

Experiment 7

Program to implement Queue using Linked List.

```
#include <iostream>

using namespace std;

struct Node {
    int data;
    Node* next;
};

class Queue {
private:
    Node* front;
    Node* rear;

public:
    Queue() {
        front = nullptr;
        rear = nullptr;
    }

    void enqueue(int value) {
        Node* newNode = new Node();
        newNode->data = value;
        newNode->next = nullptr;
        if (rear == nullptr) {
            front = rear = newNode;
        } else {
            rear->next = newNode;
            rear = newNode;
        }
    }
};
```

```

    }
}

void dequeue() {
    if (isEmpty()) {
        cout << "Queue Underflow\n";
        return;
    }
    Node* temp = front;
    front = front->next;
    if (front == nullptr) {
        rear = nullptr;
    }
    delete temp;
}

```

```

int peek() {
    if (isEmpty()) {
        cout << "Queue is empty\n";
        return -1;
    }
    return front->data;
}

```

```

bool isEmpty() {
    return front == nullptr;
}

```

```

void display() {
    Node* temp = front;
    while (temp != nullptr) {

```

```

        cout << temp->data << " ";

        temp = temp->next;
    }

    cout << endl;
}

};

int main() {
    Queue queue;

    queue.enqueue(10);
    queue.enqueue(20);
    queue.enqueue(30);

    cout << "Queue elements: ";
    queue.display();

    queue.dequeue();

    cout << "Queue after dequeue: ";
    queue.display();

    cout << "Front element: " << queue.peek() << endl;

    return 0;
}

```

OUTPUT

```

Queue elements: 10 20 30
Queue after dequeue: 20 30
Front element: 20

```