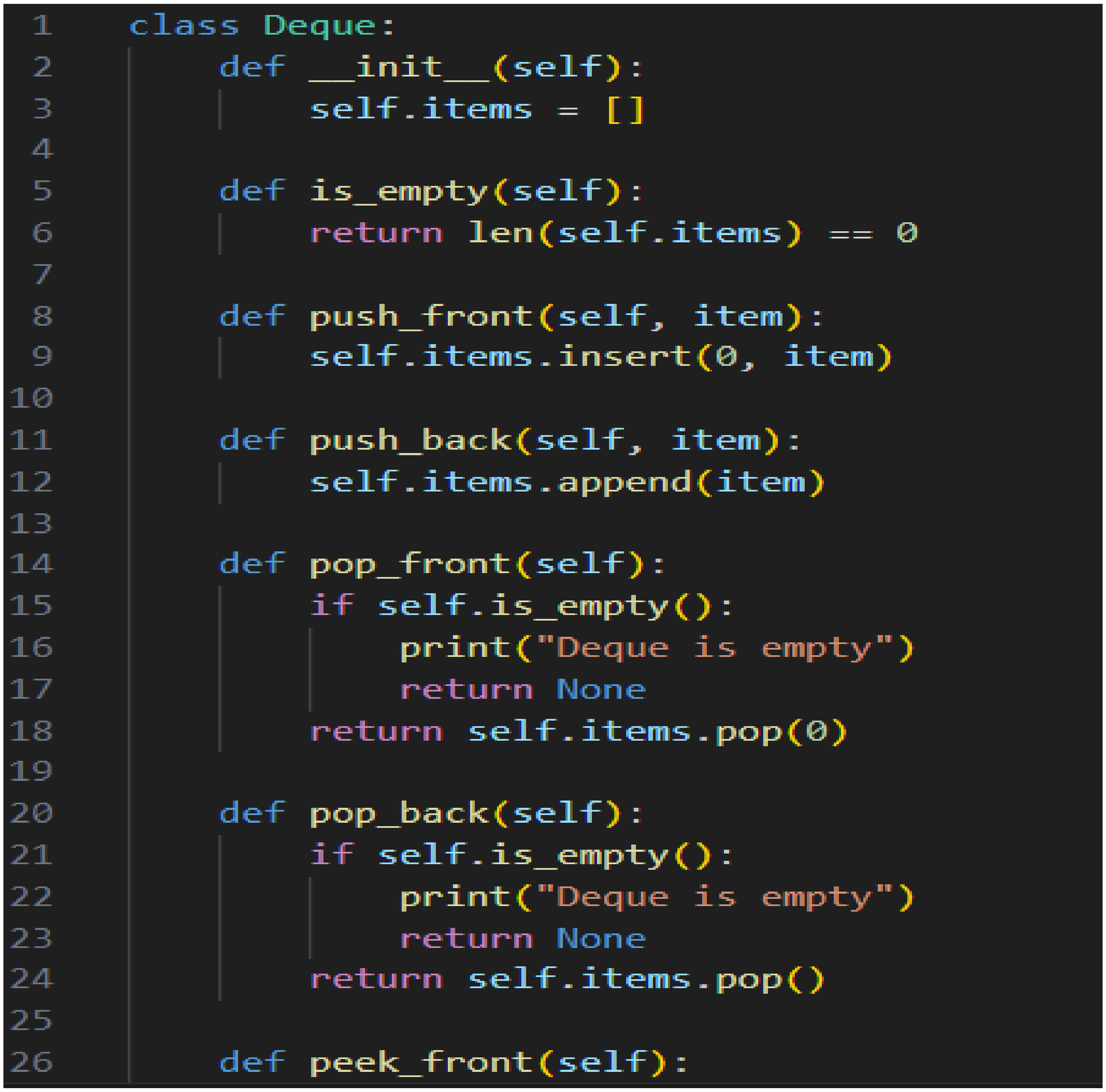
