```
#include<iostream>
#define MAX 20
using namespace std;

class dequeue
{
	public:
		int front,rear,a[MAX];
		dequeue()
		{
			front=rear=-1;
		}
		void insert_beg(int ele);
		void insert_end(int ele);
		void delete_end();
		void delete_beg();
		void display();
};

void dequeue::insert_end(int ele)
{
	if(rear>=MAX-1)
	{
		cout<<"\nInsertion not possible!!";
	}
	else
	{
		if (front==-1)
		{
			front+=1;
```

```
                rear+=1;
        }
        else
        {
                rear+=1;
        }
        a[rear]=ele;
        cout<<"\nInserted element is:"<<a[rear];
    }
}

void dequeue::insert_beg(int ele)
{
    if (front==-1)
    {
        front=0;
        a[++rear]=ele;
        cout<<"\nInserted element is:"<<ele;
    }
    else if (front!=0)
    {
        a[--front]=ele;
        cout<<"\nInserted element is:"<<ele;
    }
    else
    {
        cout<<"\nInsertion not possible!!";
    }
}
```

```
void dequeue::delete_beg()
{
	if (front==-1)
	{
		cout<<"\nDeletion not possible!!";
		return;
	}
	else
	{
		cout<<"\nInserted element is:"<<a[front];
		if(front==rear)
		{
			front=rear=-1;
			return;
		}
		else
		{
			front+=1;
		}
	}
}

void dequeue::delete_end()
{
	if (front==-1)
	{
		cout<<"\nDeletion not possible!!";
		return;
	}
	else
```

```
	{
		cout<<"\nInserted element is:"<<a[rear];
		if(front==rear)
		{
			front=rear=-1;
			return;
		}
		else
		{
			rear+=1;
		}
	}
}

void dequeue::display()
{
	if (front==-1)
	{
		cout<<"\nDequeu is empty.";
	}
	else
	{
		for (int i=front;i<=rear;i++)
		{
			cout<<a[i]<<" ";
		}
	}
}

int main()
```

```
{
        dequeue q,p;
        int val,ch,ch1;

        do
        {
                cout<<"\n\tShopping List";
                cout<<"\n1.Input restricted Queue\n2.Output restricted Queue\n3.Exit\nEnter
choice:";
                cin>>ch;
                switch(ch)
                {
                        case 1:
                        {
                                do
                                {
                                        cout<<"\n1.Insert\n2.Delete at front\n3.Delete at
rear\n4.Display\n5.Exit\nEnter choice:";
                                        cin>>ch1;
                                        switch(ch1)
                                        {
                                                case 1:
                                                {
                                                        cout<<"Enter Element: ";
                                                        cin>>val;
                                                        q.insert_end(val);
                                                        break;
                                                }
                                                case 2:
                                                {
                                                        q.delete_beg();
```

```
                    break;
                }
                case 3:
                {
                    q.delete_beg();
                    break;
                }
                case 4:
                {
                    q.display();
                    break;
                }
            }
        }while(ch1!=5);
        break;
    }
    case 2:
    {
        do
        {
            cout<<"\n1.Insert at begin\n2.Insert at end\n3.Delete\n4.Display\n5.Exit\nEnter choice:";
            cin>>ch1;
            switch(ch1)
            {
                case 1:
                {
                    cout<<"Enter Element: ";
                    cin>>val;
                    p.insert_beg(val);
                    break;
```

```
                                        }
                                        case 2:
                                        {
                                                cout<<"Enter Element: ";
                                                cin>>val;
                                                p.insert_end(val);
                                                break;
                                        }
                                        case 3:
                                        {
                                                p.delete_beg();
                                                break;
                                        }
                                        case 4:
                                        {
                                                p.display();
                                                break;
                                        }
                                }
                        }while(ch1!=5);
                        break;
                }
        }
    }while(ch!=3);
    return 0;
}
```