**Department of Electrical Engineering and   
Computer Science**

**Faculty Member:** Dr. Mohsin Kamal **Dated:** 21/09/2023

**Semester:** 7th **Section:** BEE 12C

**CS-477 Computer Vision**

Lab 2: Data Structures and NumPy

**Group Members**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **PLO4 - CLO4** | | **PLO5 -CLO5** | **PLO8 -CLO6** | **PLO9 -CLO7** |
| **Name** | **Reg. No** | **Viva / Quiz / Lab Performance** | **Analysis of Data in Lab Report** | **Modern Tool Usage** | **Ethics and Safety** | **Individual and Teamwork** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** |
| Muhammad Ahmed Mohsin | 333060 |  |  |  |  |  |
| Muhammad Umer | 345834 |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Table of Contents**

[2 Data Structures and NumPy 3](#_Toc146294020)

[2.1 Introduction 3](#_Toc146294021)

[2.2 Objectives 3](#_Toc146294022)

[2.3 Software 3](#_Toc146294023)

[3 Lab Tasks 4](#_Toc146294024)

[3.1 Task 1 4](#_Toc146294025)

[3.2 Task 2 4](#_Toc146294026)

[3.3 Task 3 5](#_Toc146294027)

[3.4 Task 4 5](#_Toc146294028)

[3.5 Task 5 6](#_Toc146294029)

[3.6 Task 6 7](#_Toc146294030)

[4 Conclusion 8](#_Toc146294031)

# Data Structures and NumPy

## Introduction

This lab report will introduce you to the basic data structures and operations in Python, including lists and dictionaries. You will learn how to create, alter, and loop through lists, as well as how to use slicing to access a range of items in a list. You will also learn how to use various list methods, such as append, insert, extend, remove, and pop. Additionally, you will learn how to implement 2-D lists and dictionaries. Finally, you will get a brief introduction to NumPy, a popular Python library for scientific computing.

## Objectives

The following are the main objectives of this lab:

* Implement data structures such as lists and dictionaries in python
* Create, alter, and loop through lists
* Use slicing to access range of items in a list
* Utilize various list methods such as append, insert, extend, remove, pop etc.
* Implement 2-D lists.
* Create and implement a dictionary.
* Introduction to NumPy

## Software



Data structures are an important part of python. The 4 main data structures are lists, tuples, sets and dictionaries. Lists are most used so they will be a major part of the lab tasks. Dictionaries are also used at times. Tuples and sets are very similar to lists and are not very commonly used in robotics.

A summary of the list functions in python is provided below. (For more details, check the slides for this lab).

**append(I)** append item I to the end of the list

**insert(i, I)** insert item I at i position of the list

**extend(L)** extend/concatenate a second list L

**remove(I)** remove a specified item I from a list

**pop(i)** remove item at specific index i in the list

**count(I)** return total number of a specific item I from a list

**index(I)** return index of first occurrence of a specific item I

**reverse** reverse the items of the list

# Lab Tasks

## Task 1

Create a simple list containing the characters of the word: MANIPULATOR. Loop through the list and display each character on a new line.

### Task 1 Code Starts Here ###

name\_list = ["M", "A", "N", "I", "P", "U", "L", "A", "T", "O", "R"]

for character in name\_list:

    print(*f*"{character}")

### Task 1 Code Ends Here ###

### Task 1 Screenshot Starts Here ###



### Task 1 Screenshot Ends Here ###

## Task 2

Write a program which first prompts the user for an integer which will be the size of a list. Then, the program must repeatedly prompt the user to input the items of the list. Each item is to be added with the append function. The inputs continue until the number of items reach the size of the list. The final list is then printed. The syntax for making an empty list and appending function are given below:

my\_list = [ ]

my\_list.append(item)

### Task 2 Code Starts Here ###

size = *int*(input("Enter the size of the list: "))

number\_list = []

count = 0

while count < size:

    item = input("Enter an input: ")

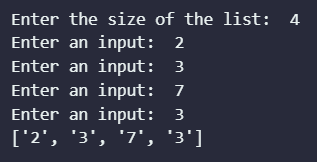
    number\_list.append(item)

    count += 1

print(number\_list)

### Task 2 Code Ends Here ###

### Task 2 Screenshot Starts Here ###



### Task 2 Screenshot Ends Here ###

## Task 3

Write a function that takes in a list input and returns true if the items of the list make a palindrome. A palindrome is a word/number that is written the same way forward and backward. Examples of palindromes include “radar”, “level”, “5445”, “8395938”, “racecar”. To use a list in a function, use the following syntax:

def my\_function(my\_list):

statement1

statement2

### Task 3 Code Starts Here ###

*def* palindrome\_check(*my\_list*):

    if my\_list == my\_list[::-1]:

        return True

    else:

        return False

test\_word = input("Enter a word: ")

if palindrome\_check(*list*(test\_word)):

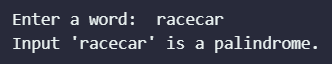
    print(*f*"Input '{test\_word}' is a palindrome.")

else:

    print(*f*"Input '{test\_word}' is not a palindrome.")