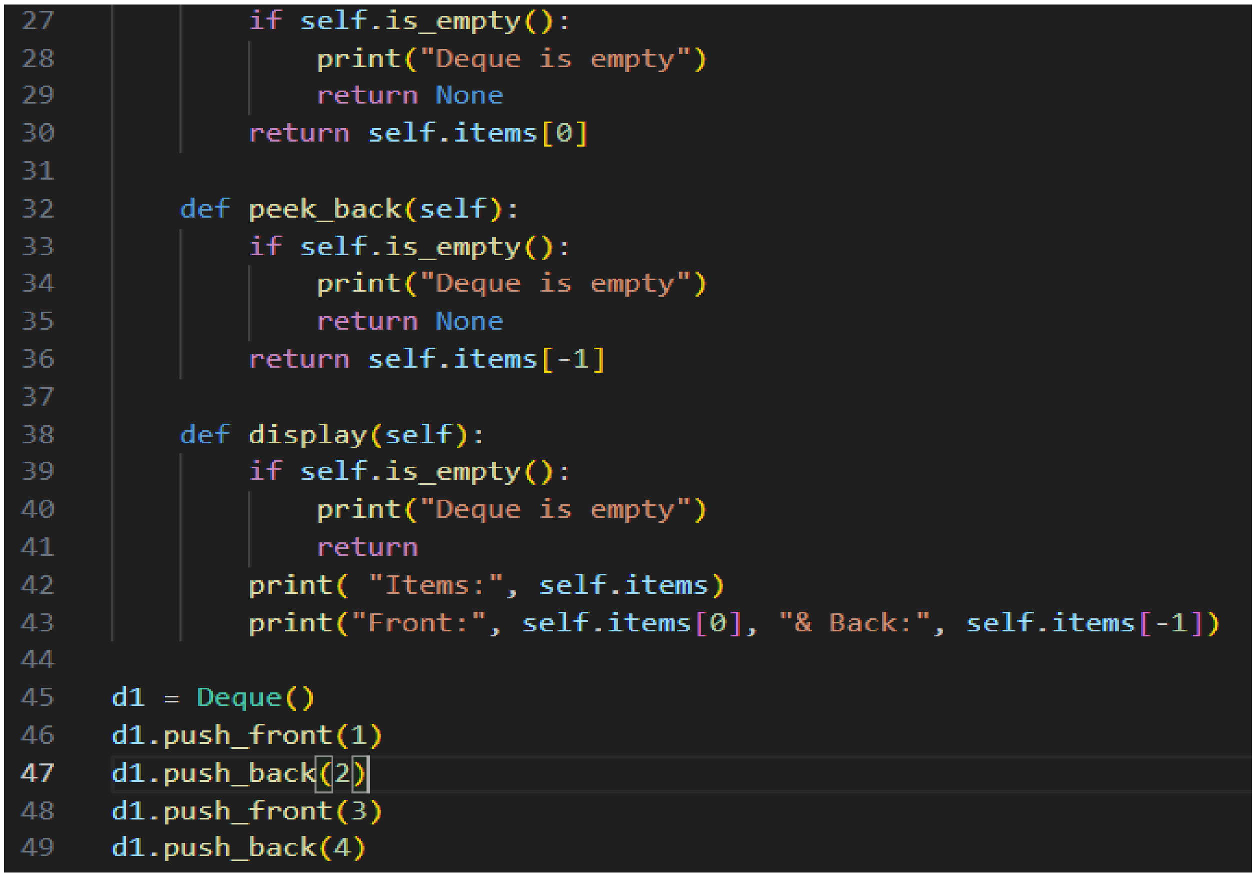
LAB#10

