UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 3

Translating Algorithm to Program

Submitted by:
Mamanao, Kurt Marwin C.

Instructor:
Engr. Maria Rizette H. Sayo

Month, DD, YYYY

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:

FLOWCHART

1. **Start/End (Oval)**
    - Indicates the beginning and termination of the algorithm.
2. **Process (Rectangle)**
    - sum_of_odds = 0: Initializes the variable to store the sum of odd numbers.
    - sum_of_odds += num: Adds the current odd number to the running total.

Decision (Diamond)

- Evaluates the condition num % 2 != 0 to check if a number is odd.

3. **Input/Output (Parallelogram)**

- Displays the final result using Print sum_of_odds.

Would you like help turning this into an actual flowchart diagram or expanding the algorithm further?

Algorithm Workflow

- The process kicks off by initializing a variable called sum_of_evens to zero.
- It then enters a loop that examines each number in the input list one by one.

For Every Number:

Even?

- If the number is even, it's added to sum_of_even, and the loop continues to the next item.

Odd?

- If the number is odd, it's ignored, and the loop moves forward without updating the sum.

Once all numbers have been processed, the final total of even numbers is displayed, and the algorithm concludes.

Sample Run

Input List: [26, 49, 98, 97, 62, 75]

Even Numbers: 26, 98, 62:

Sum Calculated: 26 + 98 + 62 = 186

ALGORITHM

1. **Start**

Begin the program.

2. **Initialize the list and set sum = 0**

   Example:

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

sum = 0

3. **Check if there are more numbers in the list**

   If yes, go to step 4.

   If no, go to step 6.

4. **Check if the current number is even**

   If yes, go to step 5.

   If no, go back to step 3 for the next number.

5. **Add the even number to sum**

   sum = sum + number

   Then go back to step 3.

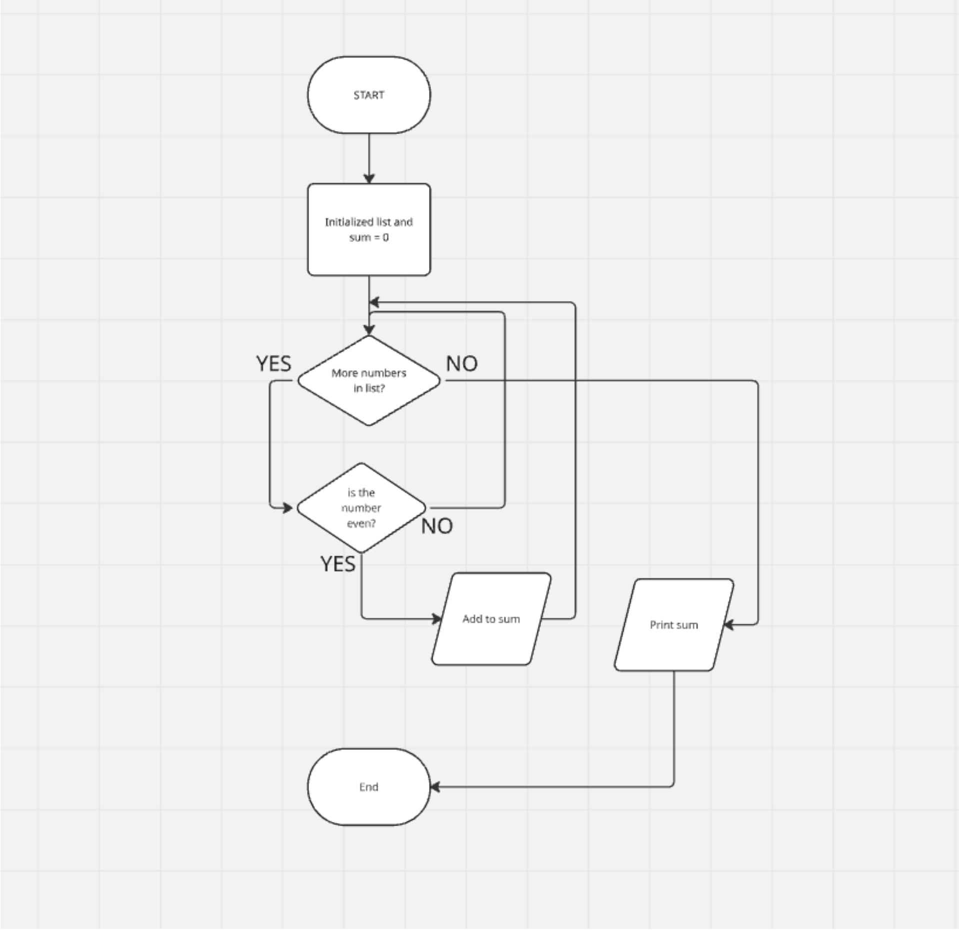
6. **Print the value of sum**

   Display the total sum of even numbers.

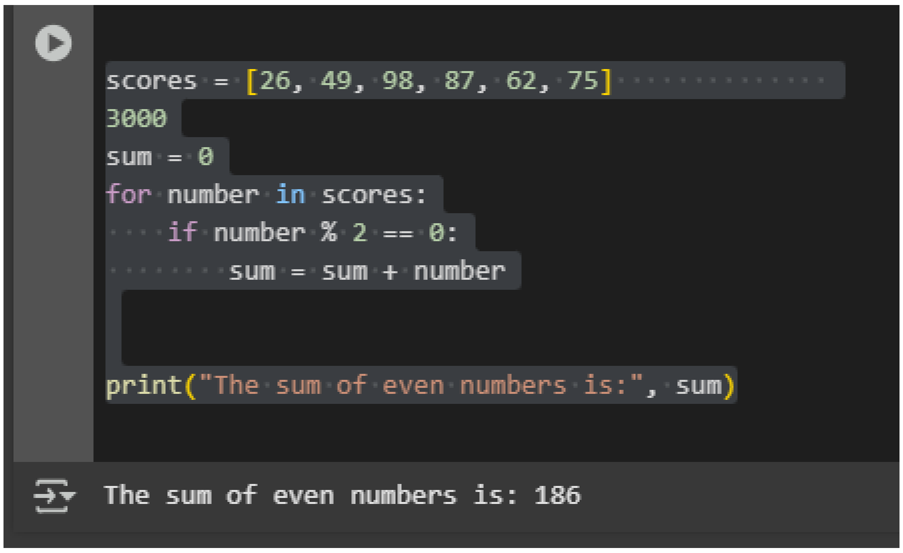
7. **End**

   The program stops.

FLOWCHART



SOURCE CODE



```
scores = [26, 49, 98, 87, 62, 75]
3000
sum = 0
for number in scores:
    if number % 2 == 0:
        sum = sum + number


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

```
The sum of even numbers is: 186
```

IV. Conclusion

I’ve been learning Python for a while now, but I still need a guide to do this kind of program, the usage of loops and conditions. My goal was to go through a list of numbers and add only the even ones. First, I made a list called `scores` and set `sum = 0`. Then I used a `for` loop to check each number. If the number was even (I used `number % 2 == 0` to check), I added it to the sum. If it was odd, It will skipped it.   The code was simple and yet I had the hard time constructing it , but it worked well. It showed me that I can use guides until I can do it on my own. When I saw the result, I felt good because I knew the program was doing what I wanted.

References

- P. S. Author, "Flow Chart To Print The Sum of First 20 Even Numbers," Scribd, May 16, 2025. [Online]. Available: https://www.scribd.com/doc/93495952/JT1113020. [Accessed: Aug. 2, 2025].