```
#include<iostream>
#define SIZE 10
using namespace std;
class dqueue{
    int queue1[SIZE];
    int front,rear;
public:
    dqueue(){
        front=-1;
        rear=-1;
    }
    int deQueueFront();
    int deQueueRear();
    void enQueueRear();
    void enQueueFront();
    void display_front();
    void display_rear();
};
void dqueue::enQueueRear()
{
    int value;
    if(rear==SIZE-1){
        cout<<"\nQueue is full, Insertion is not possible!!!";
        return;
    }
    else{
        if(front==-1){
            front=0;
            cout<<"\nEnter the value to be inserted:";
            cin>>value;
            rear=rear+1;
            queue1 [rear] = value;
        }
        else{
            cout<<"\nEnter the value to be inserted:";
            cin>>value;
            rear=rear+1;
            queue1[rear]=value;
        }
    }
}
void dqueue::enQueueFront(){
    int value;
    if(front==0){
        cout<<"\n Insertion is not possible, element exists at index 0!!!";
        return;
    }
    else{
        cout<<"\nEnter the value to be inserted:";
        cin>>value;
        if(front==-1){
            front=rear=0;
        }
        else{
            front--;
        }
        queue1[front]= value;
    }
}
int dqueue::deQueueRear(){
    int deleted_element;
    // deleted queue] [rear];
    if(front == -1)
    {
        cout<<"\nQueue is Empty!!! Deletion is not possible!!!";
        return 0;
    }
    else if(front==rear){
        front=rear=-1;
    }
    else if(rear==0)
    {
        deleted_element = queue1[rear];
        rear=rear-1;
    }
    else{
        deleted_element = queue1[rear];
        rear=rear-1;
    }
    return deleted_element;
}
int dqueue::deQueueFront(){
    int deleted_element;
    if(front == -1){
        cout<<"\nQueue is Empty!!! Deletion is not possible!!!";
        return 0;
    }

    else if(front==rear)//only one element in Q
    {
        deleted_element=queue1[front];
        front=rear=-1;
    }
    else{
        deleted_element = queue1[front];
        front=front+1;
    }
    return deleted_element;
}
void dqueue::display_front(){
    int i;
    if(front == -1)
        cout<<"\nQueue is Empty!!! Display is not possible!!!";
    else{
        cout<<"\nThe Queue_front element is:";
        cout<<queue1[front]<<"\t";
    }
}
void dqueue::display_rear(){
    int i;
    if(front ==-1)
        cout<<"\nQueue is Empty!!! Display is not possible!!!";
    else{
        cout<<"\nThe Queue Rear element is:";
        cout<<queue1[rear]<<"\t";
    }
}
int main(){
    dqueue DQ;
    char ch;
    int choice1, value;
    cout<<"\n******* Double Ended Queue operations*******\n";
    do
    {
        cout<<"\n1.Insert at rear end \n";
        cout<<"2.Delete from rear end \n";
        cout<<"3.Delete from front end \n";
        cout<<"4.Insert at front end \n";
        cout<<"5.Display_front \n";
        cout<<"6.Display_rear \n";
        cout<<"\nEnter your choice: ";
        cin>>choice1;
    switch(choicel){
        case 1: DQ.enQueueRear();
            break;
        case 2: value = DQ.deQueueRear();
            cout<<"\nThe value deleted is "<<value;
            break;
        case 3: value=DQ.deQueueFront();
            cout<<"\nThe value deleted is "<<value;
            break;
        case 4: DQ.enQueueFront();
            break;
        case 5: DQ.display_front();
            break;
        case 6: DQ.display_rear();
            break;
        default: cout<<"Wrong choice";
    }
        cout<<"\nDo you want to perform another operation (Y/y/n/N): ";
        cin>>ch;
    } while(ch=='y'||ch=='Y');
    return 0;
}
/
*********************************************************************************************************/
Output : -

ubuntu@ubuntu-OptiPlex-3090:~/Documents/dsl_practicals$ g++ practical13dsl.cpp
ubuntu@ubuntu-OptiPlex-3090:~/Documents/dsl_practicals$ ./a.out

******* Double Ended Queue operations*******

1.Insert at rear end
2.Delete from rear end
3.Delete from front end
4.Insert at front end
5.Display_front
6.Display_rear

Enter your choice: 1

Enter the value to be inserted:12

Do you want to perform another operation (Y/y/n/N): y

1.Insert at rear end
2.Delete from rear end
3.Delete from front end
4.Insert at front end
5.Display_front
6.Display_rear

Enter your choice: 1

Enter the value to be inserted:18

Do you want to perform another operation (Y/y/n/N): y

1.Insert at rear end
2.Delete from rear end
3.Delete from front end
4.Insert at front end
5.Display_front
6.Display_rear

Enter your choice: 1
Enter the value to be inserted:14

Do you want to perform another operation (Y/y/n/N): y

1.Insert at rear end
2.Delete from rear end
3.Delete from front end
4.Insert at front end
5.Display_front
6.Display_rear

Enter your choice: 2

The value deleted is 14
Do you want to perform another operation (Y/y/n/N): y

1.Insert at rear end
2.Delete from rear end
3.Delete from front end
4.Insert at front end
5.Display_front
6.Display_rear

Enter your choice: 3

The value deleted is 12
Do you want to perform another operation (Y/y/n/N): y

1.Insert at rear end
2.Delete from rear end
3.Delete from front end
4.Insert at front end
5.Display_front
6.Display_rear

Enter your choice: 4

Enter the value to be inserted:34

Do you want to perform another operation (Y/y/n/N): y

1.Insert at rear end
2.Delete from rear end
3.Delete from front end
4.Insert at front end
5.Display_front
6.Display_rear

Enter your choice: 5

The Queue_front element is:34
Do you want to perform another operation (Y/y/n/N): y

1.Insert at rear end
2.Delete from rear end
3.Delete from front end
4.Insert at front end
5.Display_front
6.Display_rear

Enter your choice: 6

The Queue Rear element is:18
Do you want to perform another operation (Y/y/n/N): n
```