EXERCICE:4.1

class Queue:

def \_\_init\_\_(self):

self.queue=[]

def is\_empty(self):

return len(self.queue)==0

def enqueue(self,item):

self.queue.append(item)

def dequeue(self):

if self.is\_empty():

raise IndexError("Queue is empty")

return self.queue.pop(0)

def peek(self):

if self.is\_empty():

raise IndexError("Queue is empty")

return self.queue[0]

def size(self):

return len(self.queue)

q=Queue()

q.enqueue(10)

q.enqueue(20)

q.enqueue(30)

print("Queue after enqueuing element:",q.queue)

print("Dequeue element:",q.dequeue())

print("Queue after dequeuing an element:",q.queue)

print("Front element:",q.peek())

print("Queue Size:",q.size())

OUTPUT:

Queue after enqueuing element: [10, 20, 30]

Dequeue element: 10

Queue after dequeuing an element: [20, 30]

Front element: 20

Queue Size: 2

EXRCISE:4.2

class Node:

    def \_\_init\_\_(self,data):

        self.data=data

        self.next=None

class Queue:

    def \_\_init\_\_(self):

        self.front=None

        self.rear=None

    def is\_empty(self):

        return self.front is None

    def enqueue(self,data):

        new\_node=Node(data)

        if self.rear is None:

            self.front=self.rear=new\_node

            return

        self.rear.next=new\_node

        self.rear=new\_node

    def dequeue(self):

        if self.is\_empty():

            return None

        data=self.front.data

        self.front=self.front.next

        if self.front is None:

            self.rear=None

        return data

    def peek(self):

        if self.is\_empty():

            return None

        return self.front.data

    def size(self):

        count=0

        current=self.front

        while current:

            count+=1

            current=current.next

        return count

q=Queue()

q.enqueue(10)

q.enqueue(20)

q.enqueue(30)

print("Queue element:",end=" ")

while not q.is\_empty():

    print(q.dequeue(),end=" ")

print("\n is the queue empty?",q.is\_empty())

OUTPUT

Queue element: 10 20 30

is the queue empty? True