PRACTICAL NO 4

a) Write a program that takes two lists and returns True if they have at least one common member.

Code :

```
#function to input the list element from user
def ListInput(li) :
    Size = int(input("Enter the size of the List : "))
    for x in range(Size) :
        y=int(input(f"Enter Element E[{x+1}] : "))
        li.append(y)

#function to check at least one common member
#start
def CheckElement(L1,L2):
    for z in L1:
        if z in L2:
            return True
            break
#end

L1=list()
L2=list()
print("First List ::")
ListInput(L1)
print("")
print("Second List ::")
ListInput(L2)
print("")
if CheckElement(L1,L2):
    print("Element found!")
else :
    print("Element not found!")
```

Output :

```
First List ::
Enter the size of the List : 5
Enter Element E[1] : 15
Enter Element E[2] : 29
Enter Element E[3] : 30
Enter Element E[4] : 98
Enter Element E[5] : 23

Second List ::
Enter the size of the List : 4
Enter Element E[1] : 24
Enter Element E[2] : 56
Enter Element E[3] : 30
Enter Element E[4] : 15

Element found!
```

b) Write a Python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements.

Code :

```
#function to input the list element from user
def ListInput(li) :
    Size = int(input("Enter the size of the List : "))
    for x in range(Size) :
        y=int(input(f"Enter Element E[{x+1}] : "))
        li.append(y)

#function to remove element at a index from list
#start
def RemoveElement(L):
    Confirm=1
    while Confirm==1 :
        Index=int(input("Enter index number to be deleted : "))
        if len(L)==0 :
            print("")
            print("your list is already empty")
            exit(0)
        else :
            try :
                if L.pop(Index) :
                    print("Element remove successfully")
                    print("Your list after deleting specified element
                    is follows : ")
                    print(L)
            except IndexError :
                print("Pop index out of range")
            print("")
            Confirm=int(input("Do you want to delete again ? [1|Yes
            2|No] : "))
#end

L=list()
ListInput(L)
print("")
RemoveElement(L)
```

Output :

```
Enter the size of the List : 5
Enter Element E[1] : 10
Enter Element E[2] : 20
Enter Element E[3] : 30
Enter Element E[4] : 40
Enter Element E[5] : 50

Enter index number to be deleted : 4
Element remove successfully
Your list after deleting specified element is follows :
[10, 20, 30, 40]

Do you want to delete again ? [1|Yes  2|No] : 1
Enter index number to be deleted : 20
Pop index out of range
```

```
Do you want to delete again ? [1|Yes  2|No] : 1
Enter index number to be deleted : 1
Element remove successfully
Your list after deleting specified element is follows :
[10, 30, 40]

Do you want to delete again ? [1|Yes  2|No] : 1
Enter index number to be deleted : -1
Element remove successfully
Your list after deleting specified element is follows :
[10, 30]

Do you want to delete again ? [1|Yes  2|No] : 1
Enter index number to be deleted : 0
Element remove successfully
Your list after deleting specified element is follows :
[30]

Do you want to delete again ? [1|Yes  2|No] : 1
Enter index number to be deleted : 0
Element remove successfully
Your list after deleting specified element is follows :
[]

Do you want to delete again ? [1|Yes  2|No] : 1
Enter index number to be deleted : 0
your list is already empty
```

c) Write a Python program to clone or copy a list

Code :

```
#function to input the list element from user
def ListInput(L) :
    Size = int(input("Enter the size of the List : "))
    for x in range(Size) :
        y=int(input(f"Enter Element E[{x+1}] : "))
        L.append(y)

#function to clone the first list
def ListClone(L):
    L2=list()
    L2=L.copy()
    return L2

L=list()
ListInput(L)
print("")
print(f"Entered list :: {L}")
print(f"Clone list   :: {ListClone(L)}")
```

Output :

```
Enter the size of the List : 5
Enter Element E[1] : 10
Enter Element E[2] : 20
Enter Element E[3] : 30
Enter Element E[4] : 40
Enter Element E[5] : 50

Entered list :: [10, 20, 30, 40, 50]
Clone list   :: [10, 20, 30, 40, 50]
```