Example#1: Write a program to create a deque (double ended queue) by using list.

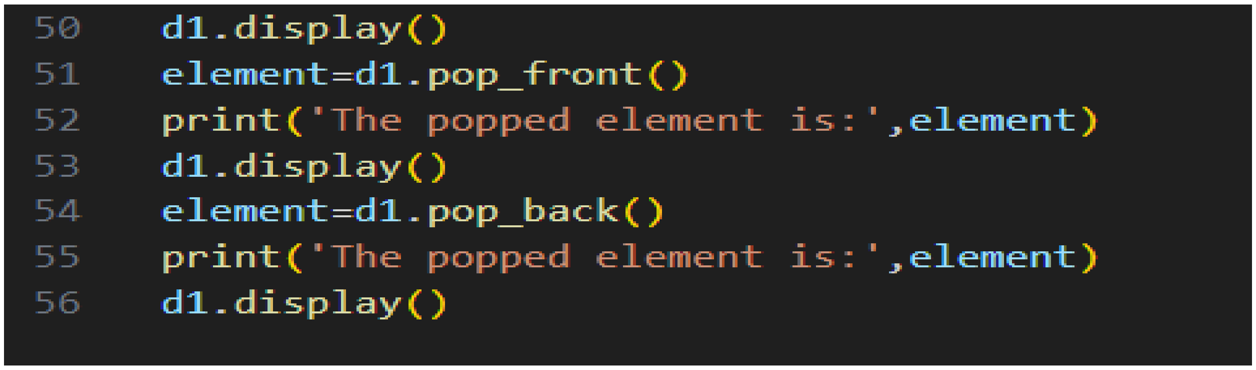
Solution:



```python
class Deque:
    def __init__(self):
        self.items = []

    def is_empty(self):
        return len(self.items) == 0

    def push_front(self, item):
        self.items.insert(0, item)

    def push_back(self, item):
        self.items.append(item)

    def pop_front(self):
        if self.is_empty():
            print("Deque is empty")
            return None
        return self.items.pop(0)

    def pop_back(self):
        if self.is_empty():
            print("Deque is empty")
            return None
        return self.items.pop()

    def peek_front(self):
```

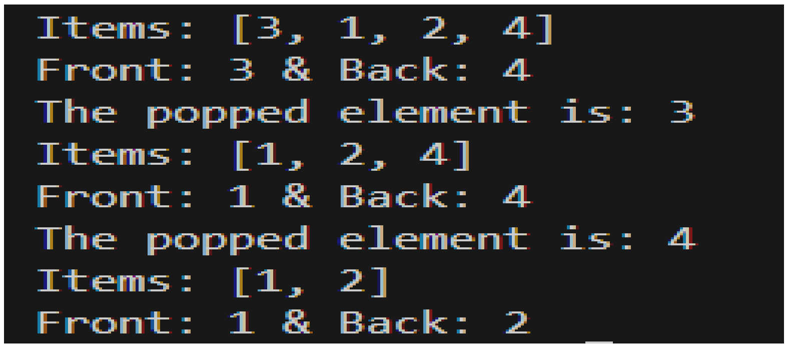


```python
        if self.is_empty():
            print("Deque is empty")
            return None
        return self.items[0]

    def peek_back(self):
        if self.is_empty():
            print("Deque is empty")
            return None
        return self.items[-1]

    def display(self):
        if self.is_empty():
            print("Deque is empty")
            return
        print( "Items:", self.items)
        print("Front:", self.items[0], "& Back:", self.items[-1])

d1 = Deque()
d1.push_front(1)
d1.push_back(2)
d1.push_front(3)
d1.push_back(4)
```



```python
d1.display()
element=d1.pop_front()
print('The popped element is:',element)
d1.display()
element=d1.pop_back()
print('The popped element is:',element)
d1.display()
```

