PYTHON LAB PROGRAM 4

4) AIM: Discuss different collections like list, tuple and dictionaries

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

```
def insertion_sort(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1
        while j >= 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key
    return arr

def merge_sort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        left_half = arr[:mid]
        right_half = arr[mid:]
        merge_sort(left_half)
        merge_sort(right_half)
        i = j = k = 0
        while i < len(left_half) and j < len(right_half):
            if left_half[i] < right_half[j]:
                arr[k] = left_half[i]
                i += 1
            else:
                arr[k] = right_half[j]
                j += 1
            k += 1

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

        while j < len(right_half):
            arr[k] = right_half[j]
            j += 1
            k += 1
        return arr
my_list = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
sorted_list = insertion_sort(my_list)
print("Sorted list using Insertion sort:",sorted_list)
sorted_list = merge_sort(my_list)
print("Sorted list using Merge sort:",sorted_list)
```

Output:

```
Sorted list using Insertion sort: [1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9]
Sorted list using Merge sort: [1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9]
```

b) Write a python program to convert Roman numbers into integer numbers usingdictionaries

```
class py_solution:
    def roman_to_int(self, s):
        rom_val = {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}
        int_val = 0
        for i in range(len(s)):
            if i > 0 and rom_val[s[i]] > rom_val[s[i - 1]]:
                int_val += rom_val[s[i]] - 2 * rom_val[s[i - 1]]
            else:
                int_val += rom_val[s[i]]
        return int_val

print(py_solution().roman_to_int('MMMCMLXXXVI'))
print(py_solution().roman_to_int('VIII'))
print(py_solution().roman_to_int('C'))
```

OUTPUT:

```
3986
8
100
```