Assignment-3

1.

import java.awt.Color;

import javax.swing.JFrame;

import javax.swing.JLabel;

public class ColorChanger implements Runnable {

private JLabel label;

private boolean running;

public ColorChanger(JLabel label) {

this.label = label;

running = true;

}

public void stop() {

running = false;

}

@Override

public void run() {

while (running) {

label.setForeground(randomColor());

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

private Color randomColor() {

int r = (int) (Math.random() \* 256);

int g = (int) (Math.random() \* 256);

int b = (int) (Math.random() \* 256);

return new Color(r, g, b);

}

public static void main(String[] args) {

JFrame frame = new JFrame("Color Changer");

JLabel label = new JLabel("Hello, world!");

frame.add(label);

frame.pack();

frame.setVisible(true);

ColorChanger colorChanger = new ColorChanger(label);

Thread thread = new Thread(colorChanger);

thread.start();

// Wait for 10 seconds before stopping the thread

try {

Thread.sleep(10000);

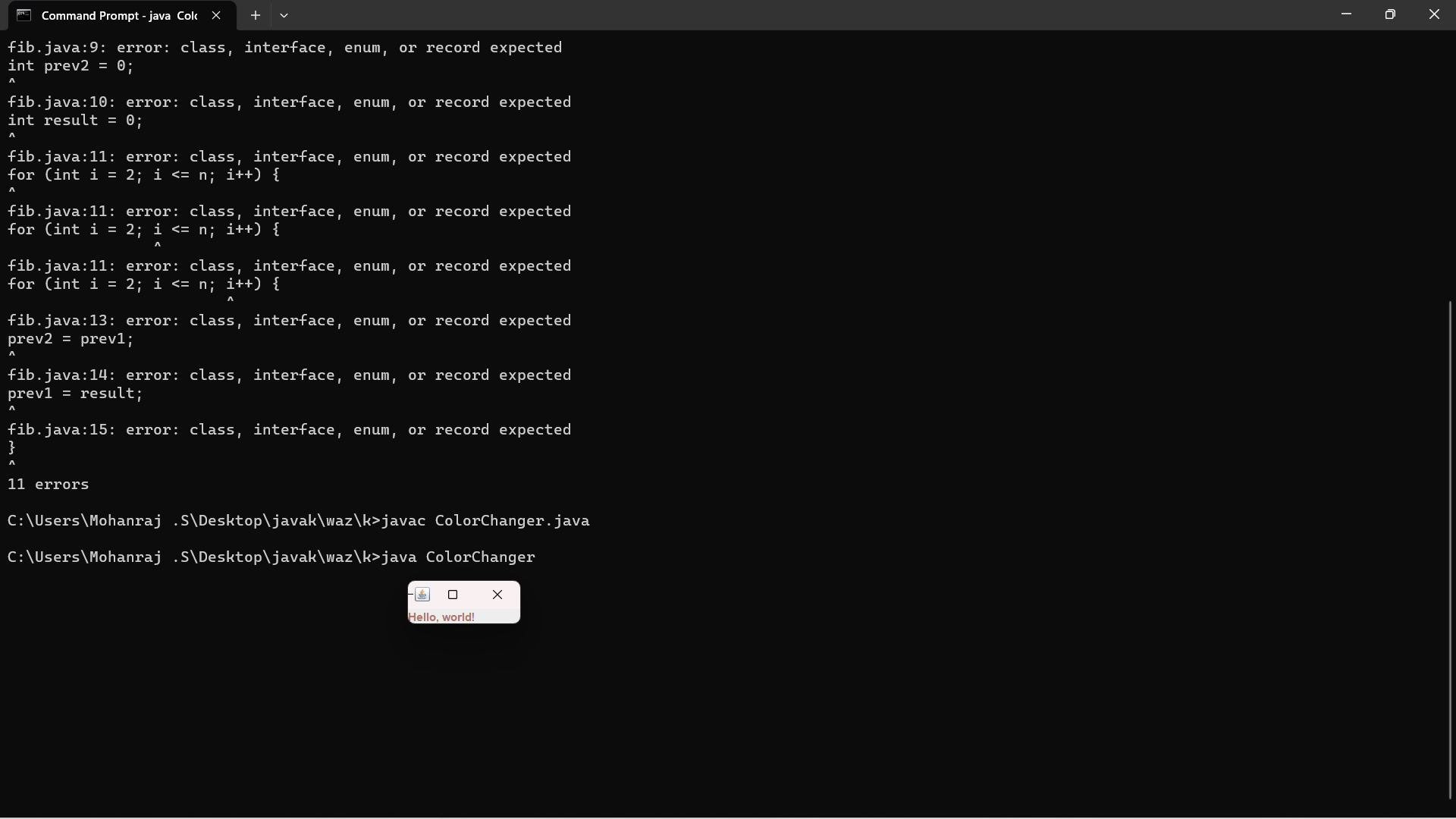
} catch (InterruptedException e) {

e.printStackTrace();

}

colorChanger.stop();

}

}

2.

class Table

{

void printTable(int n)

{

synchronized(this)

{

for(int i=1;i<=5;i++)

{

System.out.println(+n+"\*"+i+"="+(n\*i));

try

{

Thread.sleep(400);

}

catch(Exception e)

{

System.out.println(e);

}

}

}

}

}

class Mythread1 extends Thread

{

Table t;

Mythread1(Table t)

{

this.t=t;

}

public void run()

{

t.printTable(5);

}

}

class Mythread2 extends Thread

{

Table t;

Mythread2(Table t)

{

this.t=t;

}

public void run()

{

t.printTable(100);

}

}

class Use

{

public static void main(String args[])

{

Table obj = new Table();

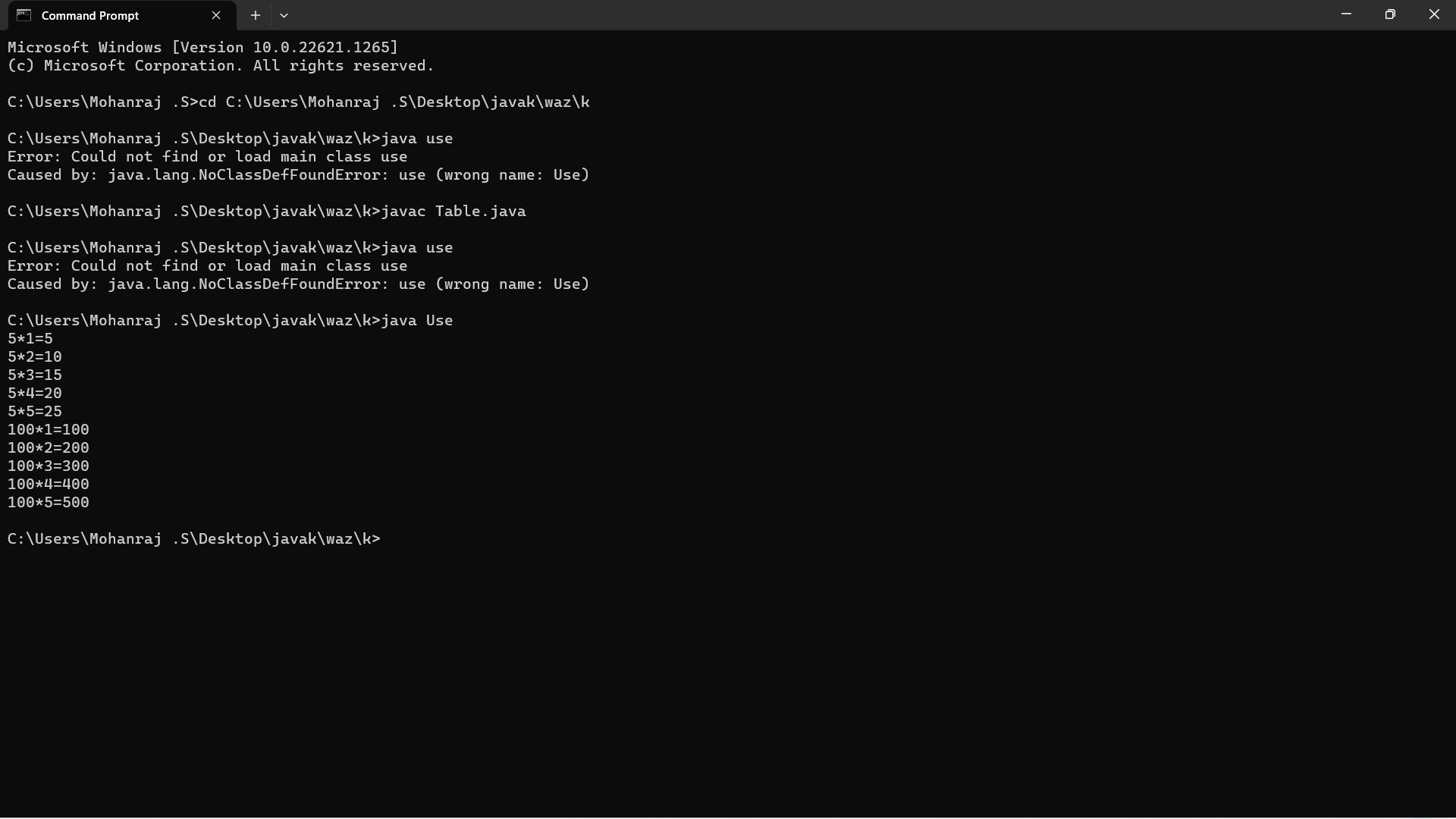
Mythread1 th1 = new Mythread1(obj);