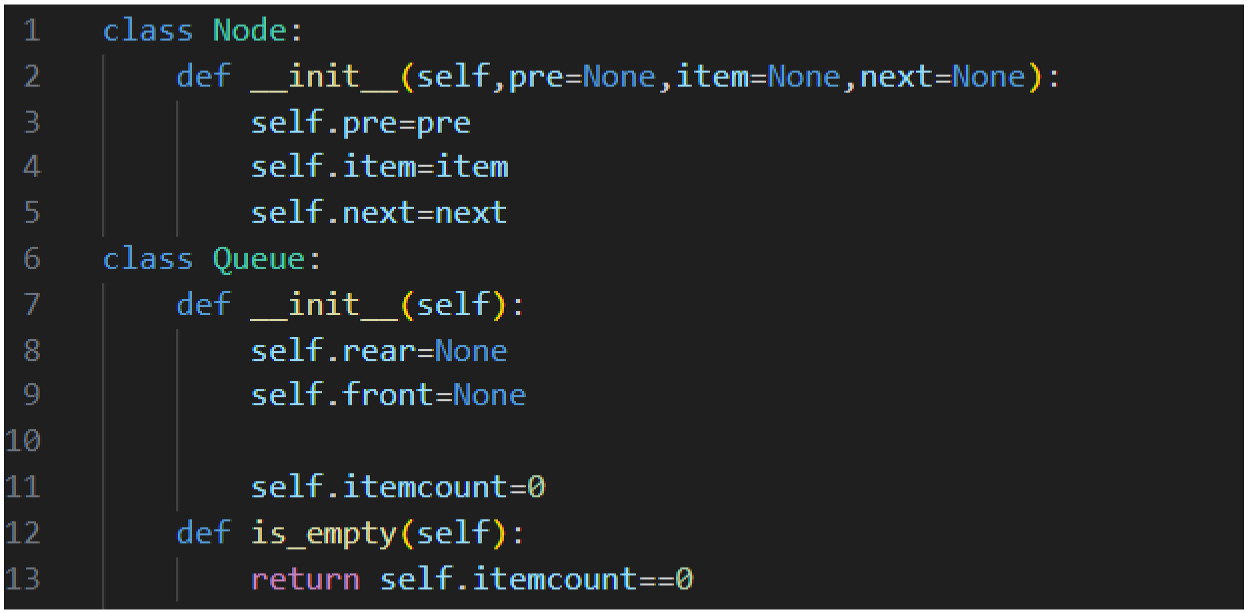
Result:



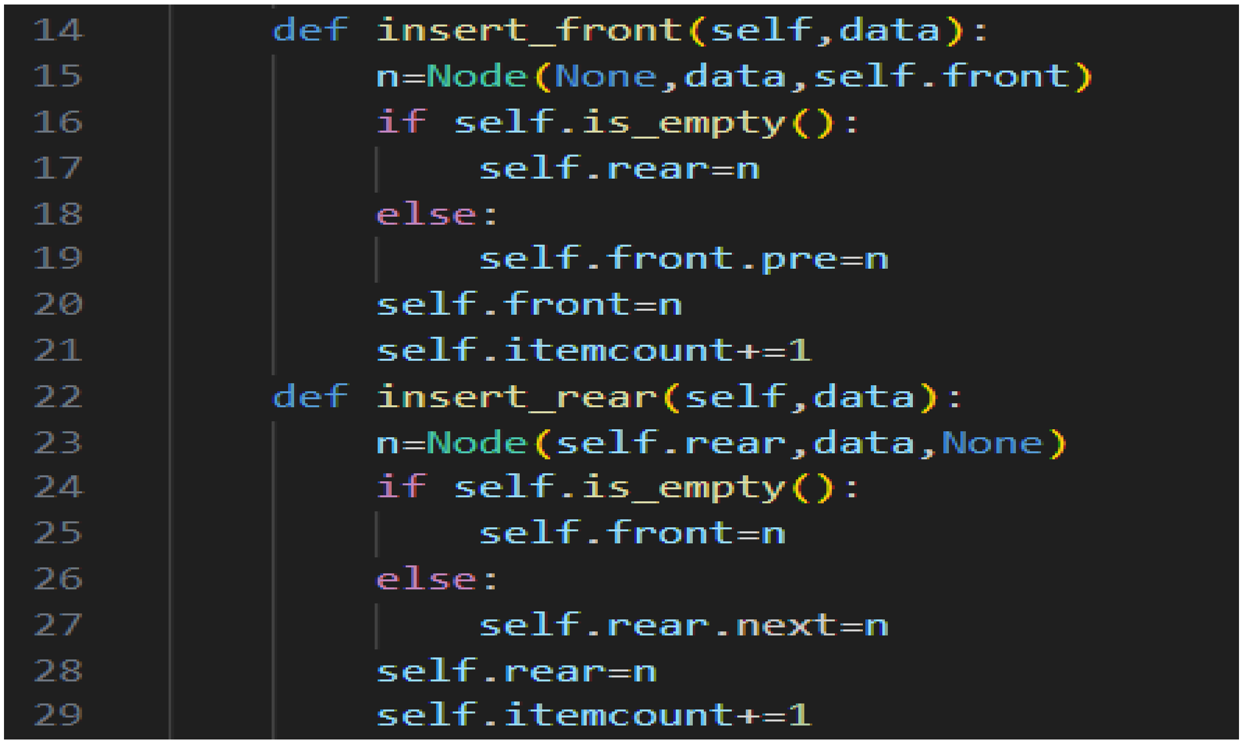
```
Items: [3, 1, 2, 4]
Front: 3 & Back: 4
The popped element is: 3
Items: [1, 2, 4]
Front: 1 & Back: 4
The popped element is: 4
Items: [1, 2]
Front: 1 & Back: 2
```

