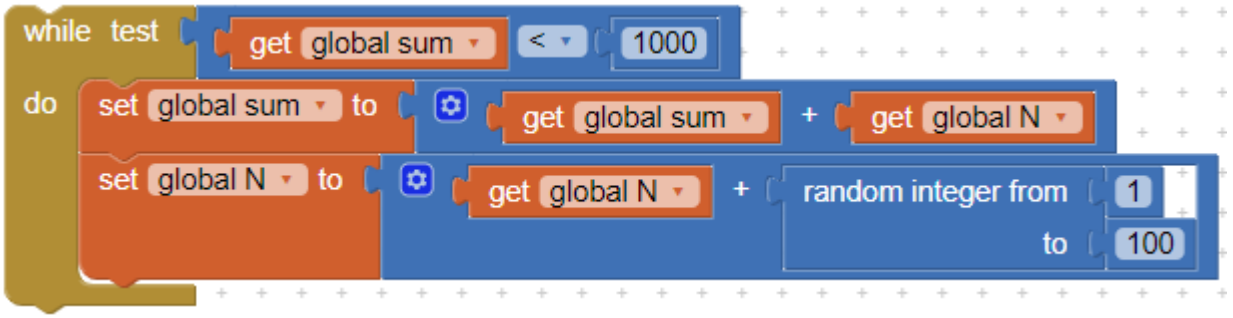
e.g.

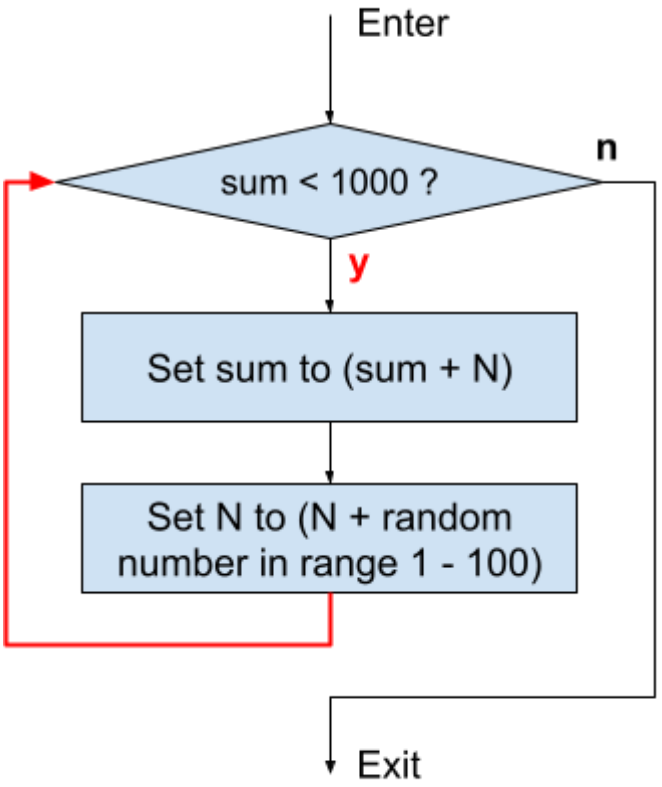




Repeat **for indefinite number of times** depending on a condition

e.g.





Note:

<u>The red lines in the above flowcharts can help you to identify the parts as a form of looping that should be constructed with looping (for or while) blocks, and should NOT be mistaken as a conditional (if-then-else) block.</u>

Exercise

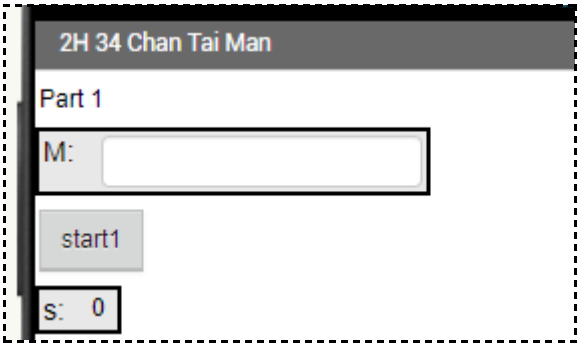
In this exercise, you just need to create simple user interface elements to investigate the workings of flowcharts.

Name your project as **Lesson4**_*class*_*classno* with your actual class and class no.

Part 1:

Interface elements:

- An input box for the value M in the following flowchart. Put a label on the left of the input box by grouping them in a horizontal arrangement.
- A start button (start1) that executes the flowchart when the button is clicked.
- Labels for the value of s in the flowchart.



