## Debug the Code: Holidays between Two Dates

You are assigned to **find and fix the bugs** in an existing piece of code, using the Visual Studio **debugger**. You should trace the program execution to find the lines of code that produce incorrect or unexpected results.

You are given a program (existing **source code**) that aims to **count the non-working days between two dates** given in format day.month.year (e.g. between **1.05.2015** and **15.05.2015** there are **5** non-working days – Saturday and Sunday).

### Sample Input and Output

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 1.05.2016  15.05.2015 | 5 | There are 5 non-working days (Saturday / Sunday) in this period: 1-May-2016, 7-May-2016, 8-May-2016, 14-May-2016, 15-May-2016 |
| 1.5.2016  2.5.2016 | 1 | Only 1 non-working day in the specified period: 1.05.2016 (Sunday) |
| 15.5.2020  10.5.2020 | 0 | The second date is before the first. No dates in the range. |
| 22.2.2020  1.3.2020 | 4 | Two Saturdays and Sundays:   * 22.02.2020 and 23.02.2020 * 29.02.2020 and 1.03.2020 |

You can **find the broken code** in the judge system: <https://judge.softuni.bg/Contests/204/Git-GitHub-Debugging-Searching-Lab>. It looks as follows:

|  |
| --- |
| HolidaysBetweenTwoDates.cs |
| using System;  using System.Globalization;  class HolidaysBetweenTwoDates  {  static void Main()  {  var startDate = DateTime.ParseExact(Console.ReadLine(),  "dd.m.yyyy", CultureInfo.InvariantCulture);  var endDate = DateTime.ParseExact(Console.ReadLine(),  "dd.m.yyyy", CultureInfo.InvariantCulture);  var holidaysCount = 0;  for (var date = startDate; date <= endDate; date.AddDays(1))  if (date.DayOfWeek == DayOfWeek.Saturday &&  date.DayOfWeek == DayOfWeek.Sunday) holidaysCount++;  Console.WriteLine(holidaysCount);  }  } |

### Hints

There are **4** **mistakes** in the code. You’ve got to **use the debugger** to find them and fix them. After you do that, submit your **fixed code in the judge contest**: <https://judge.softuni.bg/Contests/Practice/Index/204#0>.

## Search in Google: Double Palindromes

You are assigned to **write a program** that **prints the** **first 129 numbers** that are **palindromes** and their **binary representation** is also a **palindrome**. Some examples:

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Binary** | **Palindrome?** | **Binary Palindrome?** |
| **0** | 0 | yes | yes |
| **1** | 1 | yes | yes |
| 2 | 10 | yes | no |
| 633 | 1001111001 | no | yes |
| **717** | 1011001101 | yes | yes |
| 1023 | 1111111111 | no | yes |

### Sample Input and Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| *(no input)* | 0, 1, 3, 5, 7, 9, 33, 99, 313, 585, 717, 7447, 9009, 15351, 32223, 39993, 53235, 53835, 73737, 585585, 1758571, 1934391, 1979791, 3129213, 5071705, 5259525, 5841485, … |

### Hints

The idea is to **find** the numbers in **Internet**, and to create a program that **only prints them**. Search in Google in order to find the first 129 double palindromes. Maybe the phrase “**double palindrome**” is **not the right** term?