Example#2: Write a program to create a deque (double ended queue) by using doubly linked list.

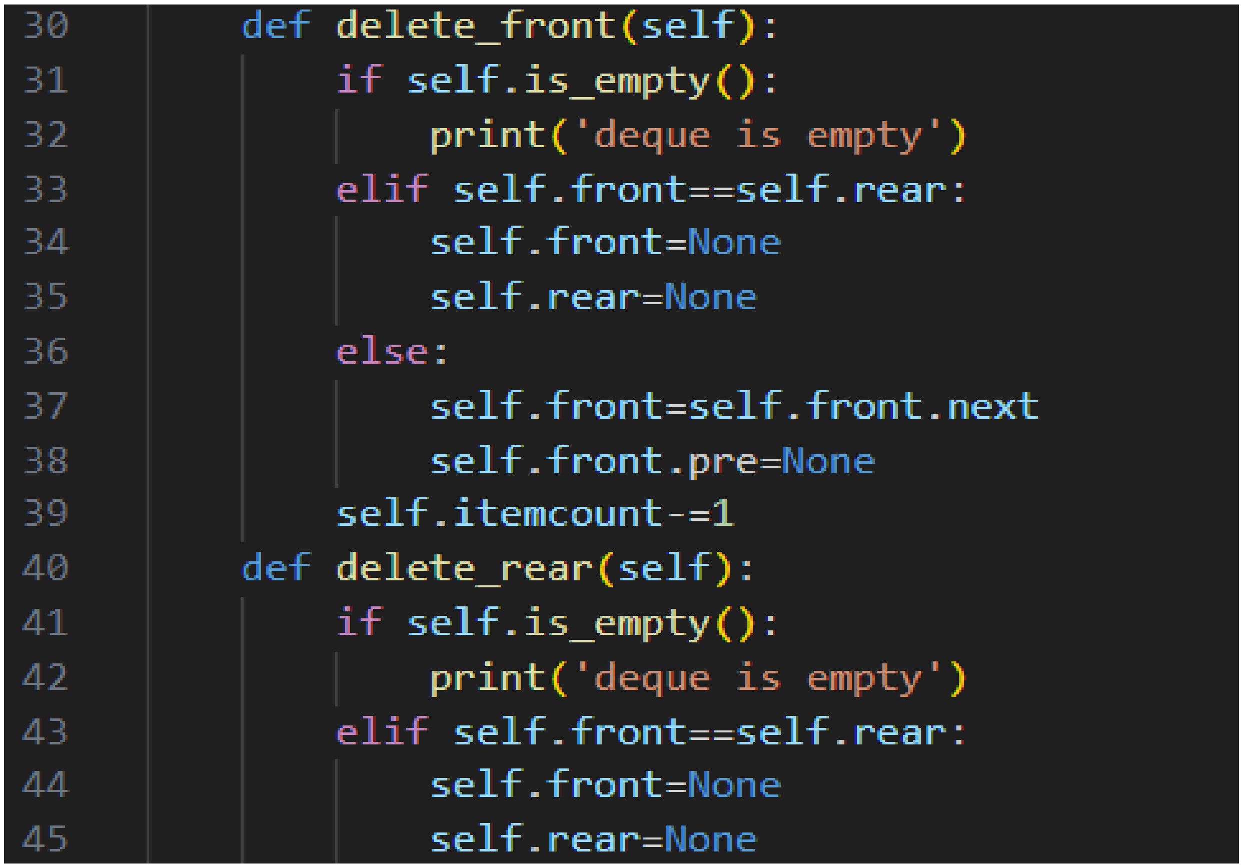
Solution:



