TASK 1:

#include <iostream>

using namespace std;

class Queue

{

private:

static const int MAX\_SIZE = 5;

int front, rear, size;

int elements[MAX\_SIZE];

public:

Queue()

{

front=-1;

rear=-1;

size=0;

}

bool isEmpty()

{

return size == 0;

}

bool isFull()

{

return size == MAX\_SIZE;

}

void push(int val)

{

if (isFull())

{

cout << "QUEUE FULL" << endl;

return;

}

if (isEmpty())

{

front = rear = 0;

}

else

{

rear = (rear + 1) ;

}

elements[rear] = val;

size++;

}

void pop()

{

if (isEmpty())

{

cout << "QUEUE EMPTY" << endl;

return;

}

if (front == rear)

{

front = rear = -1;

}

else

{

front = (front + 1) ;

}

size--;

}

int getFront()

{

if (isEmpty())

{

cout << "QUEUE EMPTY" << endl;

return -1;

}

return elements[front];

}

void show()

{

if (isEmpty())

{

cout << "QUEUE EMPTY" << endl;

return;

}

for (int i = front; i <=rear; i ++)

{

cout << elements[i] << " ";

}

cout << endl;

}

};

int main()

{

Queue q;

q.push(100);

q.push(20);

q.push(31);

q.push(11);

q.pop();

q.pop();

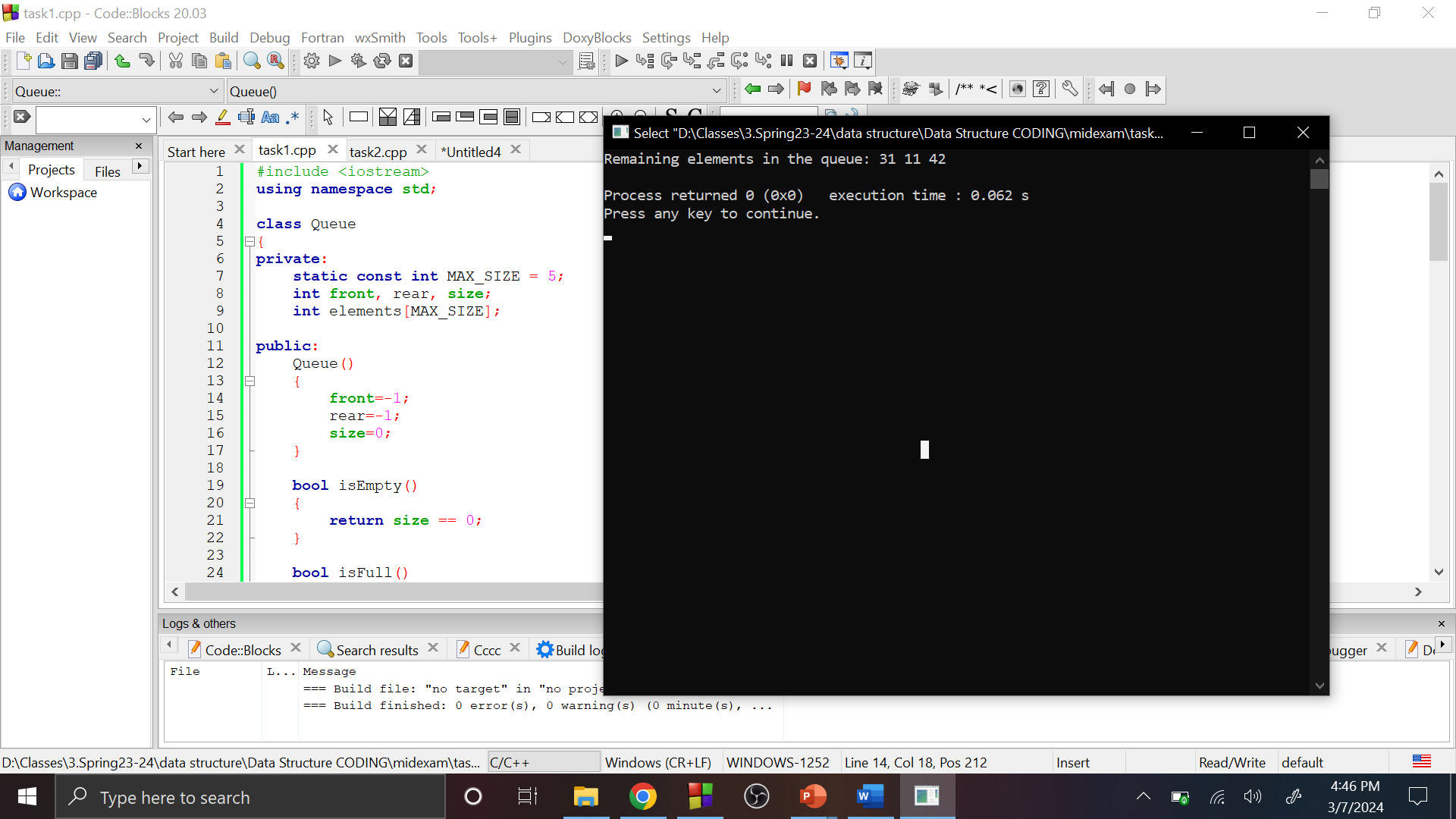
q.push(42);

cout << "Remaining elements in the queue: ";

q.show();

return 0;

}



TASK 2:

#include <iostream>

using namespace std;

class CircularQueue

{

private:

static const int MAX\_SIZE = 5;

int front, rear;

int q[MAX\_SIZE];

public:

CircularQueue()

{

front=-1;

rear=-1;

}

void enqueue(int val)

{

if ((rear+1)%MAX\_SIZE==front)

{

cout << "Queue overflowed!!!" << endl;

}

else if((front==-1 && rear==-1))

{

front=rear=0;

q[rear]=val;

}

else

{

rear=(rear+1)%MAX\_SIZE;

q[rear]=val;

}

}

void dequeue()

{

if(front==rear)

{

cout<<q[front];

front=rear=-1;

}

else

{

q[front];

front = (front + 1)%MAX\_SIZE;

}

}

void display()

{

if (front==-1 && rear==-1)

{

cout << "Queue is empty" << endl;

return;

}

else

{

for (int i = front; i <=rear; i ++)

{

cout << q[i] << " ";

}

cout<<endl;

}

}

};

int main()

{

CircularQueue cq;

int choice, value;

do

{

cout << "What you want to do?"<<endl;

cout << "1. Enqueue element in the queue"<<endl;

cout << "2. Dequeue element from the queue"<<endl;

cout << "3. Display the queue"<<endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice)

{

case 1:

cout << "Enter the value to enqueue: ";

cin >> value;

cq.enqueue(value);

break;

case 2:

cq.dequeue();

break;

case 3:

cq.display();

break;

default:

cout << "Invalid choice. Please enter a valid option.\n";

}

}

while (true);

}

