UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 5

Implementation of Arrays

Submitted by:
Regondola, Jezreel P.

Instructor:
Engr. Maria Rizette H. Sayo

August 16, 2025

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:

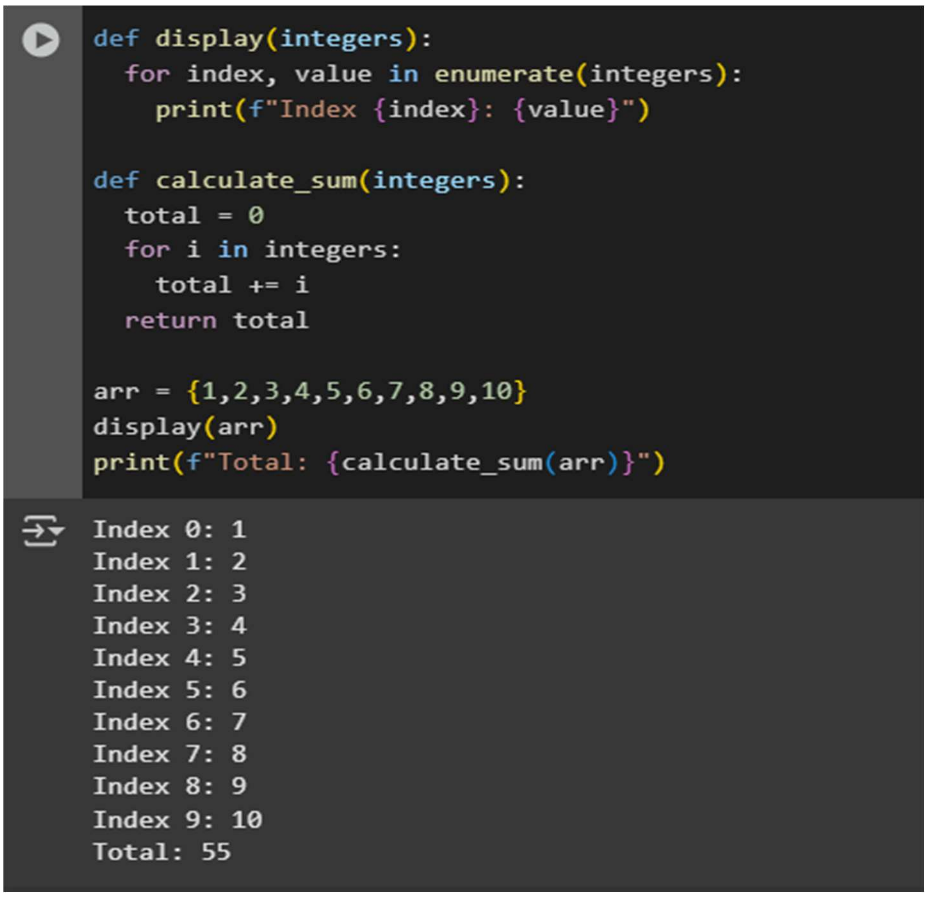
- Writing algorithms using Array data structure
- Writing a python program that can implement Array data structure

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.
- 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]

III. Results

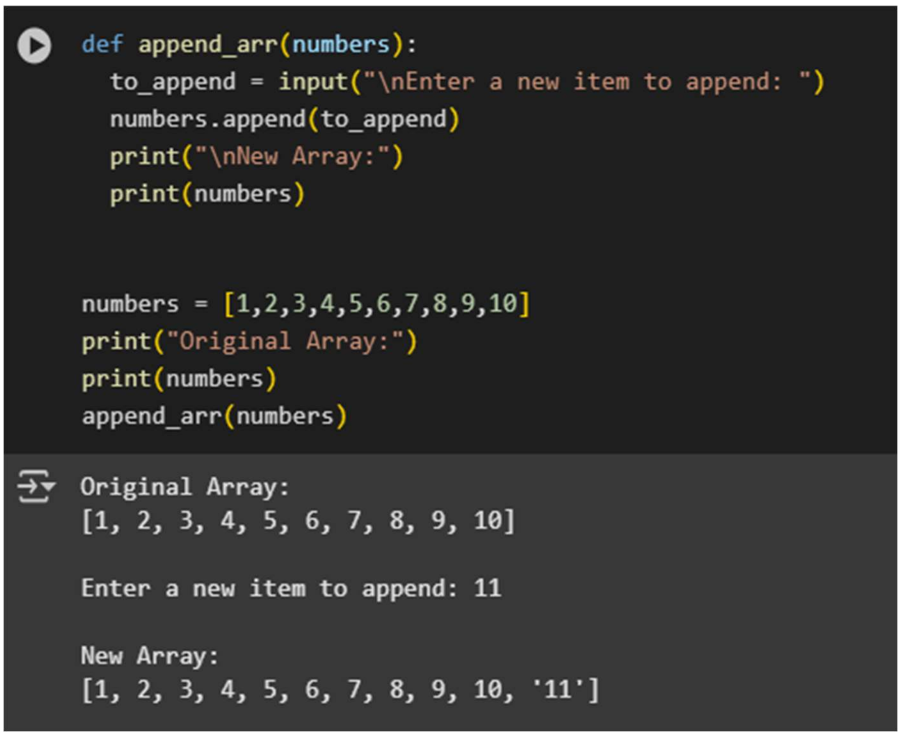


```
def display(integers):
  for index, value in enumerate(integers):
    print(f"Index {index}: {value}")

def calculate_sum(integers):
  total = 0
  for i in integers:
    total += i
  return total

arr = {1,2,3,4,5,6,7,8,9,10}
display(arr)
print(f"Total: {calculate_sum(arr)}")
```

```
Index 0: 1
Index 1: 2
Index 2: 3
Index 3: 4
Index 4: 5
Index 5: 6
Index 6: 7
Index 7: 8
Index 8: 9
Index 9: 10
Total: 55
```