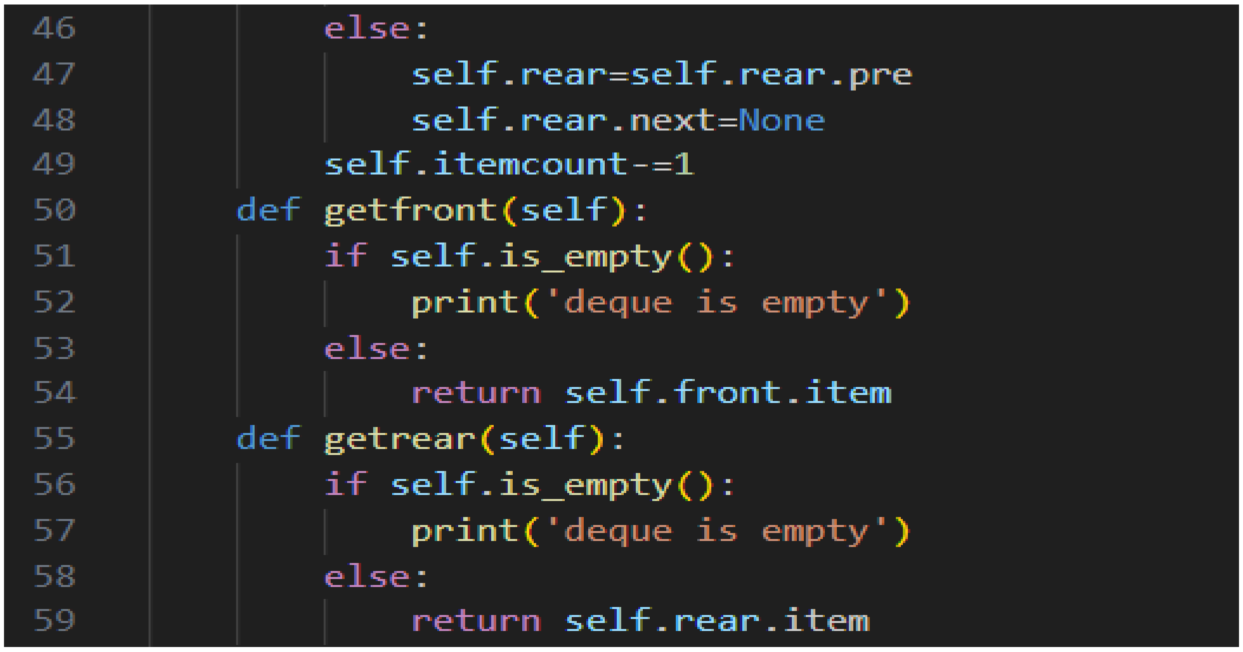
```python
class Node:
    def __init__(self,pre=None,item=None,next=None):
        self.pre=pre
        self.item=item
        self.next=next
class Queue:
    def __init__(self):
        self.rear=None
        self.front=None

        self.itemcount=0
    def is_empty(self):
        return self.itemcount==0
```



