7. Implement Queues using Linked list.

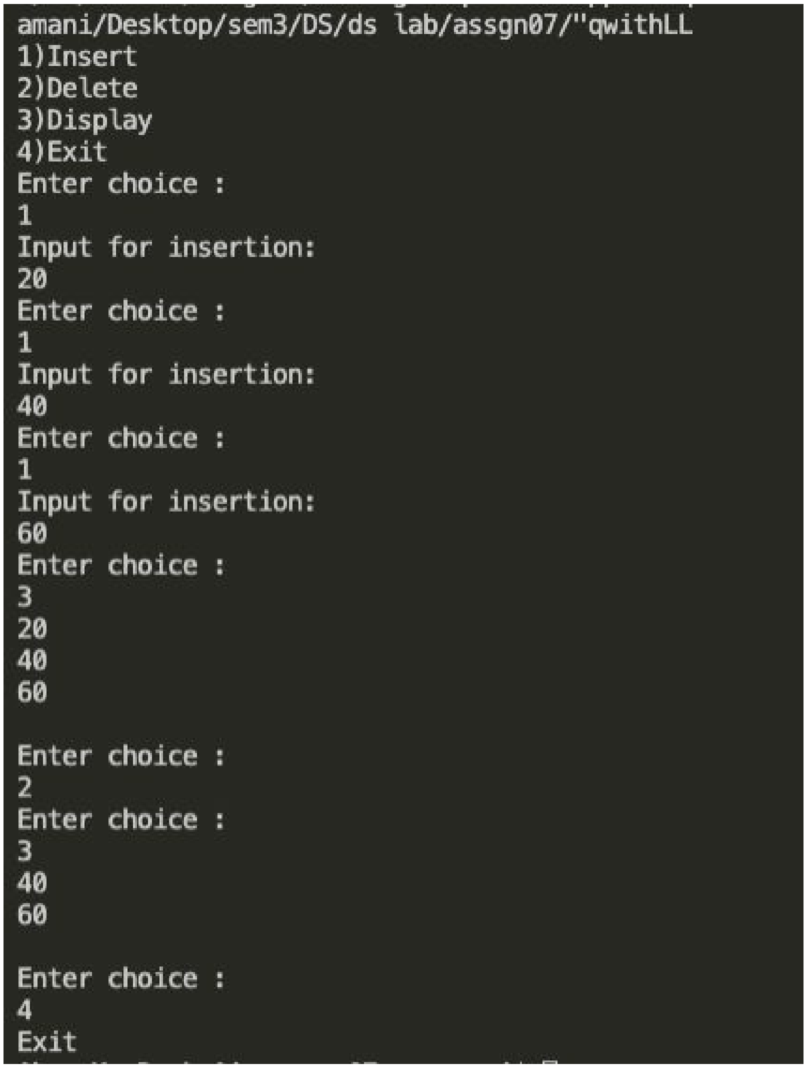
CODE :

```
#include <iostream>
using namespace std;
struct Node
{
    int data;
    struct Node *next;
} *front = NULL, *rear = NULL;
void enqueue(int x)
{
    struct Node *t;
    t = new Node;
    if (t == NULL)
        cout<<"Queue is FUll\n";
    else
    {
        t->data = x;
        t->next = NULL;
        if (front == NULL)
            front = rear = t;
        else
        {
            rear->next = t;
            rear = t;
        }
    }
}
int dequeue()
{
    int x = -1;
    struct Node *t;
    if (front == NULL)

        cout<<"Queue is Empty\n";
    else
    {
        x = front->data;
        t = front;
        front = front->next;
        free(t);
    }
    return x;
}
void Display()
{
    struct Node *p = front;
    while (p)
    {
        cout<<p->data<<endl;
        p = p->next;
    }
    cout<<endl;
}
int main()
{
  int ch, val;
    cout << "1)Insert\n";
    cout << "2)Delete\n";
    cout << "3)Display\n";
    cout << "4)Exit\n";
    do
    {

        cout << "Enter choice : " << endl;
        cin >> ch;
        switch (ch)
        {
        case 1:
            cout << "Input for insertion: " << endl;
            cin >> val;
            enqueue(val);
            break;
        case 2:
            dequeue();
            break;
        case 3:
            Display();
            break;
        case 4:
            cout << "Exit\n";
            break;
        default:
            cout << "Incorrect!\n";
        }
    } while (ch != 4);
    return 0;
}
```

OUTPUT :



```
amani/Desktop/sem3/DS/ds lab/assgn07/"qwithLL
1)Insert
2)Delete
3)Display
4)Exit
Enter choice :
1
Input for insertion:
20
Enter choice :
1
Input for insertion:
40
Enter choice :
1
Input for insertion:
60
Enter choice :
3
20
40
60

Enter choice :
2
Enter choice :
3
40
60

Enter choice :
4
Exit
```