

m05 UF2 P2 Fix Project

By YOUR NAME HERE

1.- Input Output (0.5pts)	2
Solution (0.25pts)	2
Explanation (0.25pts)	2
2.- Calculator (0.75pts)	3
Solution (0.25pts)	3
Explanation (0.5pts)	3
3.- Factorial (1pt)	4
Solution (0.5pts)	4
Explanation (0.5pts)	4
4.- Replace (1pt)	5
Solution (0.5pts)	5
Explanation (0.5pts)	5
5.- ParenthesisChecker (1pt)	6
Solution (0.5pts)	6
Explanation (0.5pts)	6
6.- Palindrome (1pt)	7
Solution (0.5pts)	7
Explanation (0.5pts)	7
7.- Primes (1pt)	8
Solution (0.5pts)	8
Explanation (0.5pts)	8
8.- DNI (1.25pts)	9
Solution (0.75pts)	9
Explanation (0.5pts)	9
9.- DayOfTheWeek (1.25pts)	10
Solution (0.75pts)	10
Explanation (0.5pts)	10
10.- Sorting (1.25pts)	11
Solution (0.5pts)	11
Theory (0.75pts)	11

1.- Input Output (0.5pts)

Solution (0.25pts)

Code

```
13 string response = Console.ReadLine().ToLower();
14 if (response[0] == 'y')
15 {
16     Console.WriteLine("Hello world");
17     running = false;
18 }
19 else if (response[0] == 'n')
20 {
21     Console.WriteLine("Goodbye world");
22     running = false;
23 }
```

Explanation (0.25pts)

Este error es muy sencillo de solucionar, si nos fijamos el propio Visual Studio nos marca el error en la línea 19 en “else if (response[0] = 'n')” y si nos fijamos arriba en la línea 14 tenemos exactamente lo mismo, pero falta algo que es lo que hace que este bien y es que falta un =. Por lo tanto quedaría tal que así: “else if (response[0] == 'n')”.

2.- Calculator (0.75pts)

Solution (0.25pts)

Code

```

20     }
21     [TestMethod]
22     public void Subtract_Test()
23     {
24         //Arrange
25         float A = (float)Utils.random.NextDouble();
26         float B = (float)Utils.random.NextDouble();
27         float result = 0;
28         float resultOK = A - B;
29         //Act
30         result = Calculator.Program.Subtract(A, B);
31         //Assert
32         Assert.AreEqual(resultOK, result);
33     }

```

```

61     public void Mod_Test()
62     {
63         //Arrange
64         float A = (float)Utils.random.NextDouble();
65         float B = (float)Utils.random.NextDouble();
66         float result = 0;
67         float resultOK = A % B;
68         //Act
69         result = Calculator.Program.Mod(A, B);
70         //Assert
71         Assert.AreEqual(resultOK, result);
72     }

```

Test results

✓ Calculator_Tests (5)	9 ms
✓ Add_Test	3 ms
✓ Divide_Test	< 1 ms
✓ Mod_Test	3 ms
✓ Multiply_Test	< 1 ms
✓ Subtract_Test	3 ms

Explanation (0.5pts)

En el mod lo que había que hacer era cambiar el orden de los factores ya que estaba escrito "float resultOK = B % A;" y para que funcione hay que poner "float resultOK = A % B;"; y en substract había que cambiar el factor B por el factor A escrito "float resultOK = B - B;" y hay que poner "float resultOK = A - B;".

3.- Factorial (1pt)

Solution (0.5pts)

Code

```
39 2 referencias | 1/1 pasando  
40 public static float average(List<float> numbers)  
41 {  
42     float sum = 0;  
43     for (int i = 0; i < numbers.Count; i++)  
44     {  
45         sum += numbers[i];  
46     }  
    return sum / numbers.Count;
```

Test results

✓ Factorial_Tests (3)	3 ms
✓ Average_Test	3 ms
✓ For_Test	< 1 ms
⚠ Recursive_Test	

Explanation (0.5pts)

Tras dar muchas vueltas he conseguido hacer que el código funcione, el error lo he encontrado en la línea 46 donde tras el "return sum" debemos añadir "/" numbers.Count" quedando tal que así "return sum / numbers.Count;".