Mythread2 th2 = new Mythread2(obj);

th1.start();

th2.start();

}

}

3.

import java.util.Scanner;

public class UglyNumbers {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the number of ugly numbers to generate: ");

int n = input.nextInt();

int[] ugly = new int[n];

ugly[0] = 1;

int i2 = 0, i3 = 0, i5 = 0;

int nextMultipleOf2 = 2, nextMultipleOf3 = 3, nextMultipleOf5 = 5;

for (int i = 1; i < n; i++) {

int nextUgly = Math.min(nextMultipleOf2, Math.min(nextMultipleOf3, nextMultipleOf5));

ugly[i] = nextUgly;

if (nextUgly == nextMultipleOf2) {

i2++;

nextMultipleOf2 = ugly[i2] \* 2;

}

if (nextUgly == nextMultipleOf3) {

i3++;

nextMultipleOf3 = ugly[i3] \* 3;

}

if (nextUgly == nextMultipleOf5) {

i5++;

nextMultipleOf5 = ugly[i5] \* 5;

}

}

System.out.println("The first " + n + " ugly numbers are:");

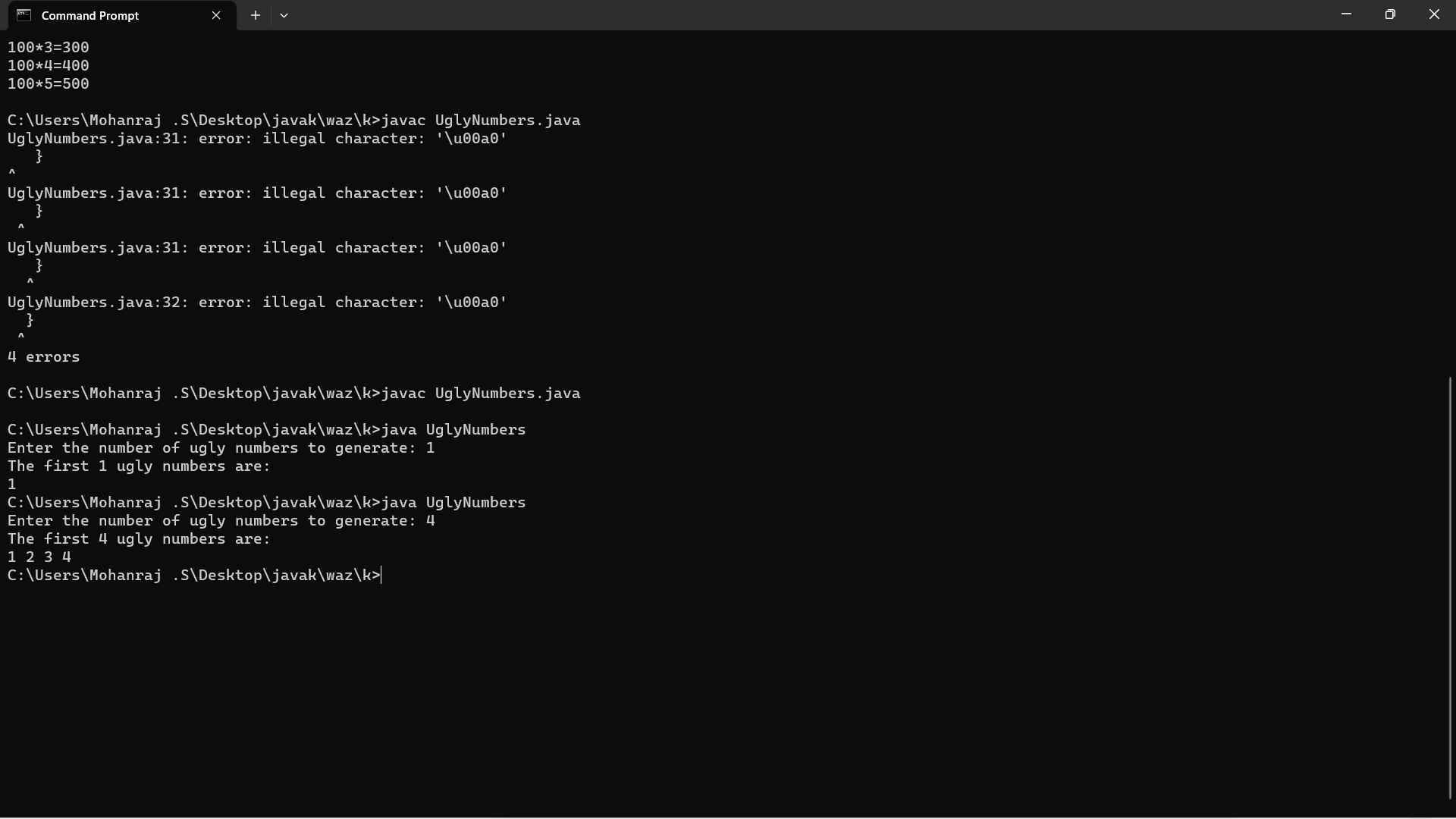
for (int i = 0; i < n; i++) {

System.out.print(ugly[i] + " ");

}

}

}



4.

class Solution {

public:

int fib(int n) {

if (n == 0) {

return 0;

}

else if (n == 1) {

return 1;

}

else {

return fib(n-1) + fib(n-2);

}

}

};

5.

class Duplicate {

static int removeDuplicates(int arr[], int n) {

if (n==0 || n==1) {

return n;

}

int[] temp = new int[n];

int j = 0;

for (int i=0; i<n-1; i++) {

if (arr[i] != arr[i+1]) {

temp[j++] = arr[i];

}

}

temp[j++] = arr[n-1];

for (int i=0; i<j; i++) {

arr[i] = temp[i];

}

return j;

}

public static void main (String[] args) {

int arr[] = {10, 20, 20, 30, 40, 40, 40, 50, 50};

int n = arr.length;

n = removeDuplicates(arr, n);