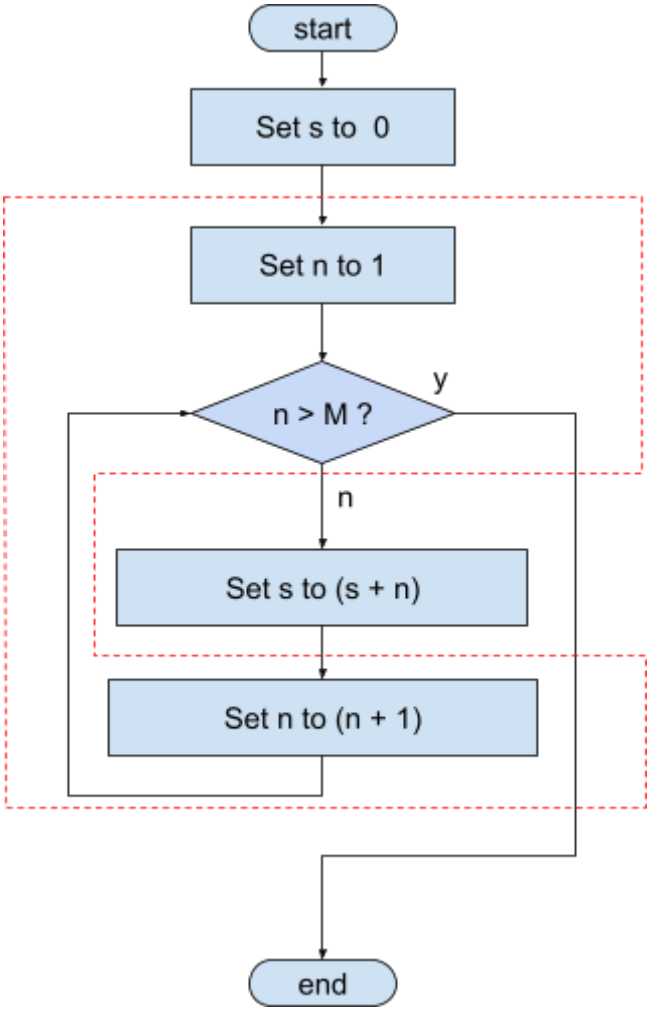
Blocks:

- Convert the flowchart into blocks and put them into the click-event handler of the start button.

Note: you may use the **.text** property of the labels for the values of s and n directly for calculations. That way you will not need to define global variables (i.e. use the interface elements' properties as places to store these values)

(Hint: use a [for each number from] block for the parts enclosed in red-dotted line)

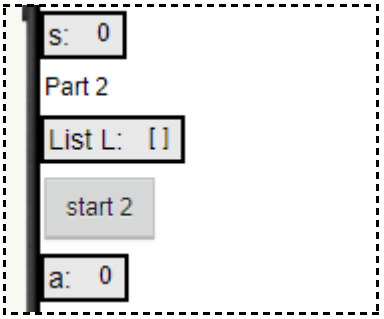
What is the purpose of this algorithm?



Part 2

Interface elements:

- Another start button (*start2*) for execution of the flowchart when clicked.
- A label to show values of list L.
- Label for value of a (re-use the label of s in Part 1 for value of s)

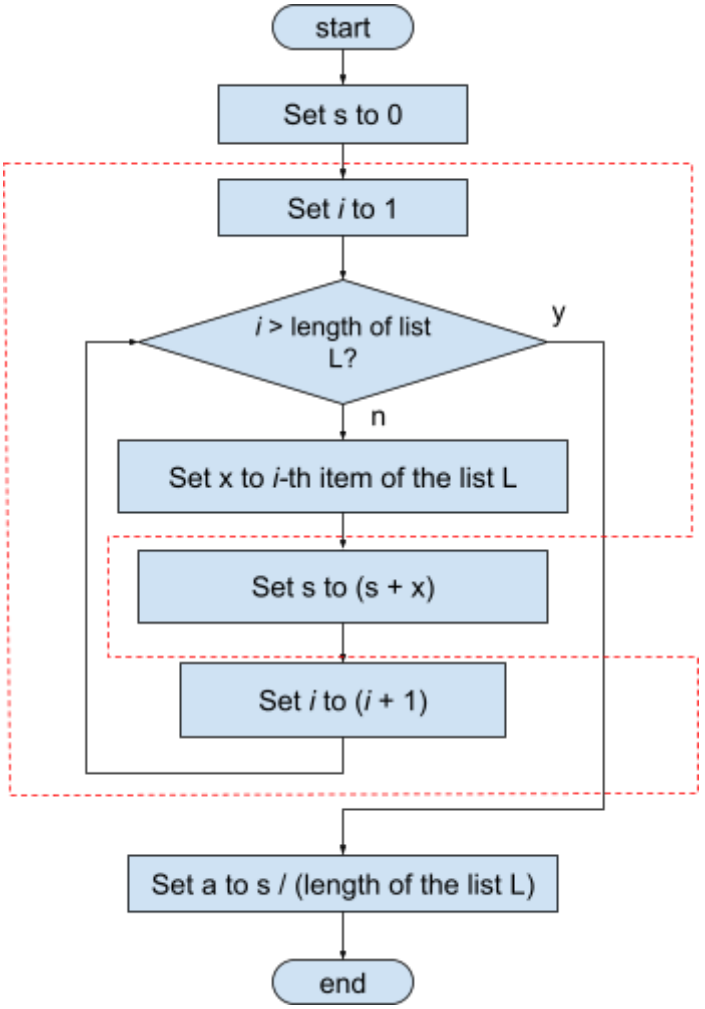


Blocks:

- Create a **global variable L** to store a list of about 10 values, including positive and negative
- When screen 1 is initialized, show values in list L in the label for L.
- Convert the flowchart into blocks and put them into the click-event handler of *start2*.

(Hint: use a [for each name in list do] block)

What is the purpose of this algorithm?



Challenge 1:

How to modify the blocks so that the step “Set s to (s + x)” is executed only if x is positive?

Challenge 2:

How to further modify the blocks such that the loop stops adding the rest of the items in L to s once the value of an item is equal to -999 ?

Export your **Lesson4_*class*_*classno*.aia** file and submit it on eClass.