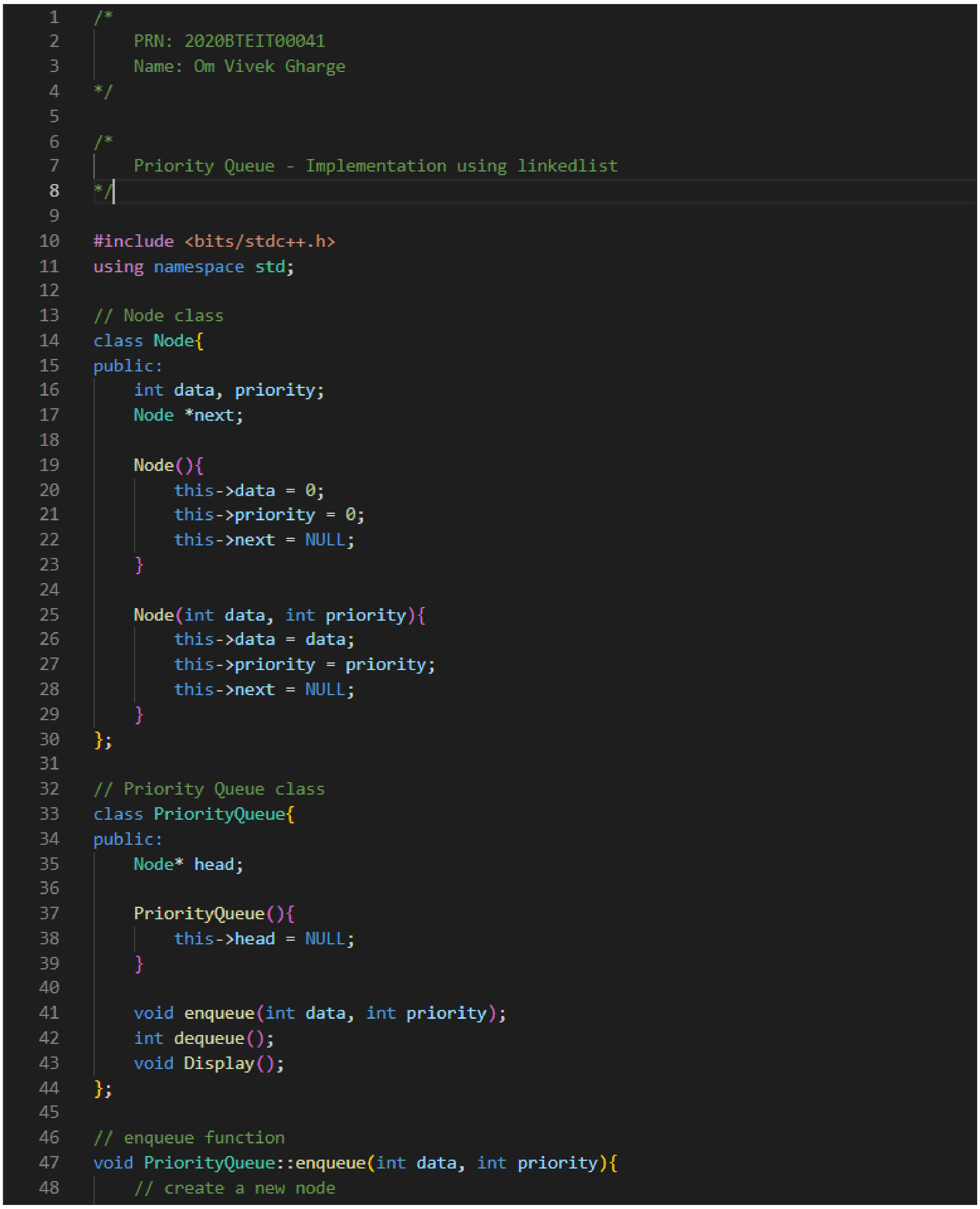
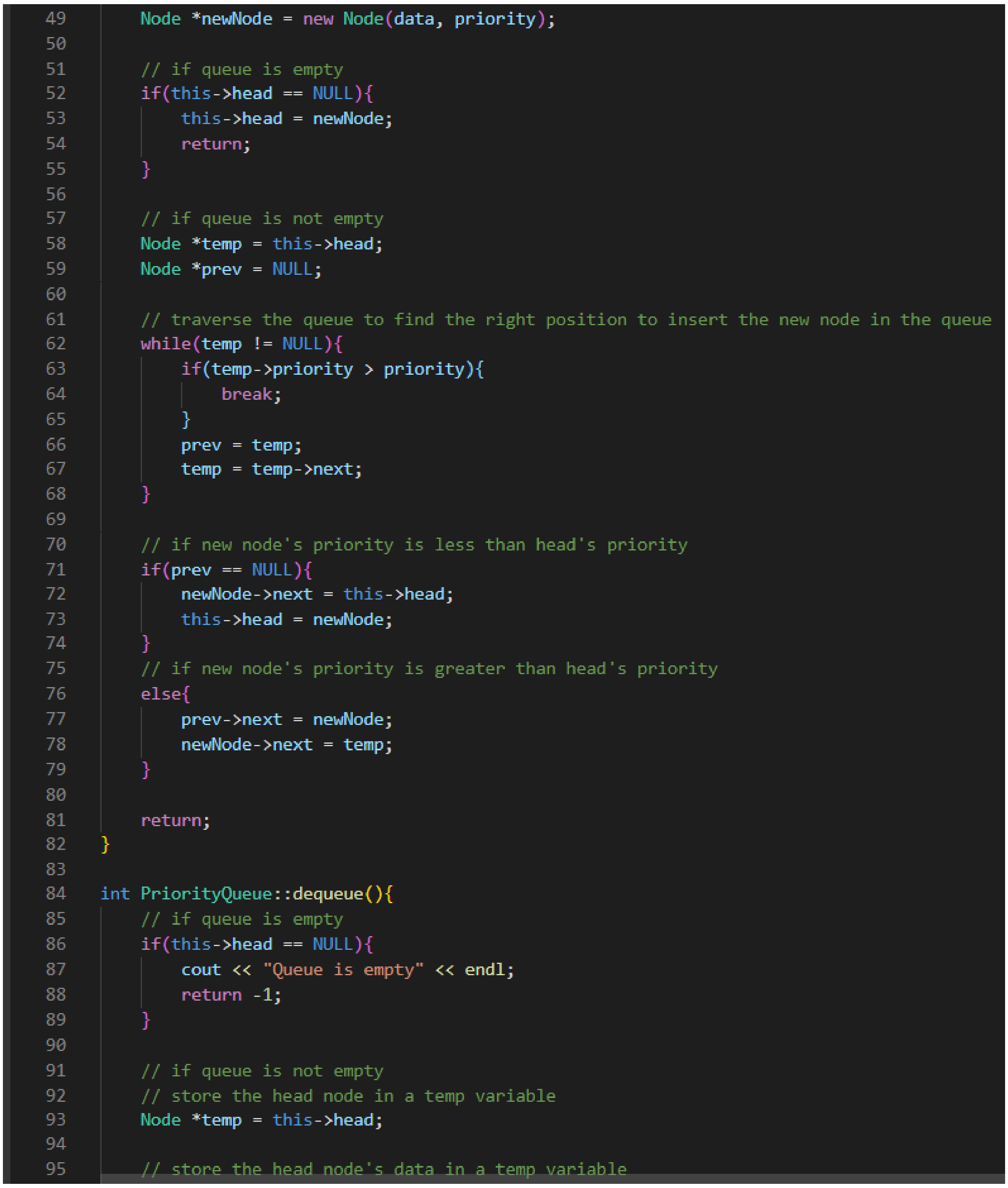
PRN: 2020BTEIT00041

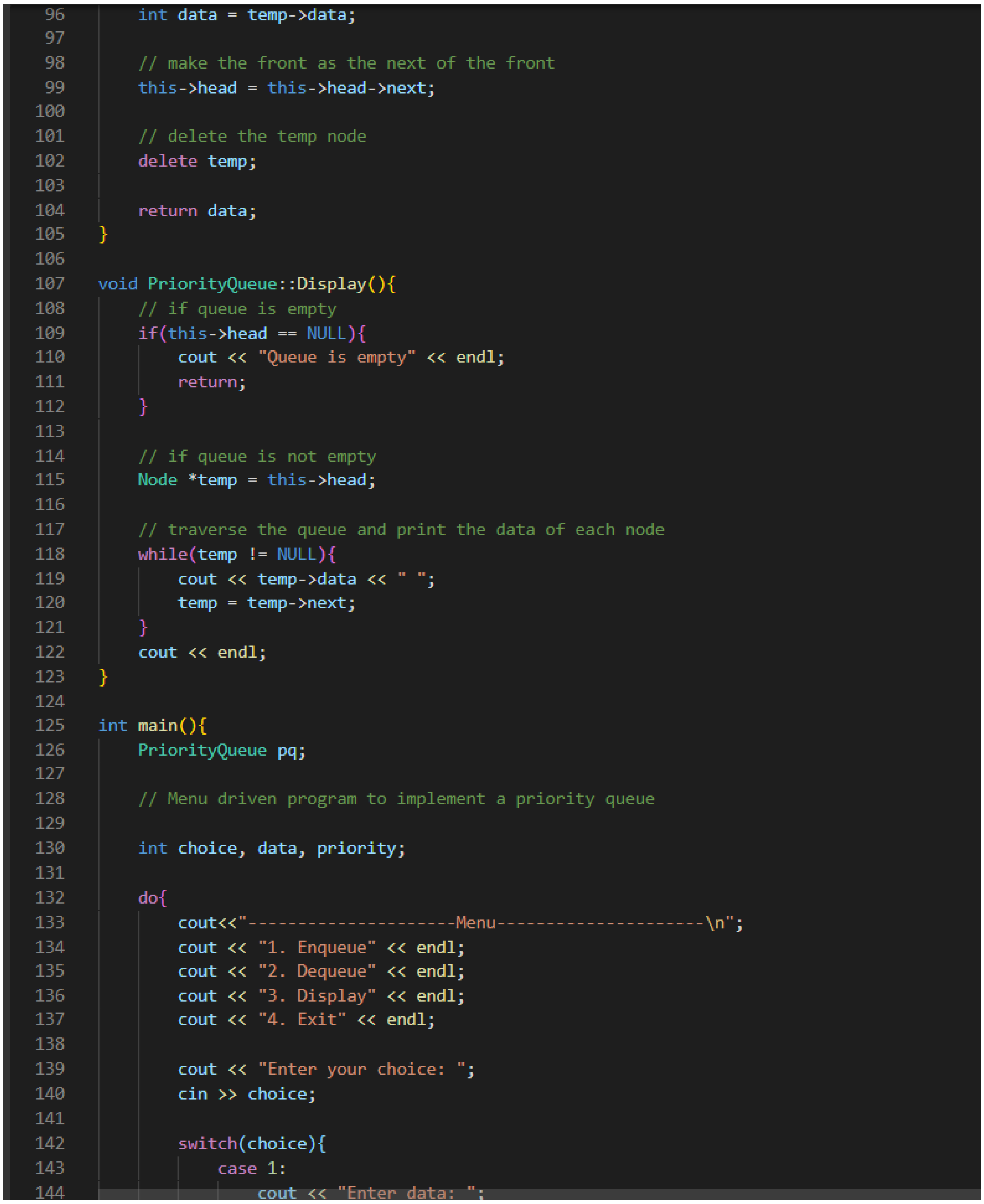
Priority Queue – Using LinkedList:



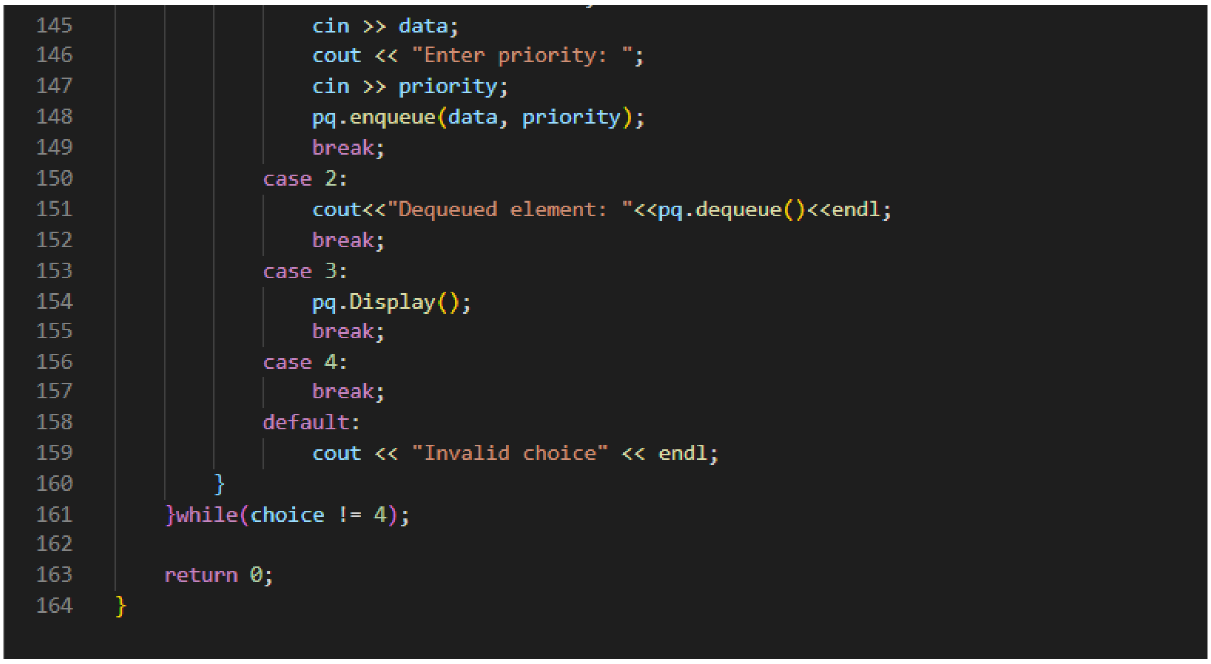
```cpp
/*
    PRN: 2020BTEIT00041
    Name: Om Vivek Gharge
*/

/*
    Priority Queue - Implementation using linkedlist
*/

#include <bits/stdc++.h>
using namespace std;

// Node class
class Node{
public:
    int data, priority;
    Node *next;

    Node(){
        this->data = 0;
        this->priority = 0;
        this->next = NULL;
    }

    Node(int data, int priority){
        this->data = data;
        this->priority = priority;
        this->next = NULL;
    }
};

// Priority Queue class
class PriorityQueue{
public:
    Node* head;

    PriorityQueue(){
        this->head = NULL;
    }

    void enqueue(int data, int priority);
    int dequeue();
    void Display();
