Data Structure and Algorithm

Laboratory Activity No. 5

Implementation of Arrays

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Luminario, Venice Lou Gabrielle M. | Engr. Maria Rizette H. Sayo |

Aug, 16, 2025

# Objectives

Introduction

Array, in general, refers to an orderly arrangement of data elements. Array is a type of data structure that stores data elements in adjacent locations. Array is considered as linear data structure that stores elements of same data types. Hence, it is also called as a linear homogenous data structure.

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

* Writing algorithms using Array data structure
* Writing a python program that can implement Array data structure

# Methods

* Write a Python program to create an array of 10 integers and display the array items. Access individual elements through indexes and compute for the sum.
* Write a Python program to append a new item to the end of the array. Original array: numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
* Write a Python program to insert a new item before the second element in an existing array. Original array: numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
* Write a Python program to reverse the order of the items in the array. Original array: numbers = [5, 4, 3, 2, 1]
* Write a Python program to get the length of the array. Original array: numbers = [5, 4, 3, 2, 1]

# Results

Linear data structures are used in these arrays to store elements in memory in a sequential fashion. Key Python array actions, such as creation, insertion, appending, reversing, and length calculation, were implemented in this lab. These exercises highlighted the fixed-size nature and insertion costs of arrays while demonstrating their effective indexing access. The practical exercise demonstrated why arrays are still essential for programming's quick element retrieval and memory-efficient data storage.

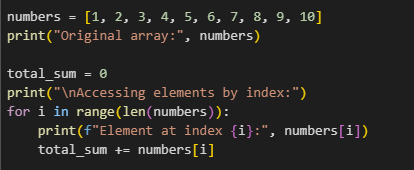


Figure 1 Screenshot of program

In order to demonstrate the basic operations of array creation, element access, and mathematical computation that form the basis of array manipulation in programming, this will create a list of ten integers, compute the sum of all the elements in the list, print the entire list, and use each element's index to display the index and associated value.

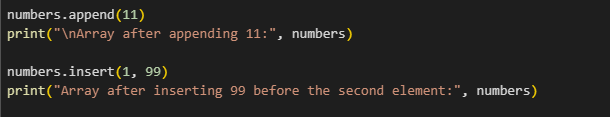


Figure 2 Screenshot of program

So this code will add the number 11 to the end of the number list using append() so it will print the updated list, after that it will insert the number 99 into the index 1, and will be placed before the second element in the list , and it will prints again the list to show it again or to show the updated list.

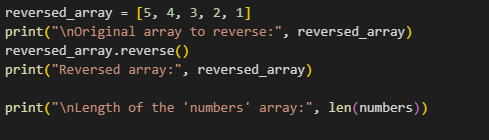


Figure 3 Screenshot of program

So this function is called the reversed\_array, it prints the original values, then it will reverse it in order using the .reverse() method and prints it. And finally it will prints the number of the elements in the numbers using the len() function.

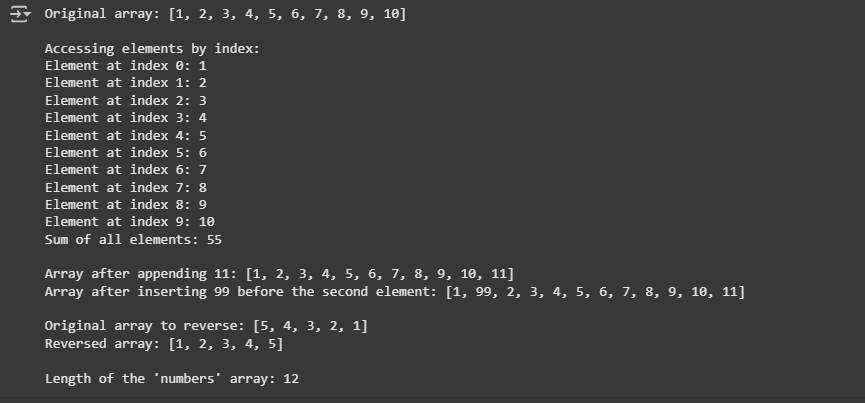


Figure 4 Result

# Conclusion

In my conclusion this lab will demonstrate the fundamental of array operations in Python, such as creation, element access, summation, insertion, reverse, and length verification. These activities demonstrated how arrays provide efficient data storage, quick access, and reliable performance. Overall, the hands-on experience highlighted the significance of arrays as a basis for more advanced data structures in programming. This is a new knowledge to me, now that know it can help me to absorb to and make a little complex code.

**References**

Singh, U. V. (2025, July 2025). *Understanding arrays: A foundational data structure across languages*. Medium.

Wikipedia contributors. (2025, August (system’s today)). *Array (data structure)*. In *Wikipedia*. Retrieved from Wikipedia.