# Упражнения: Дебъгване и ред на изпълнение

## Умножаване на четни по нечетни

Създайте програма, която прочита цяло число и умножава сумата на всичките му нечетни цифри по сумата на всичките му четни цифри:

### **Примери**

|  |  |  |
| --- | --- | --- |
| Вход | Изход | Коментари |
| 12345 | 54 | 12345 има 2 четни цифри - 2 и 4.  Сумата им е 6.  Също така числото има 3 нечетни цифри - 1, 3 и 5. Сумата им е 9.  Умножаваме 6 по 9 и получаваме 54. |
| -12345 | 54 |  |

### **Подсказки**

1. Създайте метод с име, описващо предназначението му (например GetMultipleOfEvensAndOdds). Той трябва да има един целочислен параметър и целочислена връщана стойност. Този метод ще извиква два други метода:

Text

Description automatically generated

1. Създайте два други метода, всеки от които ще сумира четните или нечетните цифри
2. Опишете логиката за сумиране на четни цифри:

Text

Description automatically generated

1. Направете същото за метода, който ще сумира нечетните цифри
2. Като тествате решението, може да забележите, че то не работи за отрицателни числа. Проследете изпълнението на програмата стъпка по стъпка и поправете грешката (идейка: може да използвате Math.Abs())

## Дебъгване на кода: Почивни дни между две дати

Имате задачата да откриете и поправите грешките във вече написан програмен код, като използвате дебъгера на Visual Studio. За целта трябва да проследите изпълнението на програмата, за да откриете тези редове от кода ѝ, които пораждат неправилен или неочакван резултат.

Разполагате с програма (т.е. със съществуващ програмен код) който се опитва да преброи неработните дни между две дати подадени във формат ден.месец.година (например между 1.05.2015 и 15.05.2015 има 5 неработни дни - съботи и недели).

### **Примери**

|  |  |  |
| --- | --- | --- |
| Вход | Изход | Коментари |
| 1.05.2016  15.05.2016 | 5 | Има 5 неработни дни (съботи и недели) в този период: 1-May-2016, 7-May-2016, 8-May-2016, 14-May-2016, 15-May-2016 |
| 1.5.2016  2.5.2016 | 1 | Само 1 неработен ден: 1.05.2016 (неделя) |
| 15.5.2020  10.5.2020 | 0 | Втората дата е преди първата. Няма дати в този диапазон. |
| 22.2.2020  1.3.2020 | 4 | По две съботи и недели:   * 22.02.2020 и 23.02.2020 * 29.02.2020 и 1.03.2020 |

Можете да намерите неработещия програмен код в платформата, файл: Broken-Solutions-1.zip. Изглежда така:

|  |
| --- |
| 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);  }  } |

### **Подсказки**

Има 4 грешки в кода. Трябва да използвате дебъгера за да ги откриете и поправите. След като сте готови, изпратете поправеният от вас код в платформата.

## Price Change Alert

Имате за задача да преработите готов код, който работи без грешки, но не е форматиран както трябва.

Предоставената ви програма следи цените на стоки и дава информация за значимостта на всяка промяна в цената. Според това, доколко е съществена, има четири типа промени: без промяна (цената е същата като предишната), малка (разлика под прага на значимост), цената расте и цената намалява.

Можете да намерите кода за поправяне във файла Broken-Solutions-1.zip.

### **Вход**

* На първия ред получавате N - броят на цените
* На втория ред получавате прага на значимост
* На следващите N реда, получавате цените

### **Изход**

* Не отпечатвате нищо за първата цена
* Ако няма разлика от предишната цена, съобщението е: "NO CHANGE: {текуща цена}"
* При малка разлика: "MINOR CHANGE: {предишна цена} to {текуща цена} ({разлика}%)"
* При голяма разлика: "PRICE UP: {предишна цена} to {текуща цена} ({разлика}%)" or "PRICE DOWN: {предишна цена} to {текуща цена} ({разлика}%)"

Процентите трябва да са закръглени до втория знак след десетичната точка.

### **Примери**

|  |  |
| --- | --- |
| Вход | Изход |
| 3  0.1  10  11  12 | PRICE UP: 10 to 11 (10.00%)  MINOR CHANGE: 11 to 12 (9.09%) |
| 3  0.1  10  10  12 | NO CHANGE: 10  PRICE UP: 10 to 12 (20.00%) |

### **Подсказки**

1. Изтеглете програмния код и се запознайте с него
2. Заемете се първо с лошото форматиране на кода - махнете ненужните празни редове, вмъкнете навътре кода където и колкото е нужно
3. Коригирайте имената на параметрите на методите
4. Дайте и на методите подходящи имена