Experiment E-31

Code:

```
#include <iostream>

using namespace std;

int const M=10;

class deque{
    private:
    int front ,rear;
    int arr[M];
    public:
    deque(){
        front=rear=-1;
    }
    void enquefront(int n);
    void enquerear(int n);
    void dequefront();
    void dequerear();
    void display();


};


void deque::enquefront(int n){
    if(front==0){
        cout<<"\ncannot insert at front"<<endl;
```

```
}
else if(front==-1){
    front=rear=0;
    arr[rear]=n;
}
else{
    front–;
    arr[front]=n;
    }



}

void deque::enquerear(int n){
    if(rear==M-1){
        cout<<"\ncannot insert at rear"<<endl;
    }
    else if(rear==-1){
        front=rear=0;
        arr[rear]=n;
    }
    else{
        rear++;
        arr[rear]=n;
    }
}
```

```
void deque:: dequefront(){
    if(front==-1){
        cout<<"\nqueue is empty"<<endl;
    }
    else if(front==rear){
        int temp=arr[front];
        front=rear=-1;
        cout<<"\ndeleted element:"<<temp<<endl;
    }
    else{
        int temp=arr[front];
        front++;
        cout<<"\ndeleted element:"<<temp<<endl;
    }
}

 void deque:: dequerear(){
    if(rear==-1){
        cout<<"\nqueue is empty"<<endl;
    }
    else if(front==rear){
        int temp=arr[rear];
        front=rear=-1;
        cout<<"\ndeleted element:"<<temp<<endl;
    }
    else{
        int temp=arr[rear];
```

```
        rear–;
        cout<<"\ndeleted element:"<<temp<<endl;
    }
}

void deque::display(){
    if(front!=-1){
        for (int i=front;i<=rear;i++){
            cout<<arr[i]<<" ";
        }
        cout<<"\n";
    }
    else{
        cout<<"\ndeque is empty"<<endl;
    }
}

int main(){
    deque q;
    int n,x;
    do{
        cout<<"\n1.insert at front \n2.insert at rear \n3.delete at front \n4.delete at end
\n5.display \n6.exit\n";
        cout <<"Enter your choice: ";
        cin>>n;
        switch(n){
            case 1:
```

```
        cout<<"\nenter the element:";
        cin>>x;
        q.enquefront(x);
        break;
    case 2:
        cout<<"\nenter the element:";
        cin>>x;
        q.enquerear(x);
        break;
    case 3:
        q.dequefront();
        break;
    case 4:
        q.dequerear();
        break;
    case 5:
        cout<<"\n Queue contains:";
        q.display();
        break;
            case 6:
                cout<<"\nprocess ended...";
                break ;
            }
        }while(n!=6);
        return 0;
}
```

Output:

1.insert at front

2.insert at rear

3.delete at front

4.delete at end

5.display

6.exit

Enter your choice: 1

enter the element:12

1.insert at front

2.insert at rear

3.delete at front

4.delete at end

5.display

6.exit

Enter your choice: 2

enter the element:24

1.insert at front

2.insert at rear

3.delete at front

4.delete at end

5.display

6.exit

Enter your choice: 2

enter the element:36

1.insert at front

2.insert at rear

3.delete at front

4.delete at end

5.display

6.exit

Enter your choice: 5

Queue contains:12 24 36

1.insert at front

2.insert at rear

3.delete at front

4.delete at end

5.display

6.exit

Enter your choice: 3

deleted element:12

1.insert at front

2.insert at rear

3.delete at front

4.delete at end

5.display

6.exit

Enter your choice: 4

deleted element:36

1.insert at front

2.insert at rear

3.delete at front

4.delete at end

5.display

6.exit

Enter your choice: 5

Queue contains:24

1.insert at front

2.insert at rear

3.delete at front

4.delete at end

5.display

6.exit

Enter your choice: 6

process ended...