4.- Replace (1pt)

Solution (0.5pts)

Code

```
2 referencias | 1/1 pasando
19 public static string Replace(string text, char a, char b)
20 {
21     StringBuilder sb = new StringBuilder();
22     for (int i = 0; i < text.Length; i++)
23     {
24         if(text[i] == a)
25         {
26             sb.Append(b);
27         }
28         else
29         {
30             sb.Append(text[i]);
31         }
32     }
33     return sb.ToString();
34 }
35 }
```

Test results

✓ Replace_Tests (1)	18 ms
✓ Replace_Test	18 ms

Explanation (0.5pts)

El cambio que hay que hacer es sencillo, tan solo consiste en cambiar en el código la línea 26 y sustituir "sb.Append(a);" por "sb.Append(b);" (cambiar la letra a por b para que así reemplace el carácter original).

5.- ParenthesisChecker (1pt)

Solution (0.5pts)

Code

```

13     if(open == close)
14     {
15         Console.WriteLine("The parenthesis close");
16     }
17     else
18     {
19         Console.WriteLine("The parenthesis don't close");
20     }
21 }

22 public static int CountChars(string text, char search)
23 {
24     int amount = 0;
25     for (int i = 0; i < text.Length; i++)
26     {
27         if(text[i] == search)
28             amount++;
29     }
30     return amount;
31 }
32

```

Test results

✓ ParenthesisChecker_Tests (1)	16 ms
✓ ParenthesisChecker_Test	16 ms

Explanation (0.5pts)

Lo primero que he observado al usar el código sin modificarlo es que cuando cerraba el paréntesis decía que no estaba cerrado y viceversa por lo que los he cambiado el uno por el otro.

Y el segundo cambio que he hecho ha sido en la línea 27 cambiar "if(text[i] != search)" por "if(text[i] == search)", un cambio que consiste en cambiar "!=" por "==".

6.- Palindrome (1pt)

Solution (0.5pts)

Code

```
2 referencias | 1/1 pasando
14 public static string Palindrome(string text)
15 {
16     StringBuilder sb = new StringBuilder();
17
18     sb.Append(text);
19     for (int i = text.Length - 1; i >= 0; i--)
20     {
21         sb.Append(text[i]);
22     }
23
24     return sb.ToString();
25 }
```

Test results

✓ Palindrome_Tests (1)	18 ms
✓ Palindrome_Test	18 ms

Explanation (0.5pts)

Sin tocar el código, la consola imprime la misma palabra que le introduces tal cual, por lo tanto hay que cambiarlo y que lo escriba al revés. Para ello he cambiado en la línea 19 “(int i = 0; i < text.Length; i++)” por “for (int i = text.Length - 1; i >= 0; i--)” para que así lo imprima al revés y haga el efecto Palíndromo.

7.- Primes (1pt)

Solution (0.5pts)

Code

Answer here

Test results

Answer here

Explanation (0.5pts)

Este ejercicio por muchas vueltas que le he dado no he conseguido resolverlo, no encuentro la manera de como hacerlo.

8.- DNI (1.25pts)

Solution (0.75pts)

Code

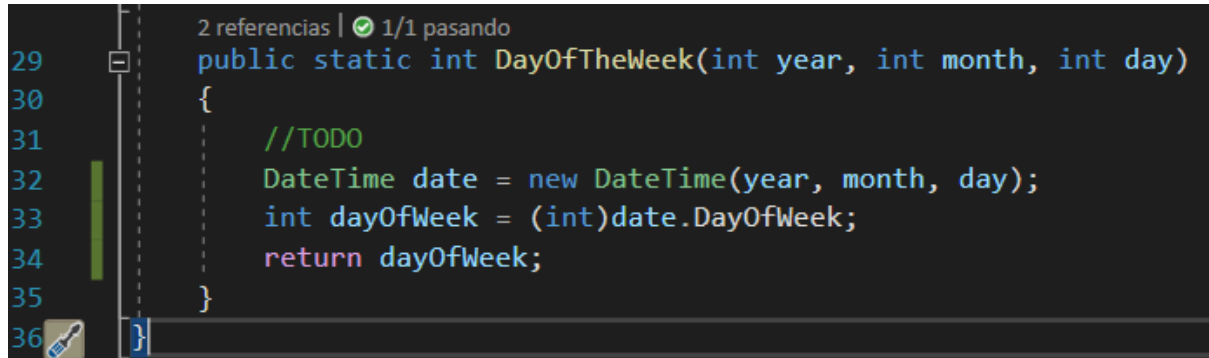
```
19 public static char LetterNIF(int number)
20 {
21     string LetterNIF = "TRWAGMYFPDXBNJZSQVHLCKE";
22     int resto = number % 23;
23     return LetterNIF[resto];
24 }
```