Figure 1 Screenshot of program

This program prints the numbers from 1 to 10 with their indexes and then calculates their total sum. The display function shows each index and value, while calculate_sum adds up all the numbers.

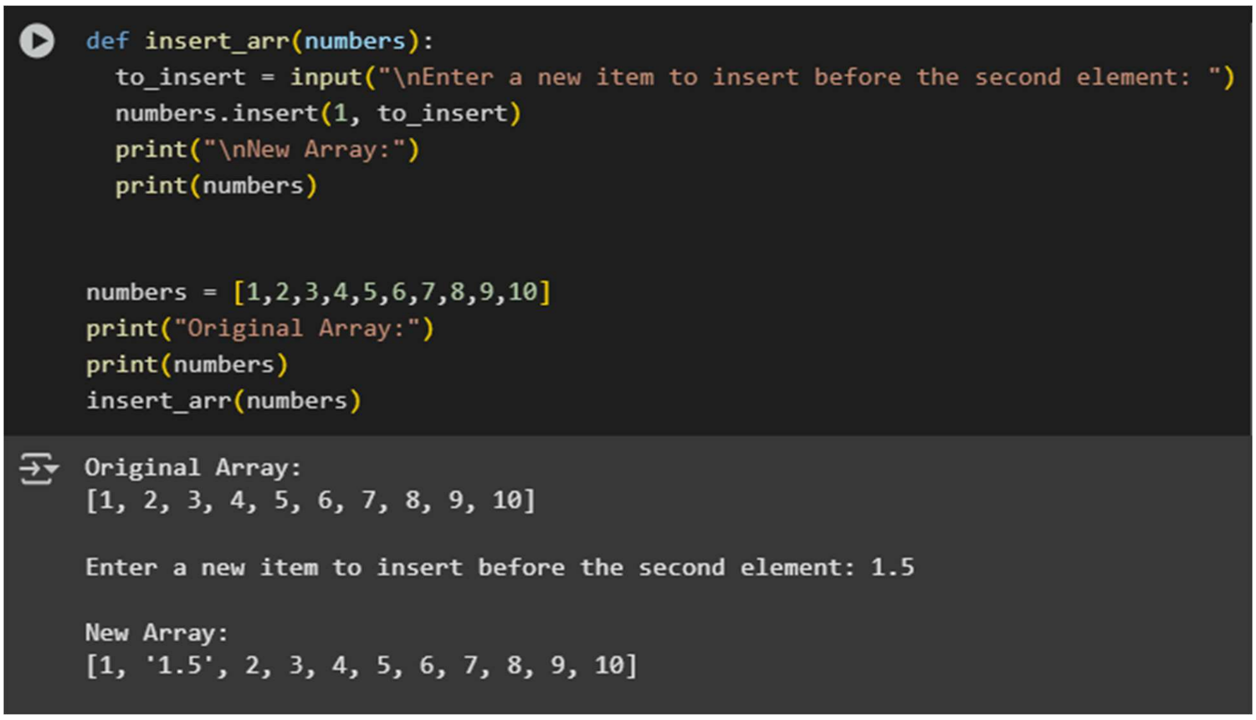


```
def append_arr(numbers):
  to_append = input("\nEnter a new item to append: ")
  numbers.append(to_append)
  print("\nNew Array:")
  print(numbers)


numbers = [1,2,3,4,5,6,7,8,9,10]
print("Original Array:")
print(numbers)
append_arr(numbers)
```

```
Original Array:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Enter a new item to append: 11

New Array:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, '11']
```

Figure 2 Screenshot of program

This program shows how to add a new item to a list. It first prints the original array, then asks the user for a value to append, and finally displays the updated array with the new item added at the end.