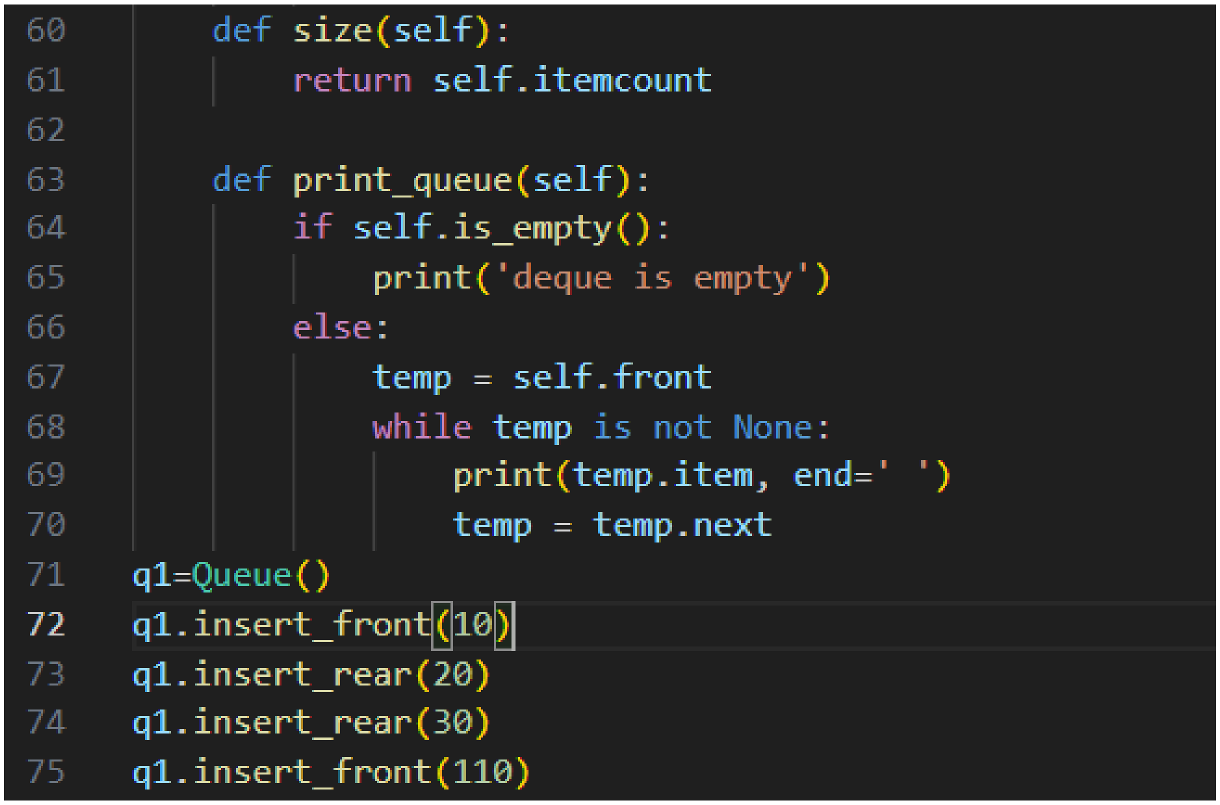
```python
    def insert_front(self,data):
        n=Node(None,data,self.front)
        if self.is_empty():
            self.rear=n
        else:
            self.front.pre=n
        self.front=n
        self.itemcount+=1
    def insert_rear(self,data):
        n=Node(self.rear,data,None)
        if self.is_empty():
            self.front=n
        else:
            self.rear.next=n
        self.rear=n
        self.itemcount+=1
```



