Day21. 酷炫的兒童黑話 Codewars\_Simple Pig Latin

今天的題目超酷的啦(難度:5kyu)

看起來很炫泡，一開始看不懂什麼叫做Pig Latin到底是什麼毛

跑去google就發現他被稱作為兒童黑話

是一種英語語言的遊戲。

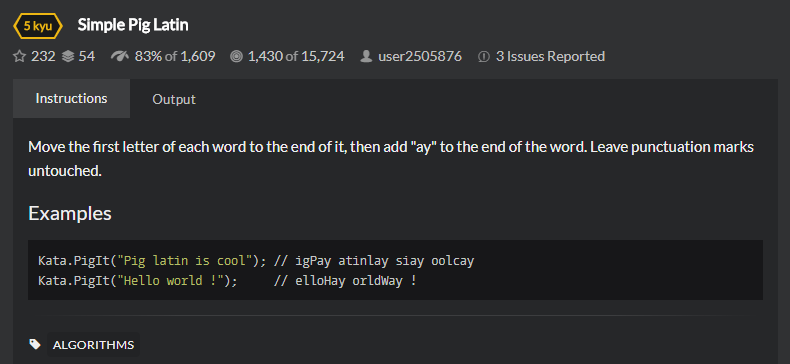
有興趣的就來這邊看一下維基百科怎麼解釋唄XD

<https://zh.wikipedia.org/wiki/%E5%85%92%E7%AB%A5%E9%BB%91%E8%A9%B1>



很像這個XD

今天的題目長這樣



今天的題目要去拆解輸入的字串，而每一個被拆解的字串要背去掉頭加在後方，並且再加上ay。

現在就來拆一下題目吧。

1. 將字串拆解->split方法
2. 去得字首加上ay方法
3. 將處理過的字串串起來->Join方法

一開始我們先來寫一個處理字首加上ay的方法吧!!

```

[TestMethod]

public void PigPart\_Input\_Empty\_Should\_Be\_Empty()

{

Assert.AreEqual(string.Empty,Kata.PigPart(string.Empty));

}

```

而Production Code 也就是老樣子會長成這個樣子

```

public static string PigPart(string s)

{

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改一下，用最簡單的方式解決他!

```

public static string PigPart(string s)

{

return string.Empty;

}  
```

接下來跑個測試，PASS! Commit~

再來寫處理加上ay字串的測試

```

[TestMethod]

public void PigPart\_Input\_a\_Should\_Be\_aay()

{

Assert.AreEqual("aay", Kata.PigPart("a"));

}

```

而Production Code就會長這個樣子

```

public static string PigPart(string s)

{

if (s.Length > 0)

{

return s + "ay";

}

return string.Empty;

}

```

接下來就可以補上字串長度為2的字串測試，就需要處理到字首的問題。

```

[TestMethod]

public void PigPart\_Input\_ab\_Should\_Be\_baay()

{

Assert.AreEqual("baay", Kata.PigPart("ab"));

}

```

Production Code就會考慮到處理字首的問題了。

```

public static string PigPart(string s)

{

if (s.Length > 0)

{

return s.Remove(0, 1) + s[0] + "ay";

}

return string.Empty;

}

```

再來就是重構之後Production Code可以變成的樣子

```

public static string PigPart(string s)

{

return s.Length > 0 ? s.Remove(0, 1) + s[0] + "ay" : string.Empty;

}

```

接下來就要開始寫主要Production Code的測試啦!

```

[TestMethod]

public void Input\_ab\_ba\_Should\_Be\_baay\_abay()

{

Assert.AreEqual("baay abay",Kata.PigIt("ab ba"));

}

```

這個測試涵蓋的是要將輸入的str用split方法之後使用PigPart方法處理之後再把字串Join起來

所以Production Code 就會長成這個樣子

```

public static string PigIt(string str)

{

var splitted = str.Split();

var result = new List<string>();

foreach (var s in splitted)

{

result.Add(PigPart(s));

}

return string.Join(" ", result);

}

```

然後Production Code可以做Linq的寫法，所以就變成這個樣子

```

public static string PigIt(string str)

{

return string.Join(" ", str.Split().Select(PigPart).ToList());

}

```

以下是所有的Production Code

```

public class Kata

{

public static string PigPart(string s)

{

return s.Length > 0 ? s.Remove(0, 1) + s[0] + "ay" : string.Empty;

