```
// Justin Dang Student ID: 1148267
/*
Creates an link based Queue using enqueue and dequeue methods

Queue is size "unlimited" only limited by computer hardware

Error is thrown when attempting to dequeue nothing from the list

only takes in data type int for each node list
*/
#include <iostream>
using namespace std;

class Node {
public:
    int data;                   // data in each node
    class Node* next;    // address of next node(or null/0 to define as end of Queue)
    Node(int info, Node* ptr = 0) { // Structure for each node
        data = info;
        next = ptr;
    }
};

class LinkedQueue {
public:
    LinkedQueue() { front = back = 0; }   // constructor that is used to ensure our first node and last are set properly
    bool isEmpty() { return front == 0; } // checks if our first node has a address(implying there is another node)
    void enqueue(int info) {
        Node* temp = new Node(info); // new node stored in temp(note no address given to show it is last node)
        nodeCount++;                 // increase node count
        if (back == 0)
            front = back = temp;     // if our last node is null/0 then we create a node that is the front and back
        else
        {
            back->next = temp;       // otherwise we set the address of our last node to our new node
            back = temp;             // set our last node = new node
        }
    }
    int dequeue() {
        if (isEmpty()) {             // if our queue is empty then throw error
            cout << "The Queue is empty.\n\n";
            nodeCount++;             // offset the nodeCount decrement
            return -999;
        }
        nodeCount--;
        Node* temp;                  // temp node where our front node goes
```

```cpp
        int frontInt = front->data;  // we take the data from the front node
        temp = front;
        front = front->next;         // our element after the first element is now the front element
        return frontInt;             // we return data for from our front node
    }
    void print() {
        cout << "There are " << nodeCount << " items in the Queue." << "\nThe queue from first to last is: ";
        for (Node* temp = front; temp != 0; temp = temp->next)
            cout << temp->data << " ";// All queue data is printed
        cout << "\n\n";
    }
private:
    Node* front, * back;
    int nodeCount;
};
int main()
{
    cout << "Creating Link Based Queue. . .\n\n";
    LinkedQueue* ptr = new LinkedQueue(); // New queue is started

    cout << "Enqueue(15)\n\n";            // begins adding ints to queue
    ptr->enqueue(15);
    ptr->print();

    cout << "Enqueue(5)\n\n";
    ptr->enqueue(5);
    ptr->print();

    cout << "Enqueue(20)\n\n";
    ptr->enqueue(20);
    ptr->print();

    cout << "Enqueue(10)\n\n";
    ptr->enqueue(10);
    ptr->print();

    cout << "Enqueue(25)\n\n";
    ptr->enqueue(25);
    ptr->print();

    cout << "Enqueue(35)\n\n";
    ptr->enqueue(35);
    ptr->print();

    cout << "----------------------------------\n\n";
    cout << "Removing all nodes from queue. . .\n\n";
    while (!ptr->isEmpty()) {                                             // while the queue is not empty
        cout << "Removing " << ptr->dequeue() << " from the queue. \n\n"; // remove an item
        ptr->print();
    }
    ptr->dequeue();                                                       // attempt to remove null to show error
}
/*// --------------------------------- case 1:
Creating Link Based Queue. . .

Enqueue(15)

There are 1 items in the Queue.
The queue from first to last is: 15

Enqueue(5)

There are 2 items in the Queue.
The queue from first to last is: 15 5

Enqueue(20)

There are 3 items in the Queue.
The queue from first to last is: 15 5 20

Enqueue(10)

There are 4 items in the Queue.
The queue from first to last is: 15 5 20 10

Enqueue(25)

There are 5 items in the Queue.
The queue from first to last is: 15 5 20 10 25

Enqueue(35)

There are 6 items in the Queue.
The queue from first to last is: 15 5 20 10 25 35

----------------------------------

Removing all nodes from queue. . .

Removing 15 from the queue.

There are 5 items in the Queue.
The queue from first to last is: 5 20 10 25 35

Removing 5 from the queue.

There are 4 items in the Queue.
The queue from first to last is: 20 10 25 35

```

```
Removing 20 from the queue.

There are 3 items in the Queue.
The queue from first to last is: 10 25 35

Removing 10 from the queue.

There are 2 items in the Queue.
The queue from first to last is: 25 35

Removing 25 from the queue.

There are 1 items in the Queue.
The queue from first to last is: 35

Removing 35 from the queue.

There are 0 items in the Queue.
The queue from first to last is:

The Queue is empty.
*/// ---------------------------------
```