



Data Structure and Algorithm

Laboratory Activity No. 5

---

# Implementation of Arrays

---

*Submitted by:*

Manigbas, Jeus Miguel T. <Leader>

Barredo, Alwin P.

Bela, Lorenzo Miguel D.

Callorina, Robert Victor A.

Guzon, Kean Louiz I.

*Instructor:*

Engr. Maria Rizette H. Sayo

November, 17, 2023

---

# I. 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.

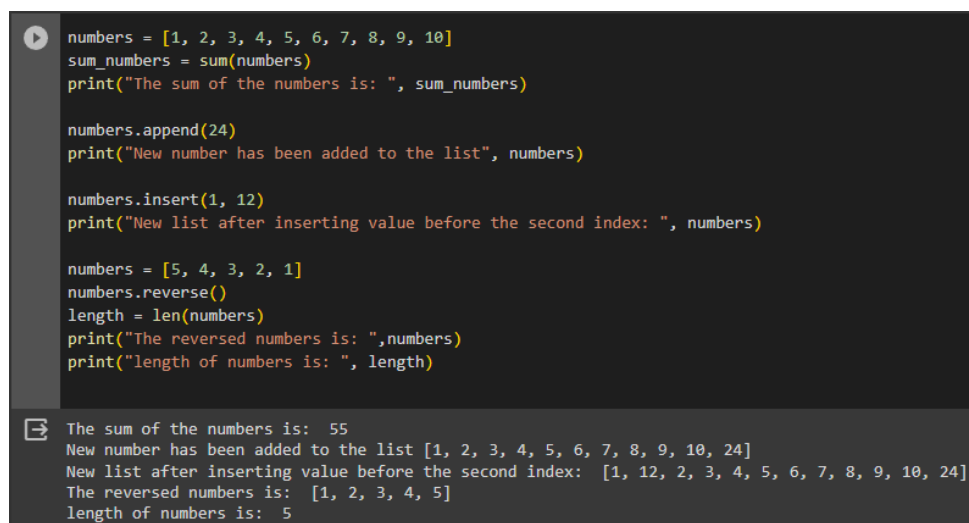
This laboratory activity aims to implement the principles and techniques in:

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

# II. 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.
- 1. 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]
- 2. 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]

# III. Results



```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
sum_numbers = sum(numbers)
print("The sum of the numbers is: ", sum_numbers)

numbers.append(24)
print("New number has been added to the list", numbers)

numbers.insert(1, 12)
print("New list after inserting value before the second index: ", numbers)

numbers = [5, 4, 3, 2, 1]
numbers.reverse()
length = len(numbers)
print("The reversed numbers is: ",numbers)
print("length of numbers is: ", length)
```

The sum of the numbers is: 55  
New number has been added to the list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 24]  
New list after inserting value before the second index: [1, 12, 2, 3, 4, 5, 6, 7, 8, 9, 10, 24]  
The reversed numbers is: [1, 2, 3, 4, 5]  
length of numbers is: 5

Figure 1: The Code and its output

Using Python functions, namely sum, append, insert, reverse, and len.

**Sum:** This function is used to calculate the sum of all the numbers in the list. It takes an iterable as its argument and returns the sum of the elements in the iterable. The syntax is `sum(iterable, start)`, where `iterable` is the sequence to sum and `start` is an optional value that is added to the return value

**Append:** This method is used to add a single element to the end of the list. It takes an element as its argument and adds it to the end of the list. The syntax is `list.append(element)`

**Insert:** This method is used to insert a specified value at a specified position in the list. It takes two parameters: the position at which to insert the value and the value to be inserted. The syntax is `list.insert(pos, elmnt)`.

**Reverse:** This method is used to reverse the order of the elements in the list. It doesn't take any arguments and updates the existing list. The syntax is `list.reverse()`.

**Len:** This function is a built-in Python function that returns the length of an object. It can be used to find the length of a string, array, list, tuple, dictionary, or any other data type syntax of the `len()` function is `len(object)`, where `object` is the object whose length needs to be determined

## IV. Conclusion

The provided code makes use of a number of Python functions and methods to efficiently modify a list of numerical values. The `sum()` function specifically computes the total amount of all the numbers in the list, providing a simplified method to computation on data collections. Furthermore, the `append()` method extends the list by appending a new number to the end, simplifying the process of list expansion without requiring complex indexing. Furthermore, the `insert()` method inserts a new number precisely at an established position within the list, granting full control over the list's contents at specific locations. Furthermore, the `reverse()` method quickly changes the sequence of elements within the list, providing a simple way to reverse their order, which can be useful for a variety of data manipulation tasks. Finally, the `len()` function efficiently calculates the length of the list, providing vital data about the size of the modified list. Overall, this code adequately demonstrates the functionality of these key Python functions and methods, emphasizing their importance in efficiently manipulating and managing lists in Python programming for a variety of data-oriented tasks.

## References

- [1] “Sum function in Python,” *GeeksforGeeks*, Nov. 24, 2017. <https://www.geeksforgeeks.org/sum-function-python/> (accessed Nov. 20, 2023).
- [2] Real Python, “Python’s .append(): Add Items to Your Lists in Place,” *Realpython.com*, Jan. 04, 2021. <https://realpython.com/python-append/> (accessed Nov. 20, 2023).
- [3] “Python List insert(),” *Programiz.com*. <https://www.programiz.com/python-programming/methods/list/insert> (accessed Nov. 20, 2023).
- [4] “Python List reverse,” *GeeksforGeeks*, Jan. 12, 2018. <https://www.geeksforgeeks.org/python-list-reverse/> (accessed Nov. 20, 2023).
- [5] Real Python, “Using the len() Function in Python,” *Realpython.com*, Oct. 20, 2021. <https://realpython.com/len-python-function/> (accessed Nov. 20, 2023).