};

// enqueue function
void PriorityQueue::enqueue(int data, int priority){
    // create a new node
```



```
    Node *newNode = new Node(data, priority);

    // if queue is empty
    if(this->head == NULL){
        this->head = newNode;
        return;
    }

    // if queue is not empty
    Node *temp = this->head;
    Node *prev = NULL;

    // traverse the queue to find the right position to insert the new node in the queue
    while(temp != NULL){
        if(temp->priority > priority){
            break;
        }
        prev = temp;
        temp = temp->next;
    }

    // if new node's priority is less than head's priority
    if(prev == NULL){
        newNode->next = this->head;
        this->head = newNode;
    }
    // if new node's priority is greater than head's priority
    else{
        prev->next = newNode;
        newNode->next = temp;
    }

    return;
}

int PriorityQueue::dequeue(){
    // if queue is empty
    if(this->head == NULL){
        cout << "Queue is empty" << endl;
        return -1;
    }

    // if queue is not empty
    // store the head node in a temp variable
    Node *temp = this->head;

    // store the head node's data in a temp variable
```

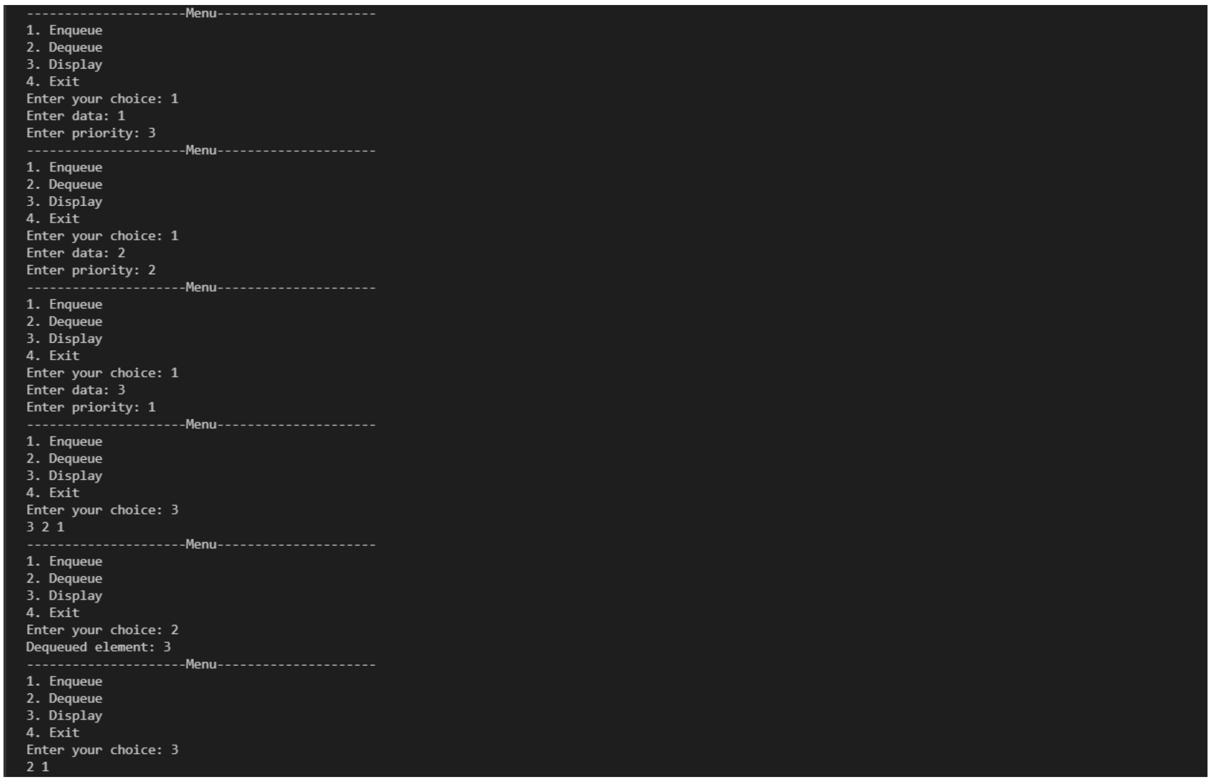


```cpp
    int data = temp->data;

    // make the front as the next of the front
    this->head = this->head->next;

    // delete the temp node
    delete temp;

    return data;
}

void PriorityQueue::Display(){
    // if queue is empty
    if(this->head == NULL){
        cout << "Queue is empty" << endl;
        return;
    }

    // if queue is not empty
    Node *temp = this->head;

    // traverse the queue and print the data of each node
    while(temp != NULL){
        cout << temp->data << " ";
        temp = temp->next;
    }
    cout << endl;
