**AI LAB ASSIGNMENT REPORT**

**On**

(BCA-17-308 “AI LAB”)

****

**Delhi COLLEGE OF TECHNOLOGY AND**

**MANAGEMENT**

**PALWAL**

**Session: JAN 2025 - MAY 2025**

**Submitted To: Submitted By:**

**Mrs.** **Nalini Sikri** **Student Name : Sumit Sharma**

**Assistant Professor, CSE Enrollment No. : 22034311052**

**DELHI COLLEGE TECHNOLOGY & MANAGEMENT, Palwal**

**SUBJECT-AI**

**BCA 6th sem**

**LIST OF EXPERIMENT**

|  |  |  |
| --- | --- | --- |
| S.NO | Experiment | REMRAKS |
| 1 | To find reverse of a list |  |
| 2 | To find factorial of a number |  |
| 3 | Write a program to print Fibonacci series |  |
| 4 | A program to find the distance using roadmap |  |
| 5 | To sort the list using Merge sort |  |
| 6 | Write a program to show union and intersection of two lists. |  |
| 7 | Write a program to concatenate two string of different lengths. |  |
| 8 | A program to find whether two lists are equal or not |  |
| 9 | A program to implement selection sort |  |
| 10 | A program to sort a list using insertion sort algorithm |  |

**Experiment-1**

AIM:- To find reverse of a list.

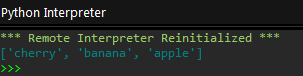
**Code:-**

fruits = ['apple', 'banana', 'cherry']

fruits.reverse()

print(fruits)

**Output**



**Experiment-2**

AIM:- To find factorial of a number.

**Code:-**

#Import math Library

import math

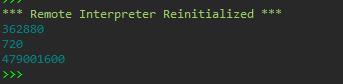
#Return factorial of a number

print(math.factorial(9))

print(math.factorial(6))

print(math.factorial(12))

**Output**

****

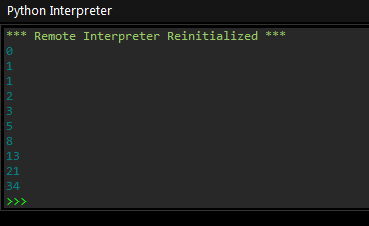
**Experiment-3**

AIM:- Write a program to print Fibonacci series

**Code:-**

def fibonacci(n):  
  if n == 0:  
    return 0  
  elif n == 1:  
    return 1  
  else:  
    return fibonacci(n-1) + fibonacci(n-2)  
  
*# Generate the first 10 numbers in the Fibonacci series*for i in range(10):  
  print(fibonacci(i))

**Output**

****

**Experiment- 4**

AIM:- A Program to find the distance using roadmap

**Code:-**

# Import math Library

import math

p = [3]

q = [1]

# Calculate Euclidean distance

print (math.dist(p, q))

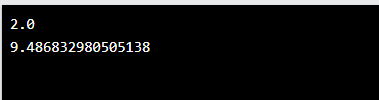
p = [3, 3]

q = [6, 12]

# Calculate Euclidean distance

print (math.dist(p, q))

**Output**

****

**Experiment- 5**

AIM:- A To sort the list using Merge Sort

**Code:-**

def merge(arr, l, m, r):

    n1 = m - l + 1

    n2 = r - m

    L = [0] \* (n1)

    R = [0] \* (n2)

    for i in range(0, n1):

        L[i] = arr[l + i]

     for j in range(0, n2):

        R[j] = arr[m + 1 + j]

        i = 0     # Initial index of first subarray

    j = 0     # Initial index of second subarray

    k = l     # Initial index of merged subarray

     while i < n1 and j < n2:

        if L[i] <= R[j]:

            arr[k] = L[i]

            i += 1

        else:

            arr[k] = R[j]

            j += 1

        k += 1

     while i < n1:

        arr[k] = L[i]

        i += 1

        k += 1

         while j < n2:

        arr[k] = R[j]

        j += 1

        k += 1

def mergeSort(arr, l, r):

    if l < r:

         m = l+(r-l)//2

         mergeSort(arr, l, m)

        mergeSort(arr, m+1, r)

        merge(arr, l, m, r)

 arr = [12, 11, 13, 5, 6, 7]

n = len(arr)

print("Given array is")

for i in range(n):

    print("%d" % arr[i],end=" ")

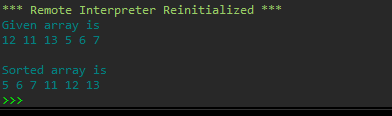
 mergeSort(arr, 0, n-1)

print("\n\nSorted array is")

for i in range(n):

    print("%d" % arr[i],end=" ")

**Output**

****

**Experiment- 6**

AIM:- Write a Program to show union and Intersection of two lists.

**Code:-**

A = {0, 2, 4, 6, 8};

B = {1, 2, 3, 4, 5};

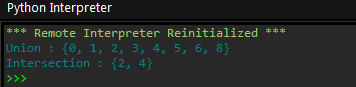
# union

print("Union :", A | B)

# intersection

print("Intersection :", A & B)

**Output**

****

**Experiment- 7**

AIM:-. Write a program to concatenate two strings of different lengths

**Code:-**

# Defining strings

str1 = "Delhi College of "

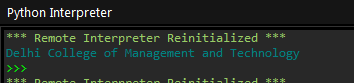
str2 = "Technology and Management"

# + Operator is used to strings concatenation

str3 = str1 + str2

**print**(str3)   # Printing the new combined string

**Output:-**

****

**Experiment- 8**

AIM:-. A Program to find whether two lists are equal or not

**Code:-**

list1 = [11, 12, 13, 14, 15]

list2 = [12, 13, 11, 15, 14]

a = set(list1)

b = set(list2)

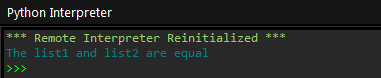
**if** a == b:

**print**("The list1 and list2 are equal")

**else**:

**print**("The list1 and list2 are not equal")

**Output**

****

**Experiment- 9**

AIM:-. A Program to implement selection sort.

**Code:-**

def selection\_sort(array):

    length = len(array)

**for** i in range(length-1):

        minIndex = i

**for** j in range(i+1, length):

**if** array[j]<array[minIndex]:

                minIndex = j

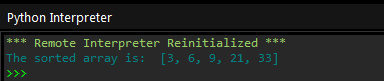
        array[i], array[minIndex] = array[minIndex], array[i]

**return** array

array = [21,6,9,33,3]

print("The sorted array is: ", selection\_sort(array))

**Output**

****

**Experiment- 10**

AIM:-. A Program to sort a list using insertion sort algorithm

**Code:-**

# creating a function for insertion

def insertion\_sort(list1):

               for i in range(1, len(list1)):

              value = list1[i]

            j = i - 1

            while j **>**= 0 and value **<** **list1**[j]:

                list1[j + 1] = list1[j]

                j -= 1

            list1[j + 1] = value

        return list1

list1 = [10, 5, 13, 8, 2]

print("The unsorted list is:", list1)

print("The sorted list1 is:", insertion\_sort(list1))

**Output**

