

[ Question 1 ]

```
l = [8, 9, 10]
```

```
print(l)
l[2] = 17      #a
print(f"a. Set the second index to 17: {l}\n")
```

```
      [8, 9, 10]
      a. Set the second index to 17: [8, 9, 17]
```

```
l.extend([4, 5, 6]) #b
print(f"b. Add 4, 5, and 6 to the end of the list: {l}\n")
```

```
      b. Add 4, 5, and 6 to the end of the list: [8, 9, 17, 4, 5, 6]
```

```
l.pop(0) #c
print(f"c. Remove the first entry from the list: {l}\n")
```

```
      c. Remove the first entry from the list: [9, 17, 4, 5, 6]
```

```
l.sort() #d
l.reverse()
print(f"d. Sort the list in descending order: {l}\n")
```

```
      d. Sort the list in descending order: [17, 9, 6, 5, 4]
```

```
l.insert(3, 25) #e
print(f"e. Insert 25 at index 3: {l}\n")
```

```
      e. Insert 25 at index 3: [17, 9, 6, 25, 5, 4]
```

```
print(f"f. ??\n")
```

```
      f. ??
```

```
l.reverse() #g
print(f"g. Reverse the List: {l}\n")
```

```
      g. Reverse the List: [4, 5, 25, 6, 9, 17]
```

```
print(f"h. Print the list: {l}\n") #h
```

```
      h. Print the list: [4, 5, 25, 6, 9, 17]
```

```
print("i. Print the count of each element in the list: ")
for i in list(set(l)): #i
    print(f"{i} : {l.count(i)}")
print()
```

```
      i. Print the count of each element in the list:
      4 : 1
      5 : 1
      6 : 1
      9 : 1
      17 : 1
      25 : 1
```

```
print(f"j. Maximum : {max(l)}\nMinimum : {min(l)}\n") #j
```

```
      j. Maximum : 25
      Minimum : 4
```

```
l2 = [1, 2, 5, 3, 4, 6, 7, 12, 8]

print(f"k. l2 = {l2}")
print(f"k. Extract the first five elements from the list: {l2[:5]}\n")
print(f"k. Extract the elements from the 3rd to the 7th position (inclusive) : {l2[3:8]}\n")
print(f"k. Extract the last three elements from the list: {l2[-3:]}\n")

k. l2 = [1, 2, 5, 3, 4, 6, 7, 12, 8]
k. Extract the first five elements from the list: [1, 2, 5, 3, 4]

k. Extract the elements from the 3rd to the 7th position (inclusive) : [3, 4, 6, 7, 12]

k. Extract the last three elements from the list: [7, 12, 8]
```

## [ Question 2 ]

```
n = int(input("Enter the size of the List: "))
l3 = [None]*n

Enter the size of the List: 10

for i in range(n):
    l3[i] = int(input(f"Enter the Value for {i}th Element: "))

Enter the Value for 0th Element: 5
Enter the Value for 1th Element: 2
Enter the Value for 2th Element: 54
Enter the Value for 3th Element: 25
Enter the Value for 4th Element: 32
Enter the Value for 5th Element: 12
Enter the Value for 6th Element: 96
Enter the Value for 7th Element: 54
Enter the Value for 8th Element: 21
Enter the Value for 9th Element: 35

def Listpop(l, index):
    """
    l2 = list()
    for i in range(len(l)):
        if i == index:
            continue
        else:
            l2.append(l[i])
    """
    l2 = [i for i in l if l.index(i) != index]
    return l2

index = int(input("Enter an Index to pop the Element: "))
if (index not in range(n)) and (index not in range(-1, -n-1, -1)):
    print(index not in range(n), index not in range(-1, -n-1, -1))
    print("IndexError: Index out Bounds")
else:
    l4 = Listpop(l3, index)
    print(f"After popping the value at Index {index}: {l4}")

Enter an Index to pop the Element: 5
After popping the value at Index 5: [5, 2, 54, 25, 32, 96, 54, 21, 35]
```

## [ Question 3 ]

```
def Listswap(l, ind1, ind2):
    l[ind1-1], l[ind2-1] = l[ind2-1], l[ind1-1]
    return l

List = [23, 65, 19, 90]
pos1 = 1
pos2 = 3
print(f"List: {List}\nPos1: {pos1}\nPos2: {pos2}\nAfter swapping: {Listswap(List, pos1, pos2)}")
```

```
List: [23, 65, 19, 90]
Pos1: 1
Pos2: 3
After swapping: [19, 65, 23, 90]
```