}

int main(){
    PriorityQueue pq;

    // Menu driven program to implement a priority queue

    int choice, data, priority;

    do{
        cout<<"---------------------Menu---------------------\n";
        cout << "1. Enqueue" << endl;
        cout << "2. Dequeue" << endl;
        cout << "3. Display" << endl;
        cout << "4. Exit" << endl;

        cout << "Enter your choice: ";
        cin >> choice;

        switch(choice){
            case 1:
                cout << "Enter data: ";
```



```
                cin >> data;
                cout << "Enter priority: ";
                cin >> priority;
                pq.enqueue(data, priority);
                break;
            case 2:
                cout<<"Dequeued element: "<<pq.dequeue()<<endl;
                break;
            case 3:
                pq.Display();
                break;
            case 4:
                break;
            default:
                cout << "Invalid choice" << endl;
        }
    }while(choice != 4);

    return 0;
}
```

OUTPUT:



```
---------------------Menu---------------------
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 1
Enter data: 1
Enter priority: 3
---------------------Menu---------------------
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 1
Enter data: 2
Enter priority: 2
---------------------Menu---------------------
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 1
Enter data: 3
Enter priority: 1
---------------------Menu---------------------
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 3
3 2 1
---------------------Menu---------------------
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 2
Dequeued element: 3
---------------------Menu---------------------
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 3
2 1
```