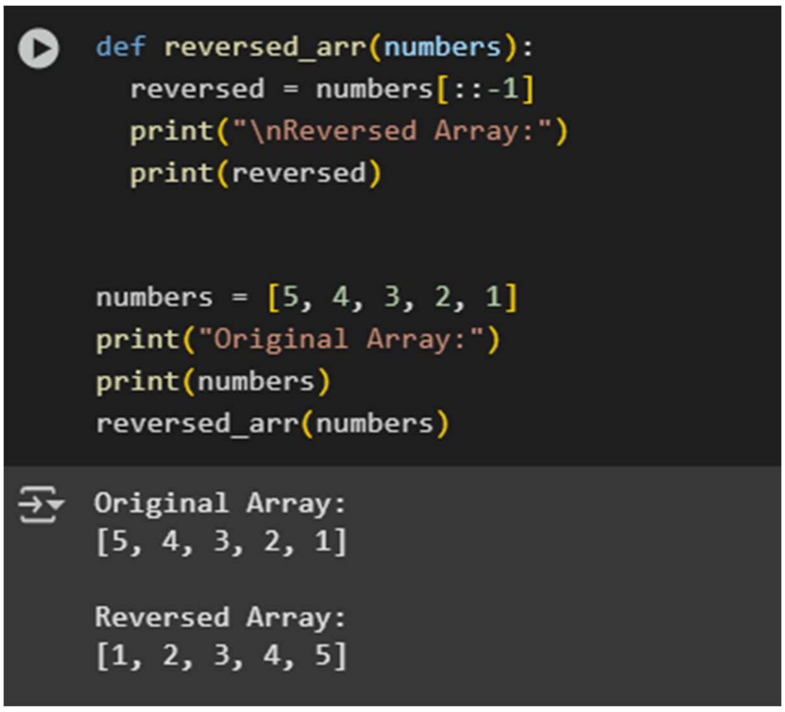


```
def insert_arr(numbers):
  to_insert = input("\nEnter a new item to insert before the second element: ")
  numbers.insert(1, to_insert)
  print("\nNew Array:")
  print(numbers)


numbers = [1,2,3,4,5,6,7,8,9,10]
print("Original Array:")
print(numbers)
insert_arr(numbers)
```

```
Original Array:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Enter a new item to insert before the second element: 1.5

New Array:
[1, '1.5', 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Figure 3 Screenshot of program

This program inserts a new item before the second element of a list. It asks the user for input, adds it at index 1, and then prints the updated list.

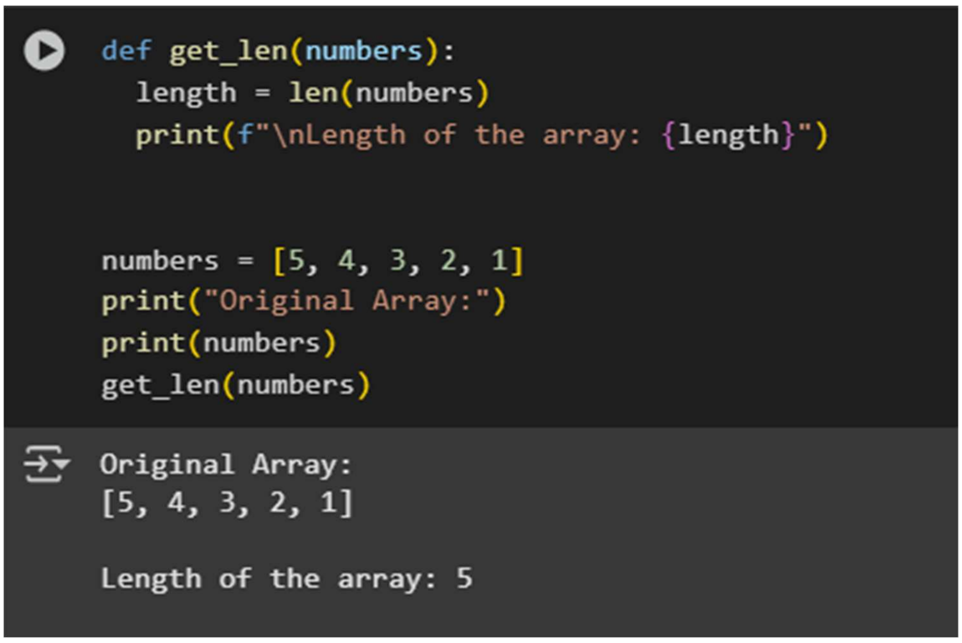


```
def reversed_arr(numbers):
  reversed = numbers[::-1]
  print("\nReversed Array:")
  print(reversed)


numbers = [5, 4, 3, 2, 1]
print("Original Array:")
print(numbers)
reversed_arr(numbers)
```

```
Original Array:
[5, 4, 3, 2, 1]

Reversed Array:
[1, 2, 3, 4, 5]
```

Figure 4 Screenshot of program

This program reverses the order of elements in a list using slicing ([::-1]) and then prints the reversed list.



```
def get_len(numbers):
  length = len(numbers)
  print(f"\nLength of the array: {length}")


numbers = [5, 4, 3, 2, 1]
print("Original Array:")
print(numbers)
get_len(numbers)
```

```
Original Array:
[5, 4, 3, 2, 1]

Length of the array: 5
```

Figure 5 Screenshot of program

This program calculates the number of elements in a list using len() and prints the length.

IV. Conclusion

This activity showed how to use arrays in Python by writing simple programs. It demonstrated basic operations like inserting, reversing, and finding the length of an array, proving how arrays help store and manage data easily.

References

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