

CFGS

DESARROLLO DE APLICACIONES MULTIPLATAFORMA Y DESARROLLO DE APLICACIONES WEB

M03A: Programación.

Unidad Formativa 1: Programación estructurada.

UF1_PAC03_ENUNCIADO



UF1. [PAC 03]. Programación estructurada.

INFORMACIÓN

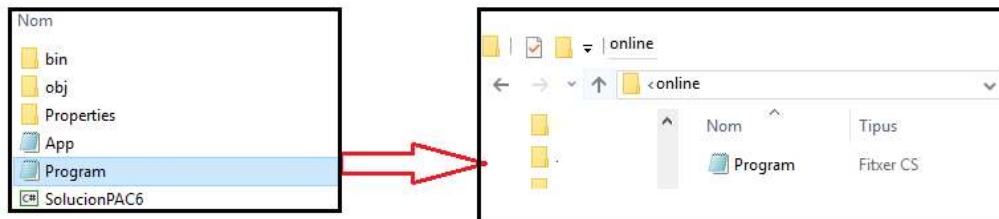
Para responder a las siguientes cuestiones deberás ayudarte del material didáctico y de internet.

Requisitos varios que deben cumplirse en vuestros trabajos:

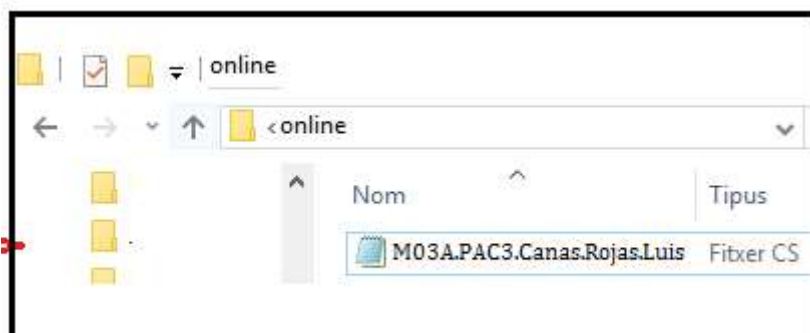
- En los ejercicios, si se requieren de cálculos, estos deben aparecer en la respuesta que planteéis.
- Siempre que utilicéis información de Internet para responder / resolver alguna pregunta, tenéis que citar la fuente (la página web) de dónde habéis sacado aquella información.
- Siempre que utilicéis información del material didáctico para responder / resolver alguna pregunta, tenéis que citar el tema y la página de dónde habéis sacado aquella información.
- No se aceptarán respuestas sacadas de Internet utilizando la metodología de copiar y pegar. Podéis utilizar Internet para localizar información, pero el redactado de las respuestas ha de ser vuestro.
- Las respuestas a las preguntas deben estar bien argumentadas, no se admiten respuestas escuetas o monosílabas.
- Se valorará la presentación de vuestro trabajo hasta con un punto y medio **(1)** de la nota final.

Instrucciones UF1 PAC3

- La fecha de entrega es improporrogable y la tenéis especificada en el campus.
- **La PAC se debe realizar en C#**
- En la primera línea del fichero, debe aparecer vuestro nombre, como comentario.
- Para hacer la entrega, cuando tengáis la PAC terminada, debéis hacer lo siguiente:
 - Copiar el fichero Program.cs de vuestro proyecto, a donde queráis, por ejemplo a una carpeta que se llame "online":



- Cambiar el nombre del fichero Program.cs, al siguiente: **M03A.PAC3.Apellido1.Apellido2.Nombre**. Los apellidos y nombres se escriben sin acentos, espacios, eñes....Por ejemplo, el alumno Luis Cañas Rojas, debe usar el siguiente nombre para el fichero de la PAC3: M03A.PAC3.Canas.Rojas.Luis.



- Éste es el fichero que debéis subir al campus.

Evaluación genérica de los ejercicios

Los ejercicios entregados deben **compilar** para ser evaluados. Para cada ejercicio se verificará todo lo que se expone a continuación.

- **Funcionalidad(65%):** El programa debe mostrar resultados correctos para entradas razonables. Haced pruebas en casos "críticos" o "fronterizos". Por ejemplo, si el programa debe pedir la edad del usuario, obviamente introduciré un número cuando yo lo pruebe(no letras), pero... qué pasa si entro el -7? tiene sentido?
- **Control de errores(10%):** El programa debe controlar tantos errores como sea posible, e indicar al usuario porque ha fallado, y, en la medida de lo posible, darle la posibilidad de rectificar el error.
- **Comentarios (10%):** El código debe ser comentado, explicando la manera que habéis usado para programar vuestra solución al problema.
- **Elegancia y eficiencia de la solución(15%):** el programa no debe hacer operaciones que malgasten tiempo de cálculo. El código debe estar bien estructurado, con nombres de variables claros.

Prohibiciones

- **No se pueden usar ni arrays, ni funciones ni ficheros**

Enunciado UF1 PAC3

Cuando Anduriña, emigrante gallega, descubrió hace algunos meses que acababan de abrir un cocedero de mariscos en su barrio, no pudo