Queue

1] Queue Using List :

```
q=[]
q.append(10)
q.append(20)
q.append(30)

print(q)
print(q.pop())
print(q)
q.append(40)
print(q)
print(q.pop(0))
print(q)
print(len(q))
print(q[0])
print(q[-1])
```

OUTPUT :

[10, 20, 30]

30

[10, 20]

[10, 20, 40]

10

[20, 40]

2

20

40

2] Using Queue :

```
from _collections import deque

q=deque()
q.append(10)
q.append(20)
q.append(30)
print(q)
print(q.popleft())
q.append(40)
print(q.popleft())
print(len(q))
print(q[0])
print(q[-1])
```

OUTPUT :

deque([10, 20, 30])

10

20

2

30

40

3] Linked List Implementation of Queue :

```
class Node:
    def __init__(self,k):
        self.key=k
        self.next=None

class MyQueue:
    def __init__(self):
        self.front=None
        self.rear=None
        self.sz=0

    def size(self):
        return self.sz

    def isEmpty(self):
        return self.sz==0

    def getFront(self):
        return self.front.key

    def getRear(self):
        return self.rear.key

    def enque(self,x):
        temp=Node(x)
        if self.rear==None:
            self.front=temp
        else:
            self.rear.next=temp

        self.rear=temp
        self.sz=self.sz+1

    def deque(self):
        if self.front==None:
            return None
        else:
            res=self.front.key
            self.front=self.front.next
            if self.front==None:
                self.rear=None
            self.sz=self.sz-1
            return res
```

```
q=MyQueue()
q.enque(10)
print(q.getFront(),q.getRear())
q.enque(20)
print(q.getFront(),q.getRear())
q.enque(30)
print(q.getFront(),q.getRear())
q.deque()
print(q.getFront(),q.getRear())
```

OUTPUT :

10 10

10 20

10 30

20 30

4] Queue Implementation using Circular List :

```
class MyQueue:
    def __init__(self,c):
        self.l=[None]*c
        self.cap=c
        self.size=0
        self.front=0

    def getFront(self):
        if self.size==0:
            return None
        else:
            return self.l[self.front]

    def getRear(self):
        if self.size==0:
            return None
        else:
            rear=(self.front+self.size-1)%self.cap
            return self.l[rear]

    def enque(self,x):
        if self.size==self.cap:
            return

        else:
            rear=(self.front+self.size-1)%self.cap
            rear=(rear+1)%self.cap
            self.l[rear]=x

            self.size=self.size+1
    def deque(self):
        if self.size==0:
            return None
        else:
            res=self.l[self.front]
            self.front=(self.front+1)%self.cap
            self.size=self.size-1

            return res

# Main
q=MyQueue(4)
q.enque(10)
print(q.getFront(),q.getRear())
```

```
q.enque(20)
print(q.getFront(),q.getRear())
q.enque(30)
print(q.getFront(),q.getRear())
q.enque(40)
print(q.getFront(),q.getRear())
q.deque()
print(q.getFront(),q.getRear())
q.deque()
print(q.getFront(),q.getRear())
q.enque(50)
print(q.getFront(),q.getRear())
```

OUTPUT :

10 10

10 20

10 30

10 40

20 40

30 40

30 50