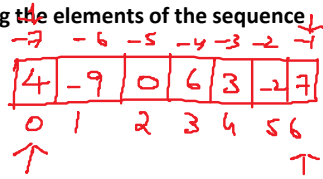# Day11_Loops(cont)

#Write a program on accessing the elements of the sequence
lst = [4,-9,0,6,3,-2,7]
Expected Output
4
-9
0
6
3
-2
7

lst = [4,-9,0,6,3,-2,7]

```
i=0
while i<len(lst):
    print(lst[i])
    i=i+1
```

4
-9
0
6
3
-2
7

```
i=-7
while i<=-1:
    print(lst[i])
    i=i+1
```
4
-9
0
6
3
-2
7
#elements in reverse
```
i=1
while i<=len(lst):
    print(lst[-i])
    i=i+1
```

#Getting the number of positives,negatives ans zeros in the list
```
lst = [4,-9,0,6,3,-2,7]
pos=neg=zer=0
i=0
while i<len(lst):
    if lst[i]>0:
        pos=pos+1
    elif lst[i]<0:
        neg=neg+1
    elif lst[i]==0:
        zer=zer+1
    i=i+1

print("Positive:",pos)
print("Negative:",neg)
print("Zero:",zer)
```
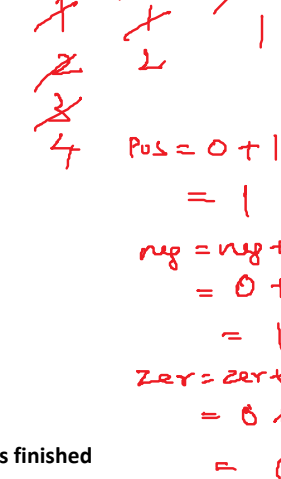
2. **for loop in python** : 1. A for loop in python is said to be a definite loop.
   2. In a for loop we provide a sequence and the loop automatically terminates as the sequence is finished
   3. A for loop is designed for iterating over a sequence of elements

3. A for loop is designed for iterating over a sequence of elements

Syntax :
```
for var in sequence:
    statements
```

Ex:
```
for ch in "Hello":
    print(ch)
```
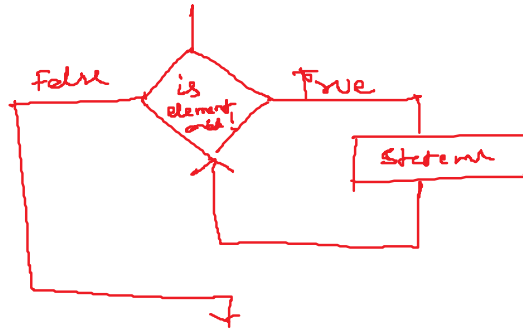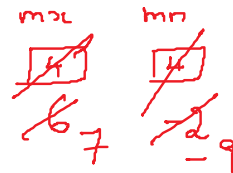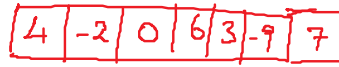
```
#For finding the no. of vowels in a string
cnt=0
for ch in "Hello! today is Thursday":
    if ch in ['a','e','i','o','u','A','E','I','O','U']:
        cnt=cnt+1

print("No. of vowels=",cnt)
#No. of vowels= 7
```

```
#Get the maximum and minimum element from a list
lst = [4,-2,0,6,3,-9,7]
mx,mn = lst[0],lst[0]

for val in lst:
    if val>mx:
        mx=val
    elif val<mn:
        mn=val

print("Minimum:",mn)
print("Maximum:",mx)
```
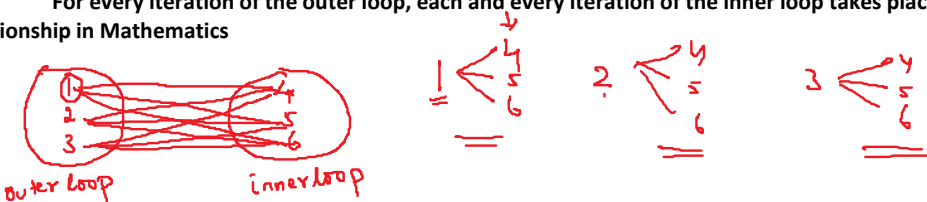
Minimum: -9
Maximum: 7

3. Nested loops : A loop inside another loop. Any kind of loop can be made available inside any other kind of a loop
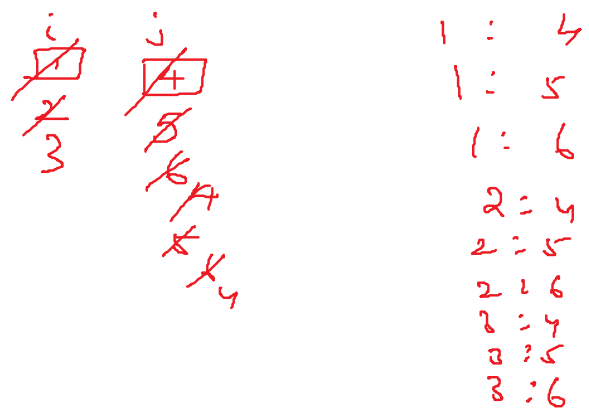
Syntax :
```
outerloop:
    innerloop
```

For every iteration of the outer loop, each and every iteration of the inner loop takes place, the execution is similar to a Many to to Many relationship in Mathematics
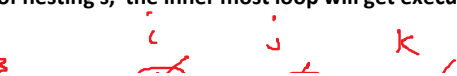
```
for i in range(1,4):
    for j in range(4,7):
        print(i,":",j)
```

```
1 : 4
1 : 5
1 : 6
2 : 4
2 : 5
2 : 6
3 : 4
3 : 5
3 : 6
```

We can have any number of nesting's, the inner most loop will get executed completely then the control will be given to the outer loop

```
for i in range(1,3):
```
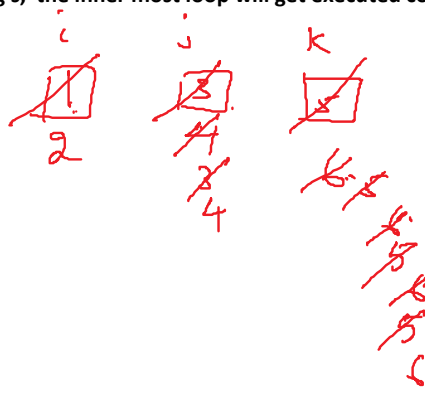
We can have any number of nesting's, the inner most loop will get executed completely then the control will be given to the outer loop

```
for i in range(1,3):
    for j in range(3,5):
        for k in range(5,7):
            print(i,":",j,":",k)
```

1 : 3 : 5
1 : 3 : 6
1 : 4 : 5
1 : 4 : 6
2 : 3 : 5
2 : 3 : 6
2 : 4 : 5
2 : 4 : 6