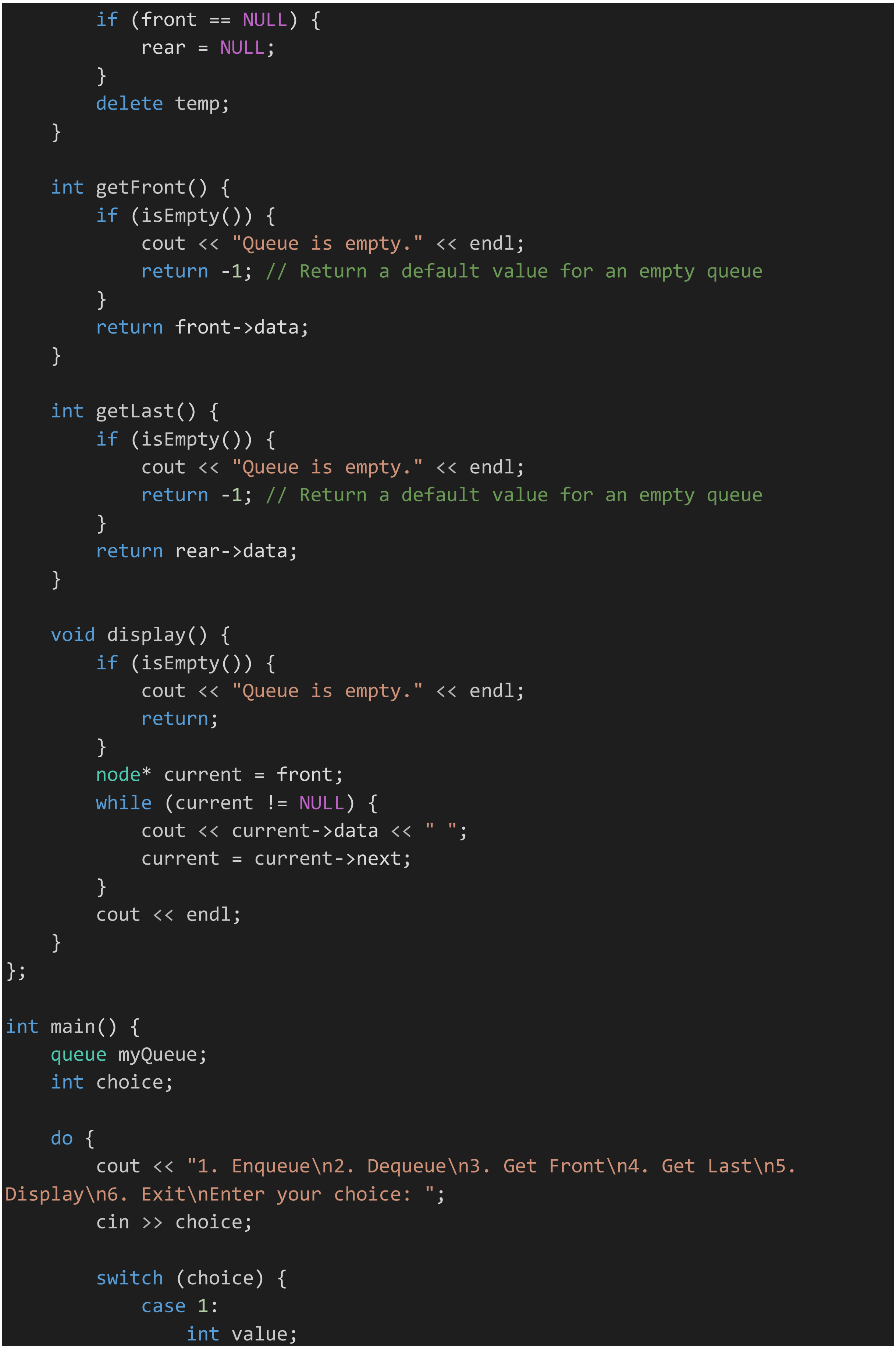
Q Queue using singly linked list

```
#include <iostream>

using namespace std;

class node {
public:
    int data;
    node* next;
    node(int val) {
        data = val;
        next = NULL;
    }
};

class queue {
private:
    node* front;
    node* rear;

public:
    queue() {
        front = NULL;
        rear = NULL;
    }

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

    void enqueue(int data) {
        node* newnode = new node(data);
        if (isEmpty()) {
            front = newnode;
            rear = newnode;
        } else {
            rear->next = newnode;
            rear = newnode;
        }
    }

    void dequeue() {
        if (isEmpty()) {
            cout << "Queue is empty. Cannot dequeue." << endl;
            return;
        }
        node* temp = front;
        front = front->next;
```



```
        if (front == NULL) {
            rear = NULL;
        }
        delete temp;
    }

    int getFront() {
        if (isEmpty()) {
            cout << "Queue is empty." << endl;
            return -1; // Return a default value for an empty queue
        }
        return front->data;
    }

    int getLast() {
        if (isEmpty()) {
            cout << "Queue is empty." << endl;
            return -1; // Return a default value for an empty queue
        }
        return rear->data;
    }

    void display() {
        if (isEmpty()) {
            cout << "Queue is empty." << endl;
            return;
        }
        node* current = front;
        while (current != NULL) {
            cout << current->data << " ";
            current = current->next;
        }
        cout << endl;
    }
};

int main() {
    queue myQueue;
    int choice;

    do {
        cout << "1. Enqueue\n2. Dequeue\n3. Get Front\n4. Get Last\n5.
Display\n6. Exit\nEnter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                int value;
                cout << "Enter the value to enqueue: ";
                cin >> value;
                myQueue.enqueue(value);
                break;
            case 2:
                myQueue.dequeue();
                break;
            case 3:
                cout << "Front element: " << myQueue.getFront() << endl;
                break;
            case 4:
                cout << "Last element: " << myQueue.getLast() << endl;
                break;
            case 5:
                cout << "Queue elements: ";
                myQueue.display();
                break;
            case 6:
                cout << "Exiting the program." << endl;
                break;
            default:
                cout << "Invalid choice. Please enter a valid option." <<
endl;
        }
    } while (choice != 6);

    return 0;
}
```

OUTPUT ➔

```
PS C:\Users\HP\Desktop\DSA> cd "c:\Users\HP\Desktop\DSA\" ; if ($?) { g++ queue.cpp -o queue } ;
if ($?) { .\queue }

1. Enqueue

2. Dequeue

3. Get Front

4. Get Last

5. Display

6. Exit

Enter your choice: 1

Enter the value to enqueue: 23

1. Enqueue
```

```
2. Dequeue
3. Get Front
4. Get Last
5. Display
6. Exit
Enter your choice: 1
Enter the value to enqueue: 24
1. Enqueue
2. Dequeue
3. Get Front
4. Get Last
5. Display
6. Exit
Enter your choice: 1
Enter the value to enqueue: 45
1. Enqueue
2. Dequeue
3. Get Front
4. Get Last
5. Display
6. Exit
Enter your choice: 1
Enter the value to enqueue: 34
1. Enqueue
2. Dequeue
3. Get Front
4. Get Last
5. Display
6. Exit
Enter your choice: 5
Queue elements: 23 24 45 34
```

```
1. Enqueue

2. Dequeue

3. Get Front

4. Get Last

5. Display

6. Exit

Enter your choice: 6

Exiting the program.
```