

```
1 public class Counter
2 {
3     private int _count;
4     private string _name;
5
6     public Counter(string name)
7     {
8         _name = name;
9         _count = 0;
10    }
11
12    public void Increment()
13    {
14        checked
15        {
16            _count++;
17        }
18    }
19
20    public void Reset()
21    {
22        _count = 0;
23    }
24
25    public string Name
26    {
27        get
28        {
29            return _name;
30        }
31        set
32        {
33            _name = value;
34        }
35    }
36
37    public int Ticks
38    {
39        get
40        {
41            return _count;
42        }
43    }
44
45    public void ResetByDefault()
46    {
47        unchecked
48        {
49            _count = (int)2147483647473;
```

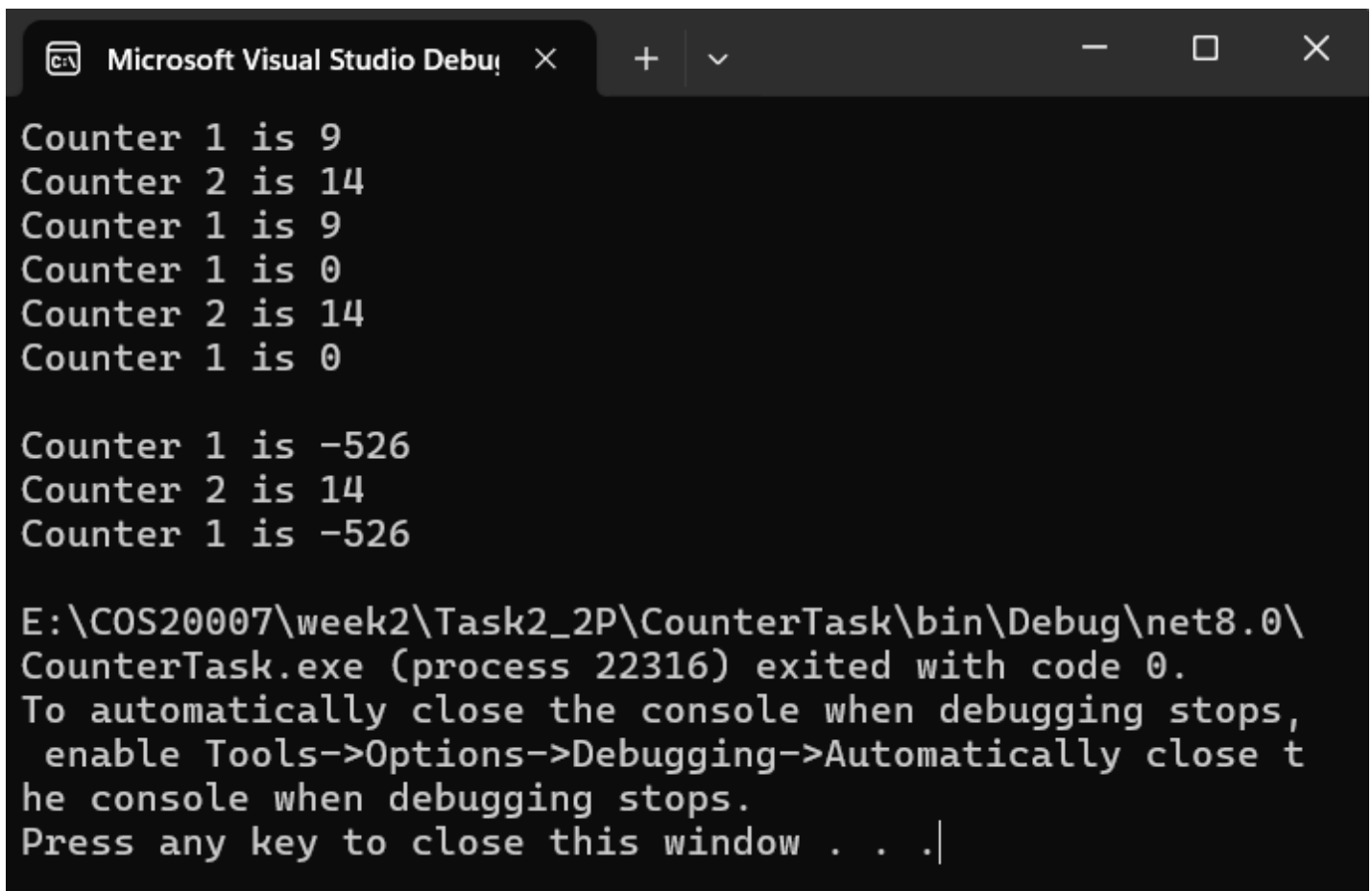
```
50     }
51
52     }
53 }
```

```
1 using System;
2 namespace CounterTask
3 {
4     internal class Program
5     {
6         private static void PrintCounters(Counter[] counters)
7         {
8             foreach (Counter c in counters)
9             {
10                 Console.WriteLine("{0} is {1}", c.Name, c.Ticks);
11             }
12         }
13
14         static void Main(string[] args)
15         {
16             Counter[] myCounters= new Counter[3];
17             myCounters[0] = new Counter("Counter 1");
18             myCounters[1] = new Counter("Counter 2");
19             myCounters[2] = myCounters[0];
20
21             for(int i=1; i<10; i++)
22             {
23                 myCounters[0].Increment();
24             }
25
26             for(int i=1; i<15; i++)
27             {
28                 myCounters[1].Increment();
29             }
30
31             PrintCounters(myCounters);
32             myCounters[2].Reset();
33             PrintCounters(myCounters);
34
35             Console.WriteLine("");
36             myCounters[2].ResetByDefault();
37             myCounters[2].Increment();
38             PrintCounters(myCounters);
39         }
40     }
41 }
42
```

13. Tell the Counter to increase the count value. Does the code still run without any bugs/crash? What is the reason behind? You can provide the answers as the comments in the code.

-The code will still run normally because the "unchecked" method will force the code runner to ignore the overflown value but the value of myCounter[2] will be negative, the reason behind this is because the maximum value that an integer can have in C# is 2,147,483,647 but the value assigned by the ResetByDefault method is bigger(2,147,483,647,473).

Program's output:

A screenshot of the Microsoft Visual Studio Debug Console window. The window has a title bar with the text "Microsoft Visual Studio Debug Console" and standard window controls (minimize, maximize, close). The console output shows the following text:

```
Counter 1 is 9
Counter 2 is 14
Counter 1 is 9
Counter 1 is 0
Counter 2 is 14
Counter 1 is 0

Counter 1 is -526
Counter 2 is 14
Counter 1 is -526

E:\COS20007\week2\Task2_2P\CounterTask\bin\Debug\net8.0\
CounterTask.exe (process 22316) exited with code 0.
To automatically close the console when debugging stops,
enable Tools->Options->Debugging->Automatically close t
he console when debugging stops.
Press any key to close this window . . .|
```