Assignment No:10

Q.1.#WAP to find sum of all elements of list

```
li=[10,20,30,40,50,60]
sum=0
for i in range(0,len(li)):
    sum=sum+li[i]


print(f'Sum of list is:{sum}')
```

Q.2.#WAP to find maximum and minimum element in list

```
li=[30,44,57,20,100]
max=li[0]
min=li[0]
for ele in li:
    if(max<ele):
        max=ele
    elif(min>ele):
        min=ele
print("Max element:",max)
print("Min element:",min)
```

Q.3.#WAP to find second largest element in the list

```
li=[5,3,7,2,9,10,20,15]

max=li[0]
smax=0
for i in range(0,len(li)):
    if(max<li[i]):
        smax=max
        max=li[i]
    elif(smax<li[i]):
        smax=li[i]

print("second Maximum:",smax)
```

Q.4.#WAP to reverse the list

```
li=[50,60,70,80,90,100]
reversed_list=[]
length=len(li)
for i in range(length-1,-1,-1):
    reversed_list.append(li[i])

print("Original list:",li)
print("Reversed list:",reversed_list)
```

Q.5.#Accept a number from user and check if this element is present in the list or not.
#Also tell how many times it is present in the list

```
def count_ele(li,ele):
    count=0
    for i in li:
        if(i==ele):
            count+=1
    return count

li=[1,2,3,6,2,7,2,8,2,9]
num=int(input("Enter number to find:"))
print("count:",count_ele(li,num))
```

Q.6.#wap to remove duplicates from list

```
def remove_duplicate(list):
    li1=[]
    for i in list:
        if i not in li1:
            li1.append(i)
    return li1

list1=[5,6,3,4,2,5,7,8]
li1=remove_duplicate(list1)
print(f'List after removing duplicate elements:{li1}')
```

Q.7.#wap to create a new list from existing list which contains cube of each number of list

```
def cube_of_list(existing_list):
    cubed_list=[]
    for num in existing_list:
        cube=num*num*num
        cubed_list.append(cube)
    return cubed_list

existing_list=[2,4,5,6,7]
cubed_list=cube_of_list(existing_list)

print("Original list:",existing_list)
print("Cubed_list:",cubed_list)
```

Q.8.#wap to create a duplicate of an existing list.it should not point to same list.

```
def duplicate_list(original_list):
    return original_list.copy()

original_list=[10,20,30,40,70 ]
copied_list=duplicate_list(original_list)
print("Original list:",original_list)
print("Copied list:",copied_list)
```

Q.9.#wap of having n number of elements in the list and find out even
#and odd elements in that list and then create two seperate lists which will have
#even elements and other will have odd elements

```
def even_odd(li):
    even_list=[]
    odd_list=[]
    for ele in li:
        if(ele%2==0):
            even_list.append(ele)
        else:
            odd_list.append(ele)
    return even_list,odd_list

li=[11,12,13,14,15,16]
even,odd=even_odd(li)
```

```
print("Even list:",even)
print("Odd list:",odd)
```

Q.10.#wap to remove all occurences of a given element in the list

```
def remove_element(li,ele):
    new_li=[]
    for item in li:
        if(item!=ele):
            new_li+=[item]
    return new_liQ.

li=[2,4,3,2,5,6]
ele_to_remove=2
output=remove_element(li,ele_to_remove)
print("List before removing:",li)
print("List after removing:", output)
```

Q.11.#wap to print all numbers which are divisible by m and n in the list

```
def divisible_number(li,m,n):
    i=0
    while(i<len(li)):
        if(li[i]%m==0 and li[i]%n==0):
            print(li[i])
        i+=1

li=[0,2,4,6,8,12,16,20,5]
m=2
n=4
print(f"Numbers divisible by {m} and {n} are:")
divisible_number(li,m,n)
```

Q.12.#wap to create three lists of numbers,their squares and cubes.

```
def list(n):
    numbers=[]
    squares=[]
    cubes=[]
```

```
    for i in range(1,n+1):
        numbers.append(i)
        squares.append(i*i)
        cubes.append(i*i*i)

    return numbers,squares,cubes

n=int(input("Enter a number:"))
numbers,squares,cubes=list(n)

print("Numbers:",numbers)
print("Squares:",squares)
print("Cubes:",cubes)
```

Q.13.#wap to create a duplicate of an existing list

```
#It should not point to same list
def duplicate_list(original_list):
    return original_list.copy()

original_list=[10,20,30,40,50]
copied_list=duplicate_list(original_list)
print("Original list:",original_list)
print("Copied list",copied_list)
```