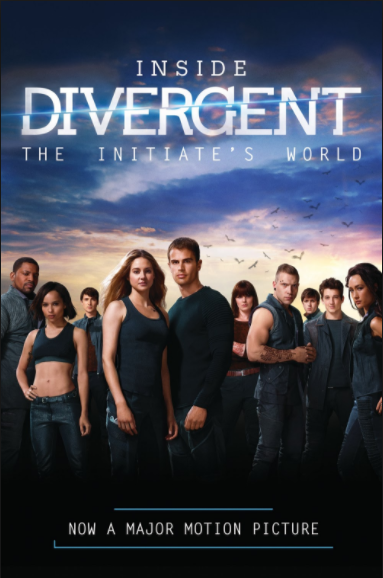
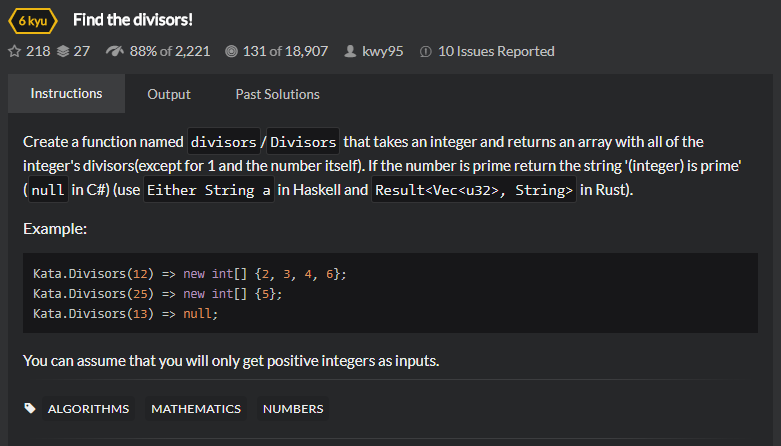
Day26. 找出分歧者不是啦，我是說要找因數 Codewars\_Find The Divisors



抱..抱歉貼錯

這才是今天的題目!



今天的題目需要找出輸入變數的因數，並且從小排序到大。

這個題目有特別註明不要找出自己跟1。

現在就來拆一下題目吧。

1. 找出因數(除以某數餘數為0)的方法。
2. 排序的方法。

一開始我們先來寫輸入0的測試吧!!

```

[TestMethod]

public void Input\_0\_Should\_Be\_Null()

{

Assert.IsNull(Kata.Divisors(0));

}

```

而Production Code 也就是老樣子會長成這個樣子

```

public static int[] Divisors(int num)

{

throw new System.**NotImplementedException**();

}  
```

老樣子，跑個測試，沒過很正常，紅燈，commit一下

接下來把Production Code改成這個樣子

```

public static int[] Divisors(int num)

{

return null;

}  
```

接下來跑個測試，PASS! Commit~

因為1、2的輸入都預期為null所以現在就來加上輸入4的測試。

當然，你也可以自己加上1跟2的測試。

```

[TestMethod]

public void Input\_4\_Should\_Be\_2()

{

CollectionAssert.AreEqual(new int[] { 2 }, Kata.Divisors(4));

}

```

因為這樣的話只需要考慮到輸出只有1個，所以Production Code長這樣

```

for (int i = 2; i <= num / 2; i++)

{

if (num % i == 0)

{

return new int[] { i };

}

}

```

再來是輸入6的測試，這個測試就會有多個元素的陣列了。

```

[TestMethod]

public void Input\_6\_Should\_Be\_2and3()

{

CollectionAssert.AreEqual(new int[] { 2, 3 }, Kata.Divisors(6));

}

```

因為要記錄因數的關係，所以需要用一個List來存取他，所以Production Code就會變成這個樣子

```

public static int[] Divisors(int num)

{

var result = new List<int>();

for (int i = 2; i <= num / 2; i++)

{

if (num % i == 0)

{

return new int[] { i };

result.Add(i);

}

}

return result.Count > 0 ? result.ToArray() : null;

}

```

這時候只要refactor一下，測試，Pass! Commit!

```

public static int[] Divisors(int num)

{

var result = Enumerable.Range(2, num / 2).Where(x => num % x == 0).ToArray();

return result.Length > 0 ? result : null;

}

```

現在已經完成所有需求啦!

