Duplicate Number

```
K = int(input("Enter Number of Test Case: "))

while(K>0):
    ls = sorted(list(map(int,input("Enter List: ").strip().split())))
    for i in range(len(ls)-1):
        if ls[i] == ls[i+1]:
            print(ls[i])
            break
    K -= 1
```

```
Enter Number of Test Case: 2
Enter List: 1 3 4 2 2
2
Enter List: 3 1 3 4 2
3
```

Maximum SubArray

```
def maxSubarray(ls,n):
    gsum,sumi,l,h,gl,gh= ls[0],ls[0],0,0,0,0
    for i in range(1,n):
        if sumi + ls[i] > ls[i]:
            sumi +=ls[i]
            h = i
        elif ls[i] > sumi:
            sumi,l,h = ls[i],i,i
        if gsum < sumi:
                gsum,gl,gh = sumi,l,h

    return (gsum,gl,gh)
```

```
N = int(input("Enter Number of Test Cases: "))
while N>0:
    ls = list(map(int,input("Enter List: ").strip().split()))
    sumi,l,h = maxSubarray(ls,len(ls))
    print(f"Maximum Subarray = {ls[l:h+1]}, Sum = {sumi}")
    print()
    N -= 1
```

```
Enter Number of Test Cases: 3
Enter List: -1 2 3 -4 5 10
Maximum Subarray = [2, 3, -4, 5, 10], Sum = 16

Enter List: 1 2 -2 -2 5
Maximum Subarray = [5], Sum = 5

Enter List: -2 -2 3 4 5 -2 -1
Maximum Subarray = [3, 4, 5], Sum = 12
```

Merging Similar Intervals

```
K = int(input("Enter Number of Test Case: "))
while(K>0):
    N = int(input("Enter the number of Intervals: "))

    ls = []
    while(N>0):
        ls.append(list(map(int,input("Enter the Interval: ").strip().split())))
        N -=1

    print("List before Merging: ",ls)
    i = 0
    while(i<len(ls)-1):
        if ls[i][1] >= ls[i+1][0]:
            ls[i][1] = ls[i+1][1]
            del ls[i+1]
        else:
            i += 1

    print("List After Merging: ",ls)
    K -= 1
```

```
Enter Number of Test Case: 2
Enter the number of Intervals: 4
Enter the Interval: 1 3
Enter the Interval: 2 6
Enter the Interval: 8 10
Enter the Interval: 15 18
List before Merging:  [[1, 3], [2, 6], [8, 10], [15, 18]]
List After Merging:  [[1, 6], [8, 10], [15, 18]]
Enter the number of Intervals: 2
Enter the Interval: 1 4
Enter the Interval: 4 5
List before Merging:  [[1, 4], [4, 5]]
List After Merging:  [[1, 5]]
```

Merging Sorted list without using extra list

```
K = int(input("Enter Number of Test Case: "))

while(K>0):
    print()
    arr1 = list(map(int,input("Enter List 1 in ascending order: ").strip().split()))
    arr2 = list(map(int,input("Enter List 2 in ascending order: ").strip().split()))

    i,j = 0,0

    while(i<len(arr1) and j < len(arr2)):
        if arr1[i] > arr2[j]:
            arr1.insert(i,arr2[j])
            j +=1
        else:
            i += 1

    while j < len(arr2):
        arr1.append(arr2[j])
        j+=1

    print("Merged List: ",arr1)

    K -=1
```

```
Enter Number of Test Case: 3

Enter List 1 in ascending order: 1 3 5 7
Enter List 2 in ascending order: 0 2 6 8 9
Merged List:  [0, 1, 2, 3, 5, 6, 7, 8, 9]

Enter List 1 in ascending order: 0 1 2 3
Enter List 2 in ascending order: 5 6 7 8 9
Merged List:  [0, 1, 2, 3, 5, 6, 7, 8, 9]

Enter List 1 in ascending order: 0 7 69 150 989 1000
Enter List 2 in ascending order: 1 69 79 80
Merged List:  [0, 1, 7, 69, 69, 79, 80, 150, 989, 1000]
```

Next Perumutation

```
def nextPermutation(ls,n):
    i,j = n,n
    while i>0 and ls[i]<ls[i-1]:
        i -= 1

    if i == 0:
        return sorted(ls)
    else:
        i = i-1

    while j > i and ls[j] < ls[i]:
        j -= 1

    ls[j],ls[i] = ls[i],ls[j]

    return ls[:i+1]+list(reversed(ls[i+1:]))
```

```
nextPermutation([3,2,1],2)
```

```
[1, 2, 3]
```

```
nextPermutation([1,2,3],2)
```

```
[1, 3, 2]
```

```
nextPermutation([3,2,6,5,4,1],5)
```

```
[3, 4, 1, 2, 5, 6]
```