Data Structure and Algorithm

Laboratory Activity No. 3

Translating Algorithm to Program

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Monoy, Justin Rhey A. | Engr. Maria Rizette H. Sayo |

August 02, 2025

# 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

# 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

# Results

Algorithm

Step 1: Start

Step 2: Initialize sum = 0

Step 3: Create a list of scores: [26, 49, 98, 87, 62, 75]

Step 4: Loop through each score:

Step 5: If the score is even (score % 2 == 0)

Step 6: Add it to sum

Step 7: Output the sum

Step 8: End

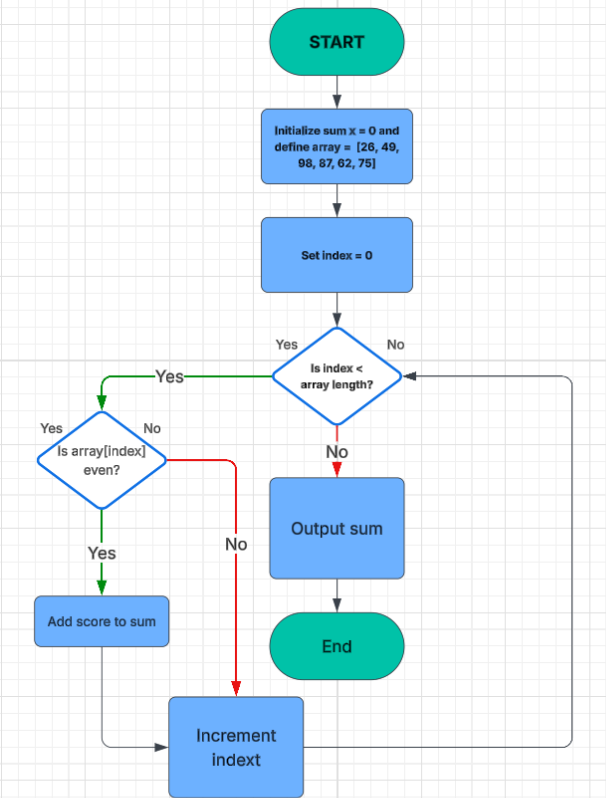


Figure 1. Flowchart

Lists and arrays are the data structures that are used to store multiple items. They both support the indexing of elements to access them, slicing, and iterating over the elements. This flowchart illustrates the process of summing only the even numbers from a list of test scores. It begins by initializing a variable sum to 0 and defining an array of scores: [26, 49, 98, 87, 62, 75]. The index is then set to 0 to start checking from the first score in the array. A decision is made to see whether the current index is still within the bounds of the array (in this case, the array length is 6). If it is, the program checks whether the current score at that index is an even number. If the number is even, it is added to the running total stored in sum. If the number is not even, it is skipped. After that, the index is incremented by 1 to move to the next score in the array, and the process repeats. This loop continues until the index reaches the end of the array. Once all the elements have been checked, the total of the even scores is displayed as the output, and the program ends.

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 2. Source code

This Python code calculates the sum of all even numbers in the list scores = [26, 49, 98, 87, 62, 75]. It starts by initializing the variable sum\_even to 0. Then, it uses a for loop to go through each value in the scores list. For each score, it checks whether the number is even using the condition score % 2 == 0. If the condition is true, the even number is added to the total using sum\_even += score. After the loop has gone through all the scores, the final sum of the even numbers is printed. In this case, the even numbers are 26, 98, and 62, and the sum is 186.

# Conclusion

In this lab, I learned how to create and use algorithms and flowcharts to solve a problem step by step. I practiced checking if a number is even and adding it to the total, then turning that logic into a Python program. I used tools like LucidChart for the flowchart and Google Colab for coding. This activity showed us how important it is to plan before coding, follow clear steps, and write efficient programs. It also improved our skills in thinking logically and working with real coding tools.

**References**

[1] “W3Schools.com.” <https://www.w3schools.com/python/ref_list_index.asp>

[2] GeeksforGeeks, “Difference between List and Array in Python,” *GeeksforGeeks*, Jul. 15, 2025. <https://www.geeksforgeeks.org/python/difference-between-list-and-array-in-python/>