所以現在就可以直接提交了。

但是Fail了，原因是因為運算時間的關係(他說我的Code寫太爛 嗚嗚)

所以我們回到剛才還沒有Refactor的地方，把演算法的問題考慮進去。

Production Code就變成這個樣子

```

public static int[] Divisors(int num)

{

var result = new List<int>();

for (int i = 2; i \* i <= num; i++)

{

if (num % i == 0)

{

result.Add(i);

result.Add(num / i);

}

}

return result.Count > 0 ? result.OrderBy(x => x).Distinct().ToArray() : null;

}

```

以下是今天所有的測試案例

```

[TestClass]

public class UnitTest1

{

[TestMethod]

public void Input\_0\_Should\_Be\_Null()

{

Assert.IsNull(Kata.Divisors(0));

}

[TestMethod]

public void Input\_4\_Should\_Be\_2()

{

CollectionAssert.AreEqual(new int[] { 2 }, Kata.Divisors(4));

}

[TestMethod]

public void Input\_6\_Should\_Be\_2and3()

{

CollectionAssert.AreEqual(new int[] { 2, 3 }, Kata.Divisors(6));

}

[TestMethod]

public void Input\_15\_Should\_Be\_3and5()

{

CollectionAssert.AreEqual(new int[] { 3, 5 }, Kata.Divisors(15));

}

[TestMethod]

public void Input\_16\_Should\_Be2and4and8()

{

CollectionAssert.AreEqual(new int[] { 2, 4, 8 }, Kata.Divisors(16));

}

[TestMethod]

public void Input\_253\_Should\_Be\_11and23()

{

CollectionAssert.AreEqual(new int[] { 11, 23 }, Kata.Divisors(253));

}

[TestMethod]

public void Input\_99999999\_SHould\_\_()

{

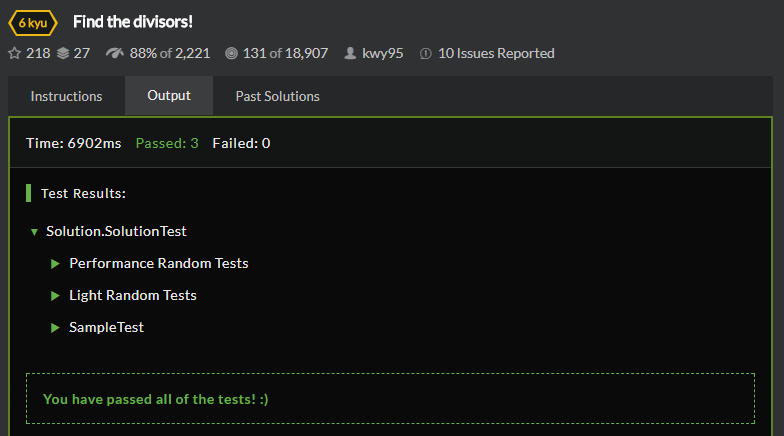
CollectionAssert.AreEqual(new int[] { 3, 9, 11, 33, 73, 99, 101, 137, 219, 303, 411, 657, 803, 909, 1111, 1233, 1507, 2409, 3333, 4521, 7227, 7373, 9999, 10001, 13563, 13837, 22119, 30003, 41511, 66357, 81103, 90009, 110011, 124533, 152207, 243309, 330033, 456621, 729927, 990099, 1010101, 1369863, 3030303, 9090909, 11111111, 33333333 }, Kata.Divisors(99999999));

}

}

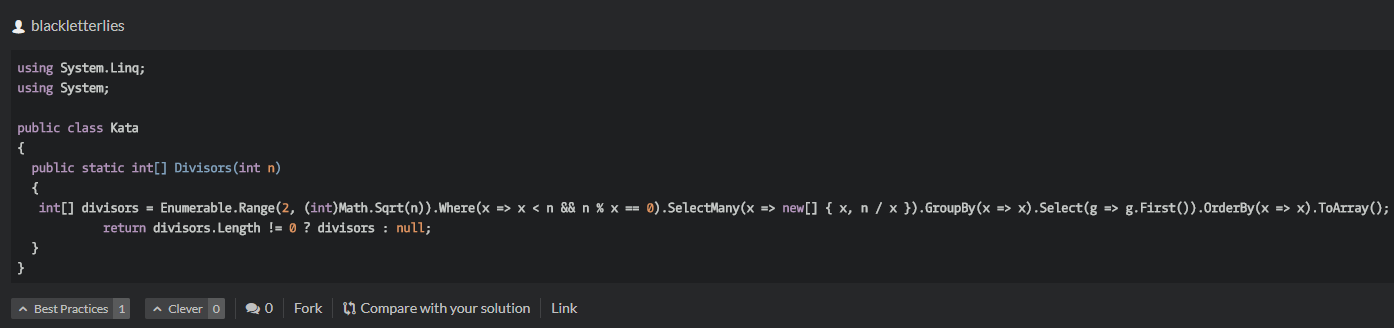
```

最後Submit到Codewars，Pass!



在Codewars上成功提交了~

來看一下其他人怎麼寫吧!



依然還是有人可以用一行Linq語法串來串去然後輸出的，超級厲害…

希望有朝一日我也能變成那個樣子XD

Git url :

<https://github.com/SQZ777/Codewars_FindTheDivisors>

Codewars Link:

<https://www.codewars.com/kata/find-the-divisors/train/csharp>

下一題，明天見!