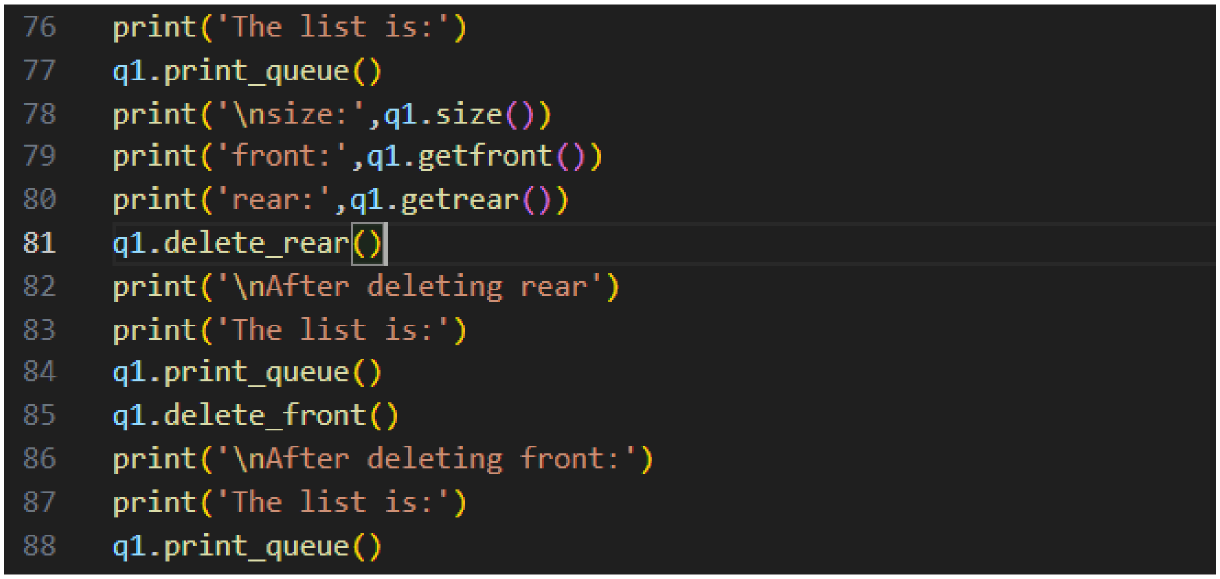
```python
    def delete_front(self):
        if self.is_empty():
            print('deque is empty')
        elif self.front==self.rear:
            self.front=None
            self.rear=None
        else:
            self.front=self.front.next
            self.front.pre=None
        self.itemcount-=1
    def delete_rear(self):
        if self.is_empty():
            print('deque is empty')
        elif self.front==self.rear:
            self.front=None
            self.rear=None
```



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

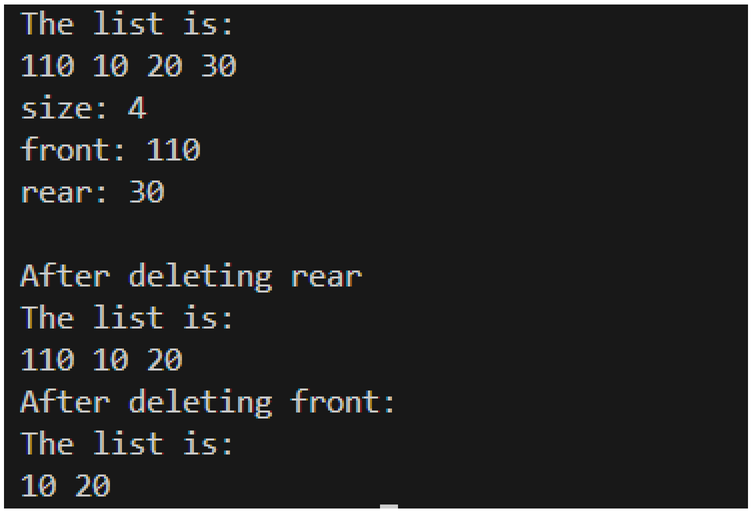


```
    def size(self):
        return self.itemcount

    def print_queue(self):
        if self.is_empty():
            print('deque is empty')
        else:
            temp = self.front
            while temp is not None:
                print(temp.item, end=' ')
                temp = temp.next
q1=Queue()
q1.insert_front(10)
q1.insert_rear(20)
q1.insert_rear(30)
q1.insert_front(110)
```



```
print('The list is:')
q1.print_queue()
print('\nsize:',q1.size())
print('front:',q1.getfront())
print('rear:',q1.getrear())
q1.delete_rear()
print('\nAfter deleting rear')
print('The list is:')
q1.print_queue()
q1.delete_front()
print('\nAfter deleting front:')
print('The list is:')
q1.print_queue()
```

Result:



```
The list is:
110 10 20 30
size: 4
front: 110
rear: 30

After deleting rear
The list is:
110 10 20
After deleting front:
The list is:
10 20
```

Class Assignment

Q.1: Modify the above program example#1 lab#10 by using array to create the deque.

Q.2: Modify the above program example#2 lab#10 by using singly linked list to create the deque.

Q.3: Modify the print_queue() function in the example#2 lab#10 by printing the elements in the list from rear to front.