LINKED LIST QUEUE WITH CLEAR

```
#include<iostream>
        using namespace std;

struct node{
        int data;
        node *next;
};
class queue{
        node *front, *rear;
        public:
                queue(){
                        front = rear = NULL;
                }
                void enqueue(int data){
                        node *newnode;
                        newnode=new node;
                        newnode->     data=data;
                        newnode->next=NULL;
                        if(front ==NULL){
                                front=rear=newnode;
                        }
                        else{
                                rear->  next=newnode;
                                rear=newnode;
                        }
                }
                void dequeue(){
```

```
        node *temp;
        if(front->next==NULL){
                cout<<"queue does not exist"<<endl;

        }
        else{
                temp=front;
                front=front->next;
        delete temp;
        }
}
void display(){
        if(front==NULL){
                cout<<"The queue is empty"<<endl;
        }
        else{
        node *temp;
        temp=front;
        while(temp!=NULL){
                cout<<temp->data<<endl;
                temp=temp->next;
        }
        cout<<endl;
}}
void clear(){
        node *temp;
        node *temp2;
        temp=front;
        while(temp!=NULL){
```

```
                temp2=temp;
                temp=temp->next;
        temp2->next=NULL;
        temp2->data=NULL;
        front=NULL;
            }
            cout<<"The queue has been deleted"<<endl;
        }
};

int main(){
    queue q;
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.dequeue();
    q.dequeue();
    q.dequeue();
```

```
        q.display();
        q.clear();
        q.enqueue(10);
        q.display();
}
```

LINKED List QUEUE WITH COUNT

```
#include<iostream>
        using namespace std;

struct node{
        int data;
        node *next;
};
class queue{
        node *front, *rear;
        public:
                queue(){
                        front = rear = NULL;
                }
                void enqueue(int data){
                        node *newnode;
                        newnode=new node;
                        newnode->     data=data;
```

```
        newnode->next=NULL;
        if(front ==NULL){
                front=rear=newnode;
        }
        else{
                rear->  next=newnode;
                rear=newnode;
        }
}
void dequeue(){
        node *temp;
        if(front==NULL){
                cout<<"queue does not exist"<<endl;

        }
        else{
                temp=front;
                front=front->next;
        delete temp;
        }
}
void display(){
        if(front==NULL){
                cout<<"The queue is empty"<<endl;
        }
        else{
        node *temp;
        temp=front;
        while(temp!=NULL){
```

```
                cout<<temp->data<<endl;
                temp=temp->next;
            }
            cout<<endl;
        }}
        int count(){
        node *temp;
        int co=0;
        temp=front;
        while(temp!=NULL){
            co++;
            temp=temp->next;

        }
        cout<<"Count : "<<co<<endl;
        return co;
        }

};

int main(){
    queue q;
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
    q.enqueue(10);
```

```
        q.enqueue(10);
        q.enqueue(10);
        q.enqueue(10);
        q.enqueue(10);
        q.enqueue(10);
        q.enqueue(10);
        q.enqueue(10);
        q.count();
        q.dequeue();
        q.dequeue();
        q.dequeue();
        q.display();
        q.count();
}
```