



UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 3

Translating Algorithm to Program

Submitted by:

Asugas, Kenneth R.

Instructor:

Engr. Maria Rizette H. Sayo

08, 02, 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

Present the visualized procedures done. Also present the results with corresponding data visualizations such as graphs, charts, tables, or image . Please provide insights, commentaries, or explanations regarding the data. If an explanation requires the support of literature such as academic journals, books, magazines, reports, or web articles please cite and reference them using the IEEE format.

Please take note of the styles on the style ribbon as these would serve as the style format of this laboratory report. The body style is Times New Roman size 12, line spacing: 1.5. Body text should be in Justified alignment, while captions should be center-aligned. Images should be readable and include captions. Please refer to the sample below:

Algorithm:

1. Start
2. Set scores = [26, 49, 98, 87, 62, 75]
3. Set sum = 0
4. For each number in scores:
 - Is number even?
 - Yes → Add to sum
 - No → skip
5. Repeat until all numbers are checked
6. Print/Display sum
7. End

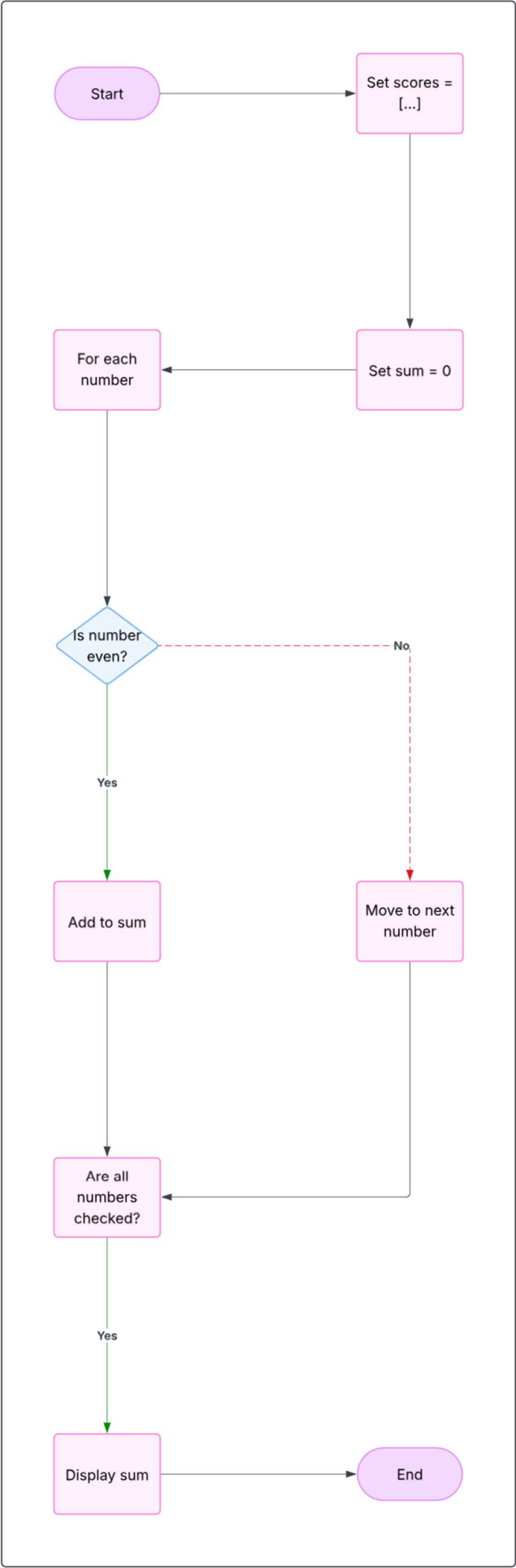
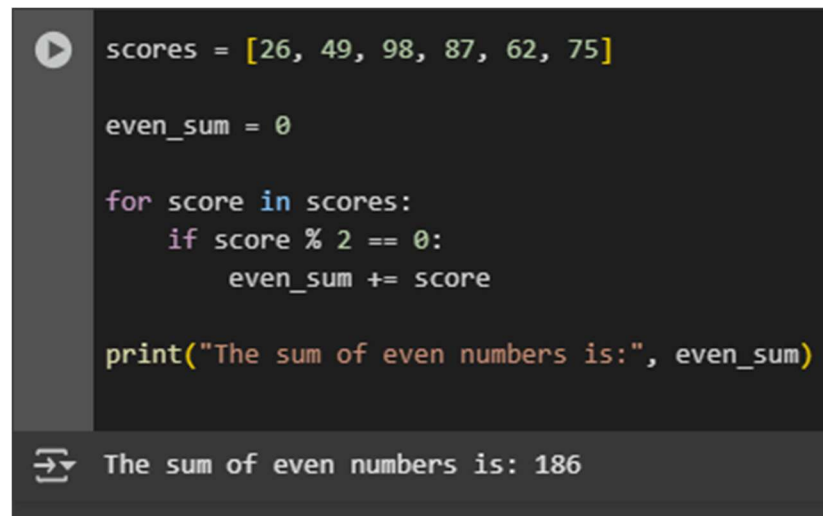


Figure 1: Flowchart

Figure 1 shows the flowchart used to determine and add only the even numbers from the list [26, 49, 98, 87, 62, 75]. It checks each number, adds it to the total if it's even, and proceeds until all numbers are processed.



```
scores = [26, 49, 98, 87, 62, 75]

even_sum = 0

for score in scores:
    if score % 2 == 0:
        even_sum += score

print("The sum of even numbers is:", even_sum)
```

The sum of even numbers is: 186

Figure 2: Screenshot of Program

Figure 2 shows the corresponding Python program was done in Google Colab. It successfully identified the even numbers 26, 98, and 62. Resulting in a total sum of **186**.

IV. Conclusion

This activity helped reinforce the connection between algorithm design and programming implementation. By filtering even numbers and translating this logic into a Python program, I learned to think step-by-step and ensure that my code reflects the intent of my algorithm.

References

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