A7 - FLOWCHARTS AND ALGORITHMS - WRITTEN EXERCISE

#10

School of Computing and Information Technologies

CLASS NUMBER: #10 SECTION: A412

NAME: Jandayan, Jessa A. DATE: Nov. 6, 2019

General Rules for Flowcharting

1. All boxes of the flowchart are connected with Arrows. (Not lines)
2. Flowchart symbols have an entry point on the top of the symbol with no other entry points. The exit point for all flowchart symbols is on the bottom except for the Decision symbol.
3. The Decision symbol has two exit points;
   a. these can be on the sides or
   b. the bottom and one side.
4. Generally, a flowchart will flow from top to bottom. However, an upward flow can be shown as long as it does not exceed 3 symbols.
5. Connectors are used to connect breaks in the flowchart. Examples are:
   - From one page to another page.
   - From the bottom of the page to the top of the same page.
   - An upward flow of more than 3 symbols
6. Subroutines and Interrupt programs have their own and independent flowcharts.
7. All flow charts start with a Terminal or Predefined Process (for interrupt programs or subroutines) symbol.
8. All flowcharts end with a terminal or a contentious loop.

Flowcharting uses symbols that have been in use for a number of years to represent the type of operations and/or processes being performed. The standardized format provides a common method for people to visualize problems together in the same manner. The use of standardized symbols makes the flow charts easier to interpret, however, standardizing symbols is not as important as the sequence of activities that make up the process.

School of Computing and Information Technologies

CLASS NUMBER: # 10 SECTION: AE193

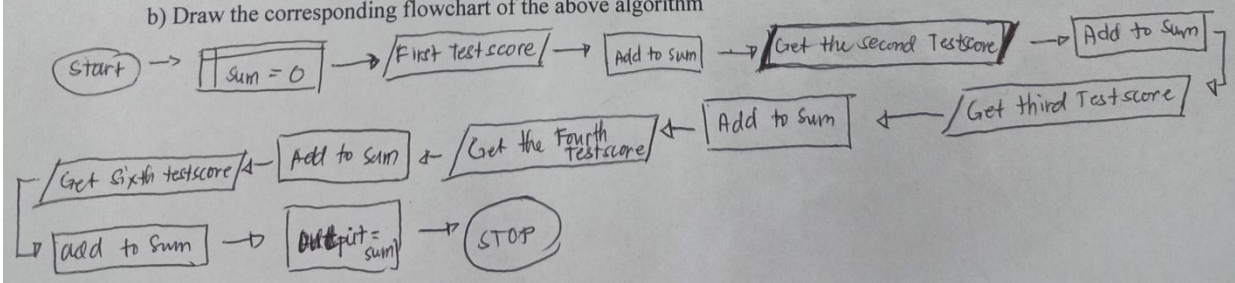
NAME: Jandayan, Jessa A DATE: Nov. 08, 2019

Exercise 1. Design **an algorithm** and **the corresponding flowchart** for adding the test scores as given below: 85, 65, 78, 52, 98, 83

a) Algorithm

1. Start
2. Sum = 0
3. Get the first testscore
4. Add first testscore to sum
5. Get the second testscore
6. Add the second testscore to Sum
7. Get the Third testscore
8. Add to Sum
9. Get the fourth testscore
10. Add the sum
11. Get the fifth testscore
12. add to sum
13. Get the sixth testscore
14. add to sum
15. Output = sum
16. STOP

b) Draw the corresponding flowchart of the above algorithm



School of Computing and Information Technologies

CLASS NUMBER: # 10 SECTION: AC 192

NAME: Jandayan, Jessa A. DATE: November 28, 2019

Flowcharting Exercises

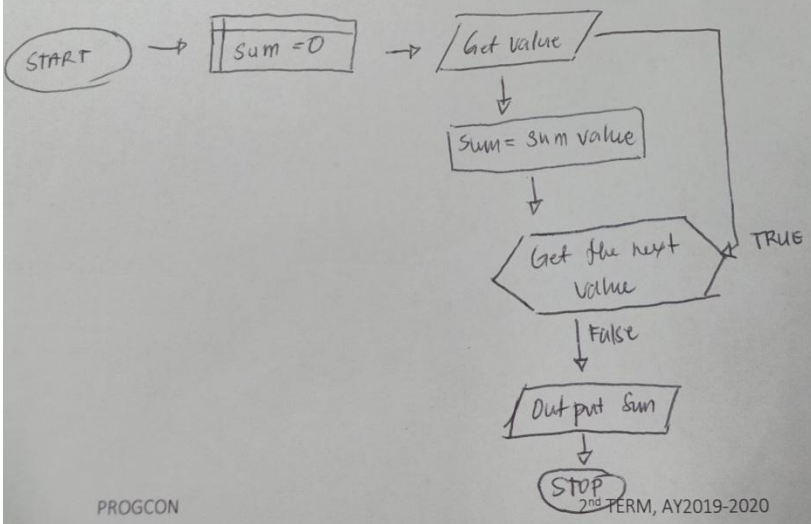
Exercise 2: The problem with the algorithm from Exercise 1 is that, some of the steps appear more than once, i.e. step 5 get second number, step 7, get third number, etc.

One could shorten the algorithm as follows:

1. Start
2. Sum = 0
3. Get a value
4. sum = sum + value
5. Go to step 3 to get next Value
6. Output the sum
7. Stop

This algorithm and its corresponding flowchart are a bit shorter than the first one. In this algorithm, step 3 to 5 will be repeated, where a number is obtained and added to sum. Similarly, the flowchart indicates a flow line being drawn back to the previous step indicating that the portion of the flowchart is being repeated.

Draw the corresponding flowchart of the above algorithm.



