



AUTo4_o7

Jesús González Díaz | 1º DAW-B | 04/03/2024

Ejercicio 1	2
Ejercicio 2	3
Ejercicio 3	4

Ejercicio 1

Esta función calcula la media de los valores en el array y luego cuenta cuántos elementos del array son menores que esa media.

```
1 referencia | 1/1 pasando
public static int numMenosMedia(int[] array)
{
    int suma = 0;
    int resultadoMedia;
    int bajoMedia = 0;

    for (int i = 0; i < array.Length; i++)
    {
        suma = suma + array[i];
    }

    resultadoMedia = suma / (array.Length - 1);

    for (int j = 0; j < array.Length; j++)
    {
        if (array[j] < resultadoMedia)
        {
            bajoMedia++;
        }
    }

    return bajoMedia;
}
```

```
[TestMethod]
0 referencias
public void probarNumBajoMedia()
{
    int[] array = { 1, 5, 5, 5, 5, 5, 5, 5, 5 };
    int resultado = OperacionesMates.numMenosMedia(array);
    Assert.AreEqual(0, resultado);
}
```

Ejercicio 2

Esta función verifica si el array está ordenado de forma descendente, es decir, si cada elemento es menor o igual que el siguiente.

```
1 referencia | 1/1 pasando
public static bool estaOrdenada(int[] array)
{
    for (int i = 1; i < array.Length; i++)
    {
        if (array[i] < array[i - 1])
        {
            return false;
        }
    }
    return true;
}
```

```
[TestMethod]
0 | 0 referencias
public void probarEstaOrdenada()
{
    int[] array = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
    bool resultado = OperacionesMates.estaOrdenada(array);
    Assert.IsTrue(resultado);
}
```

Ejercicio 3

Esta función verifica si una contraseña cumple con ciertas condiciones:

- Tiene entre 4 y 6 caracteres.
- Tiene al menos una minúscula y una mayúscula.
- Contiene al menos un dígito.
- No contiene la palabra "Paz".

```
7 referencias | 6/6 pasando
public static bool ComprobarPassword(string pass)
{
    if (pass.Length < 4 || pass.Length > 6)
    {
        return false;
    }

    bool tieneMayuscula = false;
    bool tieneMinuscula = false;
    bool tieneDigito = false;

    foreach (char c in pass)
    {
        if (char.IsUpper(c))
        {
            tieneMayuscula = true;
        }
        else if (char.IsLower(c))
        {
            tieneMinuscula = true;
        }
        else if (char.IsDigit(c))
        {
            tieneDigito = true;
        }
    }

    if (!tieneMayuscula || !tieneMinuscula || !tieneDigito)
    {
        return false;
    }

    if (pass.ToLower().Contains("paz"))
    {
        return false;
    }

    return true;
}
```

```

[TestMethod]
0 referencias
public void probarPasswordValida()
{
    string password = "Abc123";
    bool result = OperacionesMates.ComprobarPassword(password);
    Assert.IsTrue(result);
}

[TestMethod]
0 referencias
public void probarPasswordLarga()
{
    string password = "Abcd12345";
    bool result = OperacionesMates.ComprobarPassword(password);
    Assert.IsFalse(result);
}

0 referencias
public void probarPasswordCorta()
{
    string password = "Ab1";
    bool result = OperacionesMates.ComprobarPassword(password);
    Assert.IsFalse(result);
}

[TestMethod]
0 referencias
public void probarContraseñaSinMinuscula()
{
    string password = "ABC123";
    bool result = OperacionesMates.ComprobarPassword(password);
    Assert.IsFalse(result);
}

[TestMethod]
0 referencias
public void probarContraseñaSinMayuscula()
{
    string password = "abc123";
    bool result = OperacionesMates.ComprobarPassword(password);
    Assert.IsFalse(result);
}

[TestMethod]
0 referencias
public void probarContraseñaSinDigito()
{
    string password = "Abcdef";
    bool result = OperacionesMates.ComprobarPassword(password);
    Assert.IsFalse(result);
}

[TestMethod]
0 referencias
public void probarContraseñaConPaz()
{
    string password = "Paz213";
    bool result = OperacionesMates.ComprobarPassword(password);
    Assert.IsFalse(result);
}

```

Explorador de pruebas

Serie de pruebas finalizada: 8 pruebas (Superadas: 8; Con errores: 0; Omitidas: 0) ejecutadas en 99 ms

0 advertencias 0 errores

Prueba	Duración	Rasgo
ETS_Testing (8)	30 ms	
ETS_Testing (8)	30 ms	
UnitTest1 (8)	30 ms	
probarContraseñaConPaz	30 ms	
probarContraseñaSinDigito	< 1 ms	
probarContraseñaSinMayusc...	< 1 ms	
probarContraseñaSinMinusc...	< 1 ms	
probarEstaOrdenada	< 1 ms	
probarNumBajoMedia	< 1 ms	
probarPasswordLarga	< 1 ms	
probarPasswordValida	< 1 ms	

Resumen de los detalles de la prueba

probarContraseñaSinMinuscula

Origen: [UnitTest1.cs](#) línea 47

Duración: < 1 ms