Stack implementation using list

#last element added at the end - (i!=0)

```
#last element added is at end-(i!=0)
def push(stack,x):
    stack.append(x)
    print(f"{x} is added and updated stack ={stack}")

def pop(stack):
   if(stack==[]):
       print("stack empty")
   else:
       print(f"{stack.pop()} is del   and updated stack ={stack}  ")
stack=[]
push(stack, 10)
push(stack,20)
pop(stack)
```

Stack Last element added at i=0

```
#last element at index i=0
def push(stack,x):
    stack.insert(0,x)
    print(f"{x} is added and updated stack ={stack}")

def pop(stack):
   if(stack==[]):
       print("stack empty")
   else:
       print(f"{stack.pop(0)} is del   and updated stack ={stack}  ")

stack=[]
push(stack, 10)
push(stack,20)
push(stack,30)
pop(stack)
```

Queue implementation

```
#In queue rear end for adding(enqueue) & front end for removing(dequeue).
```

```
#rear end at index 0.(adding at i=0 )
def enqueue(q1,x):
    q1.insert(0,x)
    print(f"{x} is added and updated queue ={q1}")

#front end at end of list(removing from last)
def dequeue(q1):
 if(q1==[]):
       print("queue empty")
 else:
       print(f"{q1.pop()} del and updated queue ={q1}")
q1=[]
enqueue(q1, 10)
enqueue(q1,20)
enqueue(q1,30)
dequeue(q1)
```

```
#rear end at end of the list(adding at end)
def enqueue(q2,y):
   q2.append(y)
   print(f"{y} is added and updated queue ={q2}")

#front end at i=0(removing from i=0)
def dequeue(q2):
 if(q2==[]):
       print("queue empty")
 else:
    print(f"{q2.pop(0)} is del and updated queue= {q2}")

q2=[]
enqueue(q2,10)
enqueue(q2,20)
enqueue(q2,30)
dequeue(q2)
dequeue(q2)
```