```
``````````````````````````A = [1,2,8,5,7,100]           Q1
A[0]=100
A[5]=1                        # INTERCHANGE IS DONE BY INDEXING
print(A)
```

```
[100, 2, 8, 5, 7, 1]
```

```
A = [1,2,8,5,7,100]             Q2
A[0]= 2
A[1]=1                # sawping of  elemnts by indexing in list
print(A)
```

```
[2, 1, 8, 5, 7, 100]
```

```
A = ['apple','grapes','mango','orange']          Q3
A[0]='grapes'
A[1]='apple'
A[2]='orange'                                    # swaping elements in
string list by indexing
A[3]='mango'
print(A)
```

```
['grapes', 'apple', 'orange', 'mango']
```

```
A = [1,2,8,5,7,100]      # lenght of list    by len() Q4
print(len(A))
```

```
6
```

```
A = [1,2,8,5,7,100]
B = [3,4,8,7,1010,65]
print(max(A))           Q5 #max num
print(max(B))
```

```
100
1010
```

```
A = [1,2,8,5,7,100]        Q6 min
B = [3,4,8,7,1010,65]
print(min(A))
print(min(B))
```

```
1
3
```

```
A = [1,2,8,5,7,100]
print(A[2])                                          #Q7CHECKING
ELEMENT IN LIST
print(A[4])
print(A[1:])
```
```

```
print(A[:1])
print(A[0:6])
```

```
8
7
[2, 8, 5, 7, 100]
[1]
[1, 2, 8, 5, 7, 100]
```

```
A = [1,2,8,5,7,100]
A.clear()                          Q8 # CLEARING OF LIST
print(A)
```

```
[]
```

```
A = [1,2,8,5,7,100]
del A[0:6]                                  # Q8 CLEARING OF LIST
print(A)
```

```
[]
```

```
A = [1,2,8,5,7,100]              # Q9 REVERSING THE LIST
A.sort( reverse = True)
A
```

```
[100, 8, 7, 5, 2, 1]
```

```
A = [1,2,8,5,7,100]                                                    #Q10
COPY THE LIST
slice(A)
A
```

```
[1, 2, 8, 5, 7, 100]
```

```
A = [1,2,8,5,7,100]                                                    #Q10
COPY THE LIST
A.copy()
A
```

```
[1, 2, 8, 5, 7, 100]
```

```
A = [1,2,8,5,7,100,4,4,8,4,4,4]
A.count(1)
A.count(4)                     # Q11 count ocurence of element
list.count
A .count(8)
A.count(4)
```

```
5
```

```
A = [1,2,8,7,4,4,4,4,4,100]
Y = A.count(2)
Y                     #Q11     count number
```

```
1
```

```
A = [1,2,8,5,7,100]
o = sum(A)
o                                     #Q12 FIRST WE HAVE FIND  SUM OF
LIST  STOREE IN O THEN FOR AVG WE DIVIDE O BY 2
```

```
123
```

```
A = [1,2,8,5,7,100]
o = sum(A)
o                                     #Q12 FIRST WE HAVE FIND  SUM OF
LIST  STOREE IN O THEN FOR AVG WE DIVIDE O BY 2
Avg = o/2
Avg
```

```
61.5
```

```
A = [1,2,8,5,7,100,459]     #Q13 SUM OF ALL DIGITS IN LIST
sum(A)
```

```
582
```

```
A = [1,2,8,5,7,100,459]     #Q14 multiply OF ALL DIGITS IN LIST
s=(1*2*8*5*7*100*459)
s
```

```
25704000
```

```
A = [1,2,8,5,7,100,459] #Q15 SMALLEST NUMBER IN A LIST
print(min(A))
```

```
1
```

```
A = [1,2,8,5,7,100,459]     #Q16 LARGEST NUMBER IN A LIST
print(max(A))
```

```
459
```

```
A = [1,2,8,5,7,100,459]
print(max(A))
A.pop()              # Q17 FOR SECOND LARGEST NUMBER FIRST WE FIND THE
MAXIMUM NUMBER AND PRINT IT AFTER THAT WE USE POP TO REMOVE FIRST
LARGEST NUMBER AND THEN WE USE AGAIN MAX FUNCTION TO FIND SECOND
LARGEST NUMBER
A
print(max(A))
```

```
459
100
```

```
n = [1,2,3,4,5,6,7,8,9,10,11]             #Q18 program to print all
even numbers in list
for i in n:
    if i%2==0:
        print(i)
```

```
2
4
6
8
10
```

```
n = [1,2,3,4,5,6,7,8,9,10,11]           #Q19 program to print all odd
numbers in list
for i in n:
    if i%2==1:
         print(i)
```

```
1
3
5
7
9
11
```

```
n = [1,2,3,4,5,6,7,8,9]   #Q20 Program to print all even numbers in a
range
for i in range(1,10):
    if i%2==0:
       print(i)
```

```
2
4
6
8
```

```
n = [1,2,3,4,5,6,7,8,9]        #Q21   Program to print all odd numbers
in a range
for i in range(1,10):
    if i%2==1:                  #
       print(i)
```

```
1
3
5
7
9
```

```
list1 = [21,3,4,6,33,2,3,1,3,76]
#odd numbers
odd_count = len(list(filter(lambda a: (a%2 != 0) , list1)))
#Q22 Program to count Even and Odd numbers in a List
#even numbers
even_count = len(list(filter(lambda b: (b%2 == 0) , list1)))
print("Even numbers available in the list: ", even_count)
print("Odd numbers available in the list: ", odd_count)
```

```
Even numbers available in the list:  4
Odd numbers available in the list:  6
```

```
n = [1,2,3,4,5,6,7,8,9]
for i in n:           #Q23Program to print positive numbers in a list
    if i>0:
        print(i)
```