}

public static string PigIt(string str)

{

return string.Join(" ", str.Split().Select(PigPart).ToList());

}

}

```

以下是今天所有的測試案例

```

[TestClass]

public class UnitTest1

{

[TestMethod]

public void PigPart\_Input\_Empty\_Should\_Be\_Empty()

{

Assert.AreEqual(string.Empty, Kata.PigPart(string.Empty));

}

[TestMethod]

public void PigPart\_Input\_a\_Should\_Be\_aay()

{

Assert.AreEqual("aay", Kata.PigPart("a"));

}

[TestMethod]

public void PigPart\_Input\_ab\_Should\_Be\_baay()

{

Assert.AreEqual("baay", Kata.PigPart("ab"));

}

[TestMethod]

public void Input\_ab\_ba\_Should\_Be\_baay\_abay()

{

Assert.AreEqual("baay abay", Kata.PigIt("ab ba"));

}

[TestMethod]

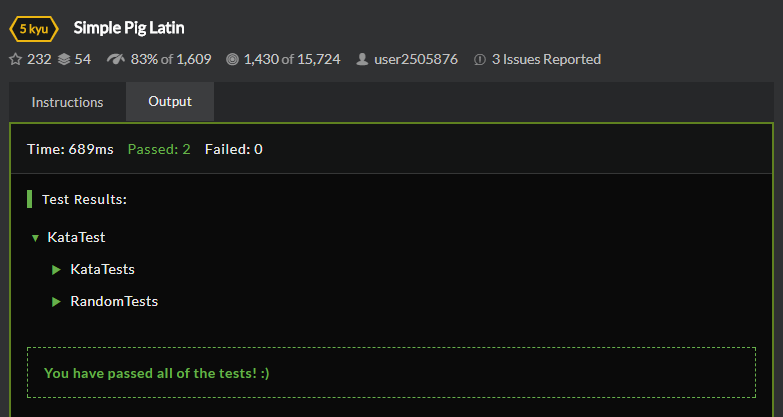
public void Input\_Hello\_World\_Should\_Be\_elloHay\_orldWay()

{

Assert.AreEqual("elloHay orldWay",Kata.PigIt("Hello World"));

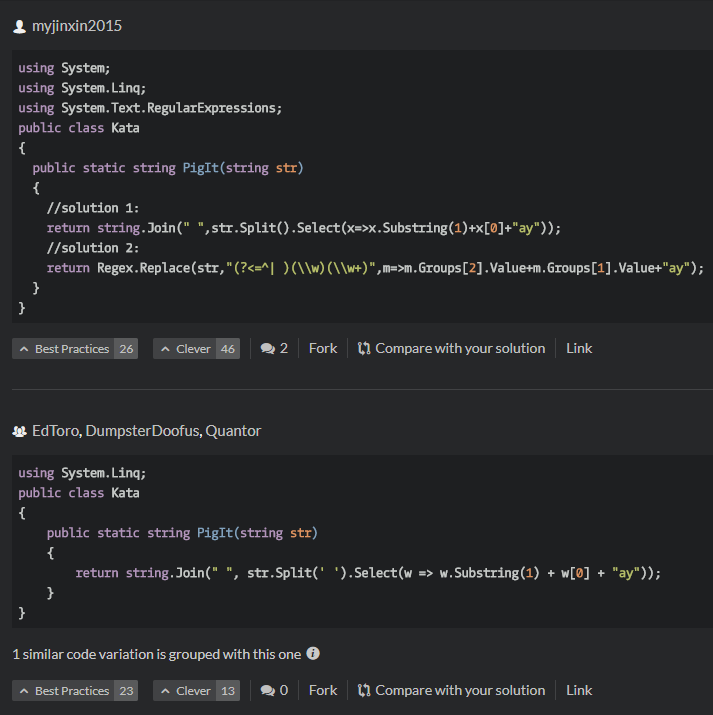
}

```



在Codewars上成功提交了~

來看一下其他人怎麼寫吧!



第一個人寫得好像就是正規表示式…

小弟我對那個沒研究，覺得很酷XD

或許有時間可以研究一下正規表示法要怎麼使用呢

第二個人寫的則是跟我一樣的邏輯

只是我把它去頭加上ay的方法抽出來了

Git url :

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

Codewars Link:

<https://www.codewars.com/kata/simple-pig-latin/train/csharp>

下一題，明天見!