Explanation (0.5pts)

En este ejercicio he tenido problemas ya que no me sale en el test de prueba, por lo tanto no se si lo he hecho bien. Lo que he hecho ha sido decirle todas las letras posibles que puede llevar un DNI y luego le pido que entre las 23 letras que le hemos dicho antes que elija la correcta. Tampoco me deja imprimir por consola para comprobarlo por lo que no se si esta bien.

9.- DayOfTheWeek (1.25pts)

Solution (0.75pts)



```
29 2 referencias | 1/1 pasando
30 public static int DayOfTheWeek(int year, int month, int day)
31 {
32     //TODO
33     DateTime date = new DateTime(year, month, day);
34     int dayOfWeek = (int)date.DayOfWeek;
35     return dayOfWeek;
36 }
```

Answer here

Test results

✓ DayOfTheWeek_Tests (1)	18 ms
✓ DayOfTheWeek_Test	18 ms

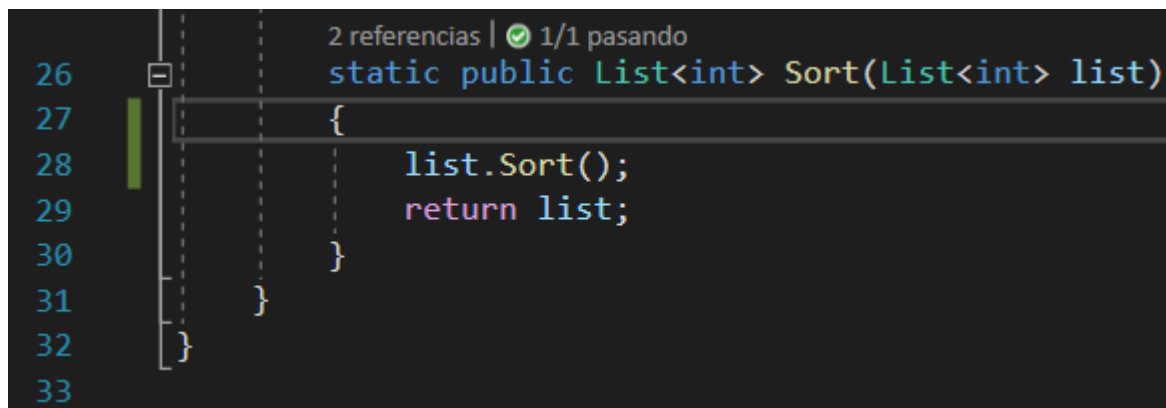
Explanation (0.5pts)

El código de base venía en el final con un "return -1;", el cual no tenía ningún sentido; si nos fijamos arriba pone "//TODO", es decir, que nos está dando la solución ya que debemos de pedirle todo. Por ello le pedimos el año, mes y día, y luego el día de la semana que le corresponde a la fecha introducida y finalmente le pedimos que nos devuelva ese día de la semana al que corresponde.

10.- Sorting (1.25pts)

Solution (0.5pts)

Code



```
26 static public List<int> Sort(List<int> list)
27 {
28     list.Sort();
29     return list;
30 }
31 }
32 }
33 }
```

Test results

Sorting_Tests (1)	16 ms
Sorted_Test	16 ms

Theory (0.75pts)

He buscado varios métodos en internet y no sabía cómo implementarlo pero al final con el más sencillo se ordenaba de forma ascendente (menor a mayor). Tan solo hay que añadir en la línea 28 "list.Sort();" que lo que hace es llamar a la función lista y ordenar los números de forma ascendente como he dicho anteriormente y "return list;" (línea 29) nos la devuelve ya ordenada.

Función sacada de internet en el siguiente enlace:

"https://www.w3schools.com/python/ref_list_sort.asp".