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

```
n = [1,2,3,4,5,6,7,8,9]
for i in n:           #Q24 Program to print negative numbers in a list
    if i<0:
        print(i)
```

```
n = [1,2,3,4,5,6,7,8,9]   #Q25 Program to print all positive numbers
in a range
for i in range(1,10):
    if i>0:
       print(i)
```

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

```
n = [1,2,3,4,5,6,7,8,9]   #Q26 Program to print all negative numbers
in a range
```

```
for i in range(1,12):
    if i<0:
       print(i)


list1 = [21,3,4,6,33,2,3,1,3,76]
#odd numbers
positive_count = len(list(filter(lambda a: (a>0) , list1)))
#Q27 Program to count positve and negative numbers in a List
#even numbers
negative_count = len(list(filter(lambda b: (b < 0) , list1)))
print("positive numbers available in the list: ", positive_count)
print("negativenumbers available in the list: ", negative_count)
```

```
positive numbers available in the list:  10
negativenumbers available in the list:  0
```

```
l1 = [1, 2, 3, 4, 3,1]
for i in l1:
    if l1.count(i)>1:            # Q28 remove multiple elements from a
list
        l1.remove(i)
print(l1)

# Way 1
l1 = [(1, 2), (), (3, 4), (), (5, 6)]
l1 = [t for t in l1 if t]
print(l1)
# Way 2
l1 = [(1, 2), (), (3, 4), (), (5, 6)]
l1 = list(filter(None, l1))                                  # Q29 remove
empty tuples from a list
print(l1)
# Way 3
l1 = [(1, 2), (), (3, 4), (), (5, 6)]
l1 = [t for t in l1 if t != ()]
print(l1)
```

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

```
l1=[1,1,2,2,3,4,4,3,5,5,6]
for i in l1:
    if l1.count(i)>1:
        print(i)
# Way 2
l1=[1,1,2,2,3,4,4,3,5,5,6]
l2=[]                                       # Q30 print duplicates from a
list of integers
for i in l1:
```

```
    if i not in l2:
        l2.append(i)
    else:
        print(i)
```

```
1
1
2
2
3
4
4
3
5
5
1
2
4
3
5
```