UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 3

Translating Algorithm to Program

Submitted by:
Catahan, Joshua A.

Instructor:
Engr. Maria Rizette H. Sayo

8, 9, 2025

I. Objectives

Introduction

Data structure is a systematic way of organizing and accessing data, and an algorithm is a step-by-step procedure for performing some tasks in a finite amount of time. These concepts are central to computing, but to be able to classify some data structures and algorithms as “good,” we must have precise ways of analyzing them.

This laboratory activity aims to implement the principles and techniques in:

- Writing a well-structured procedure in programming
- Writing algorithm that best suits to solve computing problems
- Writing an efficient Python program from translated algorithms

II. Methods

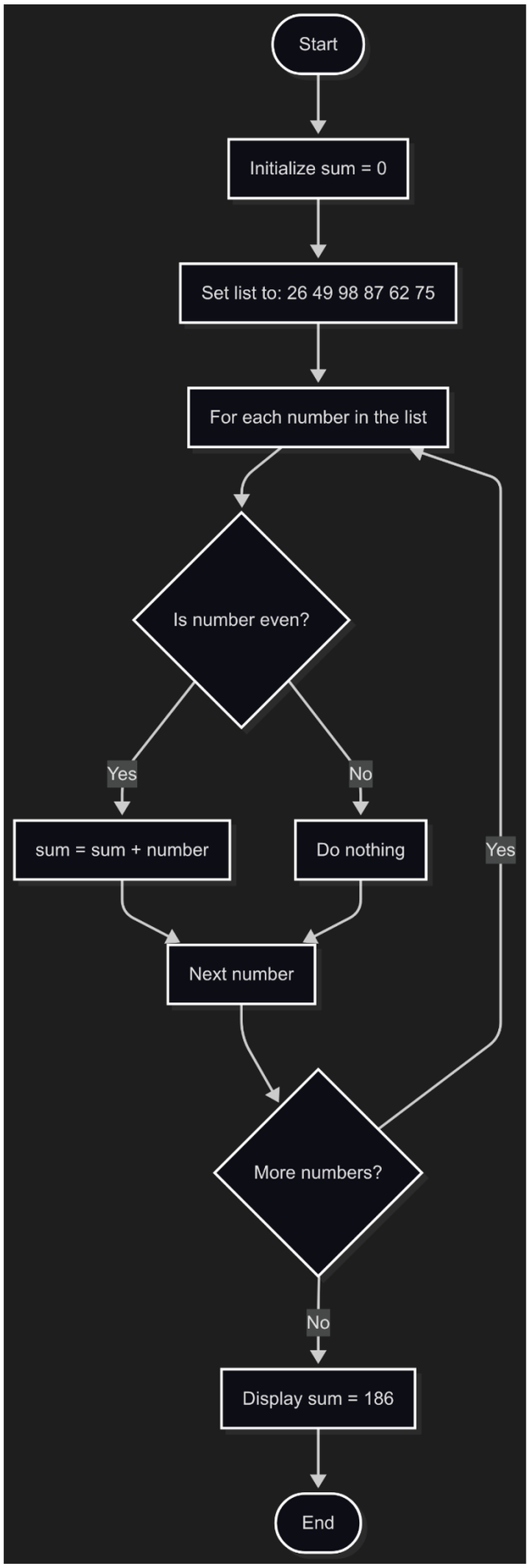
- Design an algorithm and the corresponding flowchart (Note: You may use LucidChart or any application) for adding the test scores as given below if the number is even: 26,49,98,87,62,75
- Translate the algorithm to a Python program (using Google Colab)
- Save your source codes to GitHub

III. Results

ALGORITHM

Step 1: Start
Step 2: Initialize sum = 0
Step 3: Set list = [26, 49, 98, 87, 62, 75]
Step 4: For each number in list
Step 4.1: Is number even?
    Yes: Add number to sum
    No: Do nothing
Step 5: Display sum (186)
Step 6: End

FLOWCHART


Start
Initialize sum = 0
Set list to: 26 49 98 87 62 75
For each number in the list
Is number even?
Yes
No
sum = sum + number
Do nothing
Next number
More numbers?
Yes
No
Display sum = 186
End

PYTHON PROGRAM

```
sum_even = 0

List = [26, 49, 98, 87, 62, 75]

for number in List:
    if number % 2 == 0:
        sum_even += number
    else:
        pass

print("Sum of even numbers:" , sum_even)
```

```
Sum of even numbers: 186
```

COMMENTARIES FOR ALGORITIHM, FLOWCHART AND PYTHON PROGRAM:

The algorithm starts by setting the sum to zero and listing the given test scores. Each score is checked one by one to see if it is even. If it is, the score is added to the sum; if not, it is skipped. This process continues until all scores have been checked, and the final total of the even numbers, 186, is displayed. The flowchart shows the same process in a visual format, using ovals for the start and end, rectangles for actions, and a diamond for the decision of whether a number is even. The Python program is the actual implementation of the algorithm, following the same logical sequence in code form: it initializes the sum, loops through each score, checks if it is even, adds it to the sum if true, and finally prints the result.

IV. Conclusion

This activity successfully demonstrated how a single problem can be represented and solved using an algorithm, a flowchart, and a Python program. The algorithm provided a clear and structured plan, the flowchart illustrated the flow of decisions and actions, and the Python program brought the logic to life by producing the correct output. This approach reinforced the importance of planning and visualization in programming, ensuring that the final implementation is both accurate and efficient.

References

[1] Co Arthur O.. “University of Caloocan City Computer Engineering Department Honor Code,” UCC-CpE Departmental Policies, 2020.