Aim-This program demonstrates a **priority queue**

#include <iostream>

#include <queue>

using namespace std;

class PriorityQueue {

private:

priority\_queue<int> pq;

public:

// Enqueue operation

void enqueue(int value) {

pq.push(value);

cout << value << " enqueued successfully.\n";

}

// Dequeue operation

void dequeue() {

if (pq.empty()) {

cout << "Queue is empty. Cannot dequeue.\n";

return;

}

cout << pq.top() << " dequeued successfully.\n";

pq.pop();

}

// Check if the queue is empty

bool isEmpty() {

return pq.empty();

}

// Display queue elements

void display() {

if (isEmpty()) {

cout << "Queue is empty.\n";

return;

}

// Create a temporary priority queue for display

priority\_queue<int> temp = pq;

cout << "Queue elements (in priority order): ";

while (!temp.empty()) {

cout << temp.top() << " ";

temp.pop();

}

cout << "\n";

}

};

int main() {

PriorityQueue pq;

int choice, value;

do {

cout << "\n1. Enqueue\n2. Dequeue\n3. Check if Empty\n4. Display\n5. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Enter value to enqueue: ";

cin >> value;

pq.enqueue(value);

break;

case 2:

pq.dequeue();

break;

case 3:

if (pq.isEmpty())

cout << "Queue is empty.\n";

else

cout << "Queue is not empty.\n";

break;

case 4:

pq.display();

break;

case 5:

cout << "Exiting program.\n";

break;

default:

cout << "Invalid choice. Try again.\n";

}

} while (choice != 5);

return 0;

}