### Task 3 Code Ends Here ###

### Task 3 Screenshot Starts Here ###



### Task 3 Screenshot Ends Here ###

## Task 4

Write a program that repeatedly prompts the user for input. The user will keep entering numbers which are added to a list. Each time a number is added to the list, it must be placed in such a way that the list items are always in ascending order. Each time a number is input, the list is to be printed showing the newly added number. This continues until the word “done” is input at which point the prompts will stop. The final list is then displayed. Do NOT use any inbuilt sorting function for this task.

### Task 4 Code Starts Here ###

number\_list = []

while True:

    number = input("Enter an input: ")

    if number == "done":

        break

    else:

        number = *int*(number)

        for index, item in enumerate(number\_list):

            if number < item:

                number\_list.insert(index, number)

                break

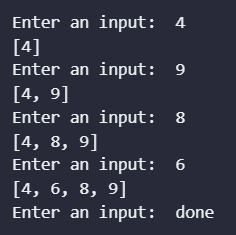
        else:

            number\_list.append(number)

        print(number\_list)

### Task 4 Code Ends Here ###

### Task 4 Screenshot Starts Here ###



### Task 4 Screenshot Ends Here ###

## Task 5

In this task, you will implement the Selection Sort algorithm using lists. A selection sort searches a list looking for the smallest element. Then, the smallest element is swapped with the first element of the list. The process is repeated for the sub-list beginning with the second element of the list. Each pass of the list results in one element being placed in its proper location. When the sub-list being processed contains one element, the list is sorted. Create a function which takes a list as input and then implements the selection sort on it. You need to print the list each time a swap is made.

### Task 5 Code Starts Here ###

my\_list = [5, 11, 8, 6, 7, 2]

print(*f*"Original list: {my\_list}\n")

for i in range(len(my\_list)):

    smallest\_index = i

    for j in range(i+1, len(my\_list)):

        if my\_list[j] < my\_list[smallest\_index]:

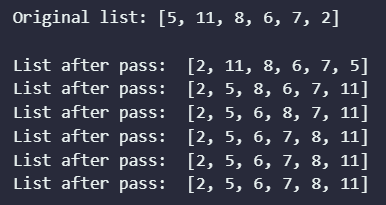
            smallest\_index = j

    my\_list[smallest\_index], my\_list[i] = my\_list[i], my\_list[smallest\_index]

    print("List after pass: ", my\_list)

### Task 5 Code Ends Here ###

### Task 5 Screenshot Starts Here ###



### Task 5 Screenshot Ends Here ###

## Task 6

Use the np.array function to define two matrices of size 3x3. Place numerical elements of your choice in the matrices. Write code to perform the following:

* Print the arrays
* Compute the sum of the matrices
* Compute the difference of the matrices
* Compute the element-wise product of the matrices
* Compute the element-wise division of the matrices
* Compute the matrix multiplication of the matrices

### Task 6 Code Starts Here ###

import numpy as np

arr\_1 = np.random.randint(1, 10, (3, 3))

arr\_2 = np.random.randint(1, 10, (3, 3))

print("Sum of Matrices: \n", arr\_1 + arr\_2)

print("\nDifference of Matrices: \n", arr\_1 - arr\_2)

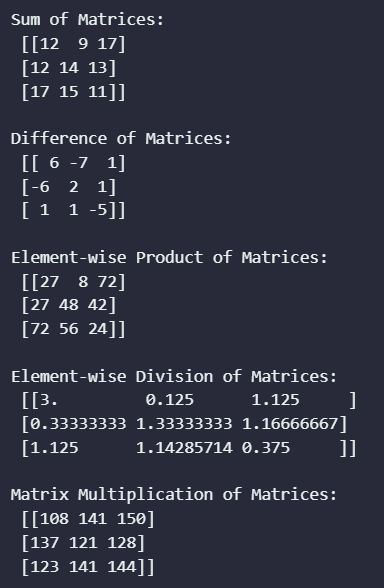
print("\nElement-wise Product of Matrices: \n", arr\_1 \* arr\_2)

print("\nElement-wise Division of Matrices: \n", arr\_1 / arr\_2)

print("\nMatrix Multiplication of Matrices: \n", arr\_1 @ arr\_2)

### Task 6 Code Ends Here ###

### Task 6 Screenshot Starts Here ###



### Task 6 Screenshot Ends Here ###

# Conclusion

In this lab report, we introduced the basic data structures and operations in Python, including lists and dictionaries. We also provided a brief introduction to NumPy, a popular Python library for scientific computing. We learned how to create, alter, and loop through lists, as well as how to use slicing to access a range of items in a list. We also learned how to use various list methods, such as append, insert, extend, remove, and pop. Additionally, we learned how to implement 2-D lists and dictionaries. Finally, we got a basic introduction to NumPy, including how to create and use NumPy arrays.