## [ Question 4 ]

```
def ListHalfswap(l):
    if not len(l)%2:
        l[len(l)//2:], l[:len(l)//2] = l[:len(l)//2], l[len(l)//2:]
    else:
        l[len(l)//2+1:], l[:len(l)//2] = l[:len(l)//2], l[len(l)//2+1:]
    return l

l = [1, 2, 3, 4, 5, 6, 7, 8, 9]
print(f"List before swapping: {l}\nList after swapping: {ListHalfswap(l)}")
l = [1, 2, 3, 4, 5, 6, 7, 8]
print(f"List before swapping: {l}\nList after swapping: {ListHalfswap(l)}")

List before swapping: [1, 2, 3, 4, 5, 6, 7, 8, 9]
List after swapping: [6, 7, 8, 9, 5, 1, 2, 3, 4]
List before swapping: [1, 2, 3, 4, 5, 6, 7, 8]
List after swapping: [5, 6, 7, 8, 1, 2, 3, 4]
```

## [ Question 5 ]

```
def peak(l):
    return max(l)

def peak2(l):
    if len(l)<=3:
        return peak(l)
    if l[0]>l[1]:
        return l[0]
    if l[-1]>l[-2]:
        return l[-1]
    for i in range(1, len(l)-1):
        if l[i] == max(l[i-1], l[i], l[i+1]):
            return l[i]

l = [10, 20, 15, 2, 23, 90, 67]
print(f"Peak of the List {l} is {peak(l)}")
print(f"Peak of the List {l} is {peak2(l)}")

Peak of the List [10, 20, 15, 2, 23, 90, 67] is 90
Peak of the List [10, 20, 15, 2, 23, 90, 67] is 20
```

## [ Question 6 ]

```
def list2(l):
    l2 = [i for i in l if l.count(i)==1]
    l3 = list(set([i for i in l if l.count(i)!=1]))
    return l2, l3

l = [1, 2, 3, 1, 2, 5, 4, 6, 3, 5, 0, 4, 8, 9, 7, 10]
l2, l3 = list2(l)
print(f"The Unique values of List l={l} are L2={l2}\nand Non Unique Values of the List are L3={l3}.")

The Unique values of List l=[1, 2, 3, 1, 2, 5, 4, 6, 3, 5, 0, 4, 8, 9, 7, 10] are L2=[6, 0, 8, 9, 7, 10]
and Non Unique Values of the List are L3=[1, 2, 3, 4, 5].
```

## [ Question 7 ]

```
def rotate(l, x):
    return l[len(l)-x:] + l[:len(l)-x]
```

```

l = [1, 2, 3, 4, 5, 6]
x=2
desired_output = [5, 6, 1, 2, 3, 4]
print(f"The List l={l} has been rotated {x} times and obtained: {rotate(l, x)}\nwhich is the same as the desired Output: {desired_output}.")

The List l=[1, 2, 3, 4, 5, 6] has been rotated 2 times and obtained: [5, 6, 1, 2, 3, 4]
which is the same as the desired Output: [5, 6, 1, 2, 3, 4].

```

## [ Question 8 ]

```

def sumpair(l1, l2, sum_):
    pairs=[]
    for i in l1:
        for j in l2:
            if i+j == sum_:
                pairs.append((i, j))
    return pairs

list1 = [-1, -2, 4, -6, 5, 7]
list2 = [6, 3, 4, 0]
x = 8

res = sumpair(list1, list2, x)

print(f"The Pairs of sums of elements from the Lists {list1} and {list2} are:\n{res}")

The Pairs of sums of elements from the Lists [-1, -2, 4, -6, 5, 7] and [6, 3, 4, 0] are:
[(4, 4), (5, 3)]

```

## [ Question 9 ]

```

def disp(mat):
    for i in mat:
        for j in i:
            print("{:<7}".format(j), end="")
        print()

mat = [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]]
print("Matrix Display:")
disp(mat)

Matrix Display:
1      2      3      4
5      6      7      8
9      10     11     12
13     14     15     16

def sumrow(mat):
    arr=[]
    for i in mat:
        arr.append([sum(i)])
    return arr

print("\n\nSum of Rows:")

Sum of Rows:

disp(sumrow(mat))

10
26
42
58

```

```
def sumcol(mat):
    arr=[]
    for i in range(len(mat[0])):
        arr.append(sum([j[i] for j in mat]))
    return [arr]

print("\n\nSum of Columns:")
disp(sumcol(mat))
```

```
Sum of Columns:
28      32      36      40
```

```
def summat(mat1, mat2):
    mat=[]
    for i, j in enumerate(mat1):
        row=[]
        for k, l in enumerate(j):
            row.append(l+mat2[i][k])
        mat.append(row)
    return mat
```

```
mat1 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
mat2 = [[1, 0, 0], [0, 1, 0], [0, 0, 1]]
```

```
print("\n\nMatrix 1:")
disp(mat1)
print("\n\nMatrix 2")
disp(mat2)
print("\n\nSum of Matrices")
disp(summat(mat1, mat2))
```



```
Matrix 1:
1      2      3
4      5      6
7      8      9
```

```
Matrix 2
1      0      0
0      1      0
0      0      1
```

```
Sum of Matrices
2      2      3
4      6      6
7      8     10
```