```
/*******************************************************************************

                    Double Ended Queue Implementation using C++

Some of the major function of Double Ended Queue are:
     1. Insertion from rear
     2. Insertion from front
     3. Deletion from rear
     4. Deletion from front
     5. Display

*******************************************************************************/
#include <iostream>
#define SIZE 10

class Deque{
   private:
   int front,rear;
   int arr[SIZE];
   public:
   Deque(){
      front=-1;
      rear=-1;
   }
   void addfromrear(int data);
   void addfromfront(int data);
   void deletefromfront();
   void deletefromrear();
   void Display();
};
using namespace std;
void Deque::addfromrear(int data){
   if((rear==SIZE-1 && front ==0)||(front==rear+1)){
      cout<<"Double ended queue is full"<<endl;
   }
   else if(front==-1 && rear ==-1){
      front++;
      rear++;
      arr[rear]=data;
   }
   else if(rear==SIZE-1){
      rear=0;
      arr[rear]=data;
   }
```

```
    else{
        rear++;
        arr[rear]=data;
    }
}
void Deque::addfromfront(int data){
    if((rear==SIZE-1 && front ==0)||(front==rear+1)){
        cout<<"Double ended queue is full"<<endl;
    }
    else if(front==-1 && rear ==-1){
        front++;
        rear++;
        arr[front]=data;
    }
    else if(front==0){
        front=SIZE-1;
        arr[front]=data;
    }
    else{
        front--;
        arr[front]=data;
    }
}
void Deque::deletefromfront(){
    if(front==-1 && rear==-1){
        cout<<"Deque is full"<<endl;
    }
    else if(front==rear){
        cout<<"The deleted element is "<<arr[front]<<endl;
        front=-1;
        rear=-1;
    }
    else if(front==SIZE-1){
        cout<<"The deleted element is "<<arr[front]<<endl;
        front=0;
    }
    else{
        cout<<"The deleted element is "<<arr[front]<<endl;
        front++;

    }
}
void Deque::deletefromrear(){
        if(front==-1 && rear==-1){
```

```
        cout<<"Deque is full"<<endl;
    }
    else if(front==rear){
        cout<<"The deleted element is "<<arr[rear]<<endl;
        front=-1;
        rear=-1;
    }
    else if(rear==0){
        cout<<"The deleted element is "<<arr[rear]<<endl;
        rear=SIZE-1;
    }
    else{
        cout<<"The deleted element is "<<arr[rear]<<endl;
        rear--;;

    }
}
void Deque::Display(){
    int i=front;
    cout<<"\nThe elements are:\n"<<endl;
    while(i!=rear){
        cout<<arr[i]<<" ";
        i=(i+1)%SIZE;
    }
    cout<<arr[rear]<<endl;
}
int main()
{
    Deque dq;
    int choice,d;
    while(1){
    cout<<"\n----------MENU---------"<<endl;
    cout<<"1. Insert from rear"<<endl;
    cout<<"2. Insert from front"<<endl;
    cout<<"3. Delete from rear"<<endl;
    cout<<"4. Delete from front"<<endl;
    cout<<"5. Display"<<endl;
    cout<<"6. Exit"<<endl;
    cout<<endl<<endl;
    cout<<"Enter a choice:"<<endl;
    cin>>choice;
    switch(choice){
        case 1: cout<<"Enter dataa to be inserted"<<endl;
        cin>>d;
```

```
        dq.addfromrear(d);
        break;
        case 2: cout<<"Enter dataa to be inserted"<<endl;
        cin>>d;
        dq.addfromfront(d);
        break;
        case 3: dq.deletefromrear();
        break;
        case 4: dq.deletefromfront();
        break;
        case 5: dq.Display();
        break;
        default: cout<<"Enter values fron 1-5"<<endl;
    }
    }


    return 0;
}
```