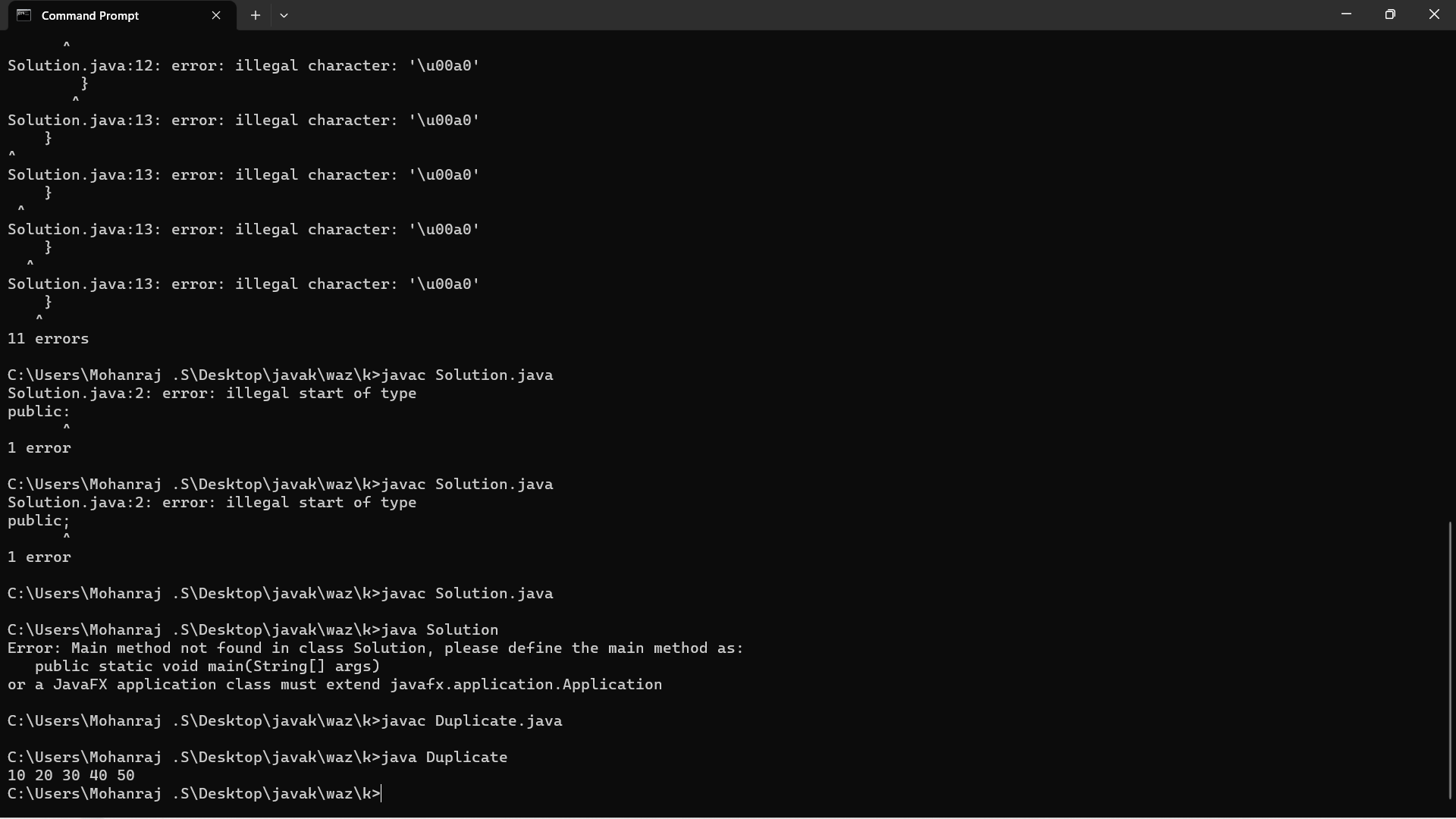
// Print updated array

for (int i=0; i<n; i++) {

System.out.print(arr[i]+" ");

       }

    }

}