

USN											
Subject:	<b>PYTHON PROGRAMMING LABORATORY</b>									Sub Code:	21CSL46
Date:	01-08-2023	Duration:	120Min's	Sem/Sec:	IV Sem-A						

**Note: Answer any Five Full Questions**

1 a.# Reading three test marks one after the other

```
marks_1_test = int (input("Enter the marks in the first test: "))
marks_2_test = int (input("Enter the marks in second test: "))
marks_3_test = int (input("Enter the marks in third test: "))
```

**# Get best two test marks**

```
if (marks_1_test > marks_2_test):
    if (marks_2_test > marks_3_test):
        totalOfBest2 = marks_1_test + marks_2_test
    else:
        totalOfBest2 = marks_1_test + marks_3_test
elif (marks_1_test > marks_3_test):
    totalOfBest2 = marks_1_test + marks_2_test
else:
    totalOfBest2 = marks_2_test + marks_3_test
```

**# Find average of best two tests and output the same to user's monitor**

```
AverageOfBest2 = totalOfBest2 / 2
```

```
print ("The average of the best two test marks is: ", AverageOfBest2)
```

1 b.

**Program:**

**# get the numeric number from the user through keyboard**

```
number_given = int(input("\nEnter the number: "))
```

```
temp_number=number_given #preserve the given number
```

```
reverse_number=0 #
```

**#reversing the number**

```
while (number_given>0):
```

```
    last_digit=number_given%10
```

```

reverse_number=reverse_number*10+last_digit

number_given=number_given//10

#checking whether both numbers are same
if(temp_number==reverse_number):

    print("\nThe number is palindrome!")

else:

    print("\nThe Number is Not a palindrome!")

number_String=str(temp_number)

for digit in range(10):

    if number_String.count(str(digit))!=0:
        print('Number of count of', digit, 'is:', number_String.count(str(digit)))

```

## 2 a Write a python program to implement insertion sort and merge sort using lists

### Algorithm:

### Program:

#### **#function for Insertion sort**

```

def insertionSort(input_list):

    for index in range(1,len(input_list)):

        current_value = input_list[index]

        current_position = index

#camparing the current_postion value with the previous value(current_position-1 )

        while current_position>0 and input_list[current_position-1]>current_value:

            input_list[current_position]=input_list[current_position-1]

            current_position = current_position-1

        input_list[current_position]=current_value

#Input for Insertion sort

input_list = [14,46,43,27,57,41,45,21,70]

insertionSort(input_list)

print(input_list)

```

**output :**

The sorted list is [14, 21, 27, 41, 43, 45, 46, 57, 70]

**#function for Merge sort**

```
def mergeSort(input_list):
    print("Splitting ",input_list)
    if len(input_list)>1:

        mid_value = len(input_list)//2
        left_half = input_list[:mid_value]
        right_half = input_list[mid_value:]

        mergeSort(left_half)
        mergeSort(right_half)
        i=j=k=0
        while i < len(left_half) and j < len(right_half):
            if left_half[i] < right_half[j]:
                input_list[k]=left_half[i]
                i=i+1
            else:
                input_list[k]=right_half[j]
                j=j+1
            k=k+1

        while i < len(left_half):
            input_list[k]=left_half[i]
            i=i+1
            k=k+1

        while j < len(right_half):
            input_list[k]=right_half[j]
```

```
        j=j+1
        k=k+1

    print("Merging ",input_list)
#Input for Merge sort
input_list = [14,46,43,27,57,41,45,21,70]
mergeSort(input_list)
print(input_list)
```

**b) Write a program to convert roman numbers in to integer values using dictionaries.**

**Algorithm:**

**Program:**

```
class int_conversion:
# function for integer to roman conversion
    def roman_to_int(self, input_value):
        rom_val = {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}
        int_val = 0
        for val in range(len(input_value)):
            if val > 0 and rom_val[input_value[val]] > rom_val[input_value[val - 1]]:
                int_val += rom_val[input_value[val]] - 2 * rom_val[input_value[val - 1]]
            else:
                int_val += rom_val[input_value[val]]
        return int_val

print(int_conversion().roman_to_int('MMMCMLXXXVI'))
print(int_conversion().roman_to_int('MMMM'))
print(int_conversion().roman_to_int('C'))
```