School of Computing and Information Technologies

CLASS NUMBER: #10 SECTION: BCIA12

NAME: Sandagan, Jessa A. DATE: Nov. 08 2019

Flowcharting Exercises

Exercise 3: From the exercise 2, one problem indicates that these steps will be repeated endlessly, resulting in an **endless** algorithm or flowchart.

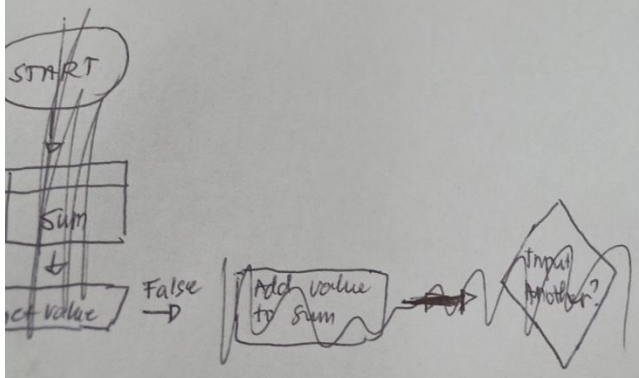
The algorithm needs to be improved to eliminate this problem. In order to solve this problem, we need to add a last value to the list of numbers given. This value should be unique so that, each time we get a value, we test the value to see if we have reached the last value.

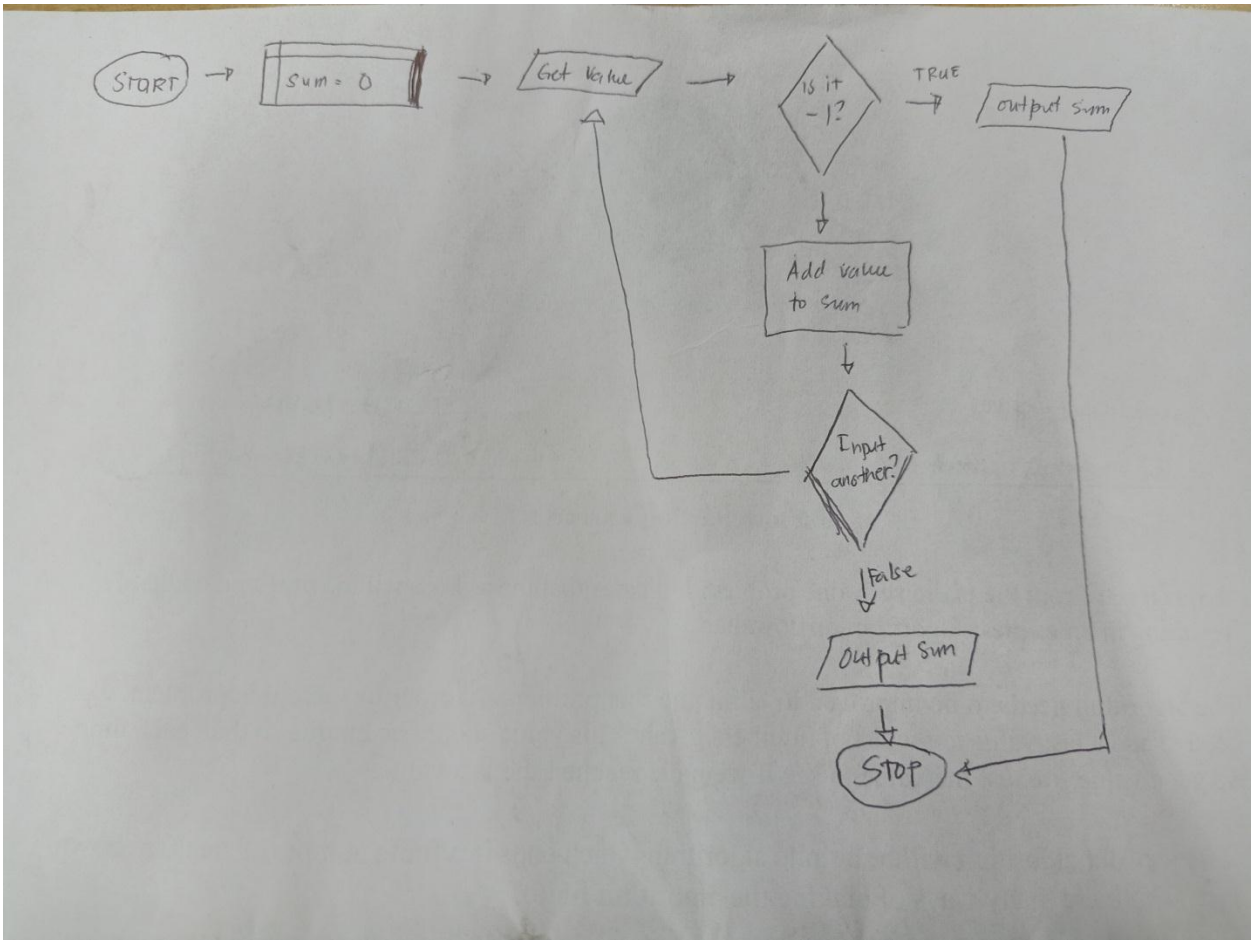
In this way our algorithm will be a finite algorithm which ends in a finite number of steps as shown below. There are many ways of making the algorithm finite.

The new list of numbers will be 10, 20, 300, 4000, 50000, 1, -1. The value –1 is a unique number since all other numbers are positive.

1. Start
2. Sum = 0
3. Get a value
4. If the value is equal to –1, go to step 7
5. Add to sum (sum = sum + value)
6. Go to step 3 to get next Value
7. Output the sum
8. Stop

Draw the corresponding flowchart of the above algorithm.




START
sum = 0
Get value
Is it -1?
TRUE
output sum
Add value to sum
Input another?
False
Output Sum
STOP