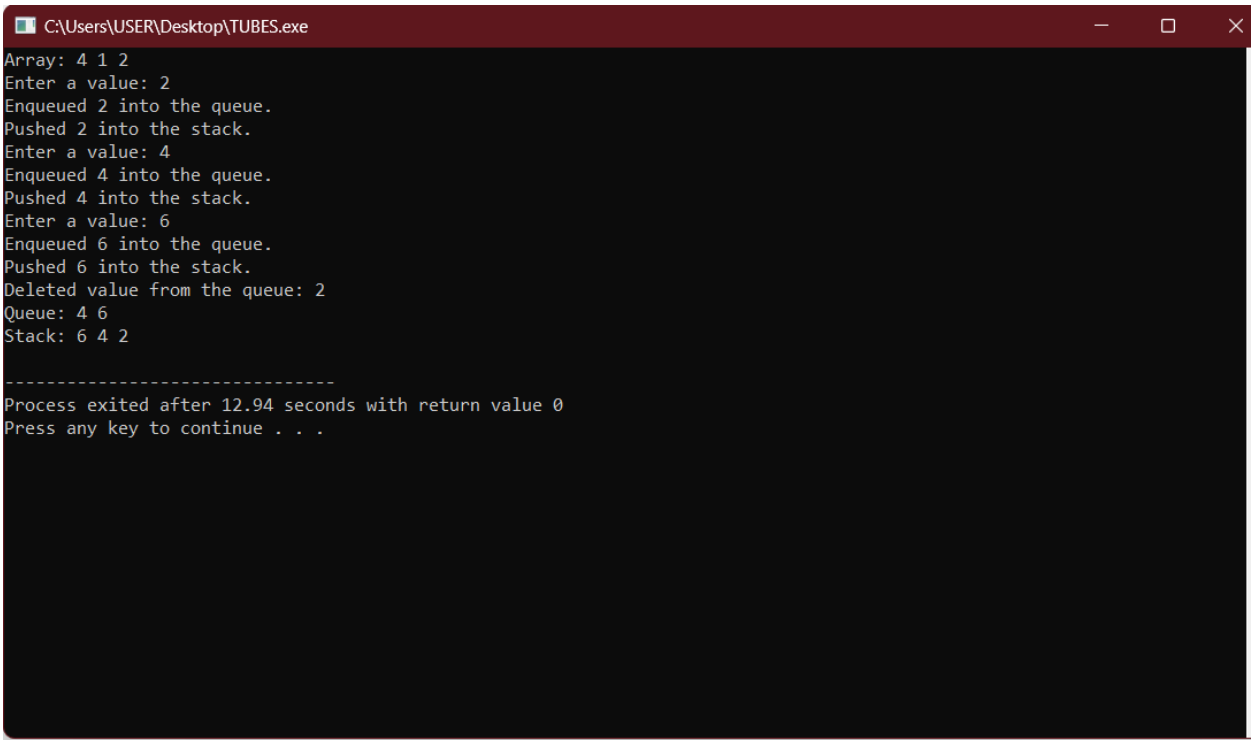
```cpp
#include <iostream>
using namespace std;

const int MAX_SIZE = 3;

class Queue {
private:
    int front, rear;
    int arr[MAX_SIZE];

public:
    Queue() {
        front = -1;
        rear = -1;
    }

    bool isFull() {
        return (rear == MAX_SIZE - 1);
    }

    bool isEmpty() {
        return (front == -1 || front > rear);
    }

    void enqueue(int data) {
        if (isFull()) {
            cout << "Queue is full. Cannot enqueue." << endl;
            return;
        }
        rear++;
        arr[rear] = data;
        if (front == -1) {
            front = 0;
        }
        cout << "Enqueued " << data << " into the queue." << endl;
    }

    int dequeue() {
        if (isEmpty()) {
            cout << "Queue is empty. Cannot dequeue." << endl;
            return -1;
        }
        int data = arr[front];
        front++;
        return data;
    }

    void display() {
        if (isEmpty()) {
            cout << "Queue is empty." << endl;
            return;
        }
        cout << "Queue: ";
        for (int i = front; i <= rear; i++) {
            cout << arr[i] << " ";
        }
        cout << endl;
    }
};

class Stack {
private:
    int top;
    int arr[MAX_SIZE];

public:
    Stack() {
        top = -1;
    }

    bool isFull() {
        return (top == MAX_SIZE - 1);
    }

    bool isEmpty() {
        return (top == -1);
    }

    void push(int data) {
        if (isFull()) {
```

```
            cout << "Stack is full. Cannot push." << endl;
            return;
        }
        top++;
        arr[top] = data;
        cout << "Pushed " << data << " into the stack." << endl;
    }

    int pop() {
        if (isEmpty()) {
            cout << "Stack is empty. Cannot pop." << endl;
            return -1;
        }
        int data = arr[top];
        top--;
        return data;
    }

    void display() {
        if (isEmpty()) {
            cout << "Stack is empty." << endl;
            return;
        }
        cout << "Stack: ";
        for (int i = top; i >= 0; i--) {
            cout << arr[i] << " ";
        }
        cout << endl;
    }
};

int main() {
    int nim = 22040700112;
    int x = (nim % 100);

    int arr[3] = {4, 1, 2};

    Queue queue;
    Stack stack;

    cout << "Array: ";
    for (int i = 0; i < 3; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;

    for (int i = 0; i < 3; i++) {
        int value;
        cout << "Enter a value: ";
        cin >> value;
        queue.enqueue(value);
        stack.push(value);
    }

    int deletedValue = queue.dequeue();
    cout << "Deleted value from the queue: " << deletedValue << endl;

    queue.display();
    stack.display();

    return 0;
}
```


C:\Users\USER\Desktop\TUBES.exe
Array: 4 1 2
Enter a value: 2
Enqueued 2 into the queue.
Pushed 2 into the stack.
Enter a value: 4
Enqueued 4 into the queue.
Pushed 4 into the stack.
Enter a value: 6
Enqueued 6 into the queue.
Pushed 6 into the stack.
Deleted value from the queue: 2
Queue: 4 6
Stack: 6 4 2

Process exited after 12.94 seconds with return value 0
Press any key to continue . . .