LAB#9

Example: Write a program to create a queue using singly linked list.

Solution:

```
class Node:
    def __init__(self,item=None,next=None):
        self.item=item
        self.next=next
class Queue:
    def __init__(self):
        self.front=None
        self.rear=None
        self.itemcount=0
    def is_empty(self):
        return self.itemcount==0
    def enqueue(self,data):
        n=Node(data)
        if self.is_empty():
            self.front=n

        else:
            self.rear.next=n
        self.rear=n
        self.itemcount+=1
    def dequeue(self):
        if self.is_empty():
```

```
            print('Queue is empty')
        elif self.front==self.rear:
            self.front=None
            self.rear=None
        else:
            self.front=self.front.next
        self.itemcount-=1
    def getfront(self):
        if self.is_empty():
            print('Queue is empty')
        else:
            return self.front.item
    def getrear(self):
        if self.is_empty():
            print('Queue is empty')
        else:
            return self.rear.item
    def size(self):
        return self.itemcount
```

```
42
43      def print_queue(self):
44          if self.is_empty():
45              print('Queue is empty')
46          else:
47              print('Queue:')
48              temp=self.front
49              while temp is not None:
50
51                  print(temp.item,end=' ')
52                  temp=temp.next
53
54  q1=Queue()
55  q1.enqueue(10)
56  q1.enqueue(20)
57  q1.enqueue(30)
58  q1.print_queue()
59
60  print('size:',q1.size())
61  print('front:',q1.getfront())
```

```
62  print('rear:',q1.getrear())
63  q1.dequeue()
64  q1.print_queue()
```

Result:

```
Queue:
10 20 30 size: 3
front: 10
rear: 30
Queue:
20 30
```

Class Assignment

Q.1: Repeat the above program lab#9 example#1 by using the list data type.

Q.2: Repeat the above program lab#9 example#1 by using array.