

In [6]: # STACK

```
class Stack:
    def __init__(self, MAX):
        self.top=-1
        self.size=MAX
        self.arr=[None]*MAX # or [None for i in range(MAX):]
    def push(self,item):
        if (self.top==self.size-1): #Overflow Condition
            print("\n Stack is Full!")
        else:
            self.top=self.top+1
            self.arr[self.top]=item
    def pop(self):
        if self.top== -1: #Underflow Condition
            print("\n Stack is Empty!")
        else:
            x=self.arr[self.top]
            self.top=self.top-1
            return x
    def display(self):
        if self.top== -1:
            print("\n Stack is Empty!")
        else:
            print("\n Elements in the Stack are: ")
            for i in range(0,self.top+1):
                print(self.arr[i], end=" ")

if __name__=="__main__":
    obj=Stack(5)
    obj.display()
    obj.push(10)
    obj.push(20)
    obj.push(30)
    obj.display()
    print("\n The first deleted element is: ", obj.pop())
    print("\n The second deleted element is: ", obj.pop())
    obj.display()
    obj.push(40)
    obj.display()
```

Stack is Empty!

Elements in the Stack are:

10 20 30

The first deleted element is: 30

The second deleted element is: 20

Elements in the Stack are:

10

Elements in the Stack are:

10 40

In [1]: # LINEAR QUEUE

```
class Queue:
    def __init__(self, MAX):
        self.front=-1
        self.rear=-1
        self.size=MAX
        self.arr=[None]*MAX
    def enqueue(self,item):
        if self.rear==self.size-1:
            print("\n The Queue is Full!")
        elif self.front==self.rear==0:
            self.front=self.rear=0
            self.arr[self.rear]=item
        else:
            self.rear=self.rear+1
            self.arr[self.rear]=item
    def dequeue(self):
        if self.front==0:
            print("\n Queue is Empty!")
        elif (self.front==self.rear):
            x=self.arr[self.front]
            self.front=self.rear=-1
            return x
        else:
            x=self.arr[self.front]
            self.front=self.front+1
            return x
    def display(self):
        if (self.front==0):
            print("\n Queue is Empty!")
        else:
            print("\n Elements in the Queue are: ")
            for i in range(self.front,self.rear+1):
                print(self.arr[i], end=" ")

if __name__=="__main__":
    obj=Queue(5)
    obj.display()
    obj.enqueue(10)
    obj.enqueue(20)
    obj.enqueue(30)
    obj.display()
    print("\n The first deleted element is: ", obj.dequeue())
    print("\n The second deleted element is: ", obj.dequeue())
    print("\n The third deleted element is: ", obj.dequeue())
    obj.display()
    obj.enqueue(40)
    obj.display()
```

Queue is Empty!

Elements in the Queue are:

10 20 30

The first deleted element is: 10

The second deleted element is: 20

The third deleted element is: 30

Queue is Empty!

Elements in the Queue are:

40

In [23]: *# Drawback of Linear Queue:*

```
# In linear queue it is not possible to insert any more new elements  
# through the rear end, although sufficient memory spaces are available  
# from the front end.  
# this problem of linear queue can be overcome through the concept of circular queue
```

In []: *#Circular Queue (INCOMPLETE)*

```
class Queue:
    def __init__(self, MAX):
        self.front=-1
        self.rear=-1
        self.size=MAX
        self.arr=[None]*MAX
    def enqueue(self, item):
        if front=(rear+1)%MAX:
            print("\n Queue is Full!")
        elif self.front==self.rear==-1:
            self.front=self.rear=0
            self.arr[self.rear]=item
        else:
            (self.rear=self.rear+1)%MAX
            self.arr[self.rear]=item
    def dequeue(self):
        if self.front==-1:
            print("\n Queue is Empty!")
        elif (self.front==self.rear):
            x=self.arr[self.front]
            self.front=self.rear=-1
            return x
        else:
            x=self.arr[self.front]
            (self.front=self.front+1)%MAX
            return x
    def display(self):
        if (self.front==-1):
            print("\n Elements in the Queue are: ")
            for i in range(self.front,self.rear+1):
                print(self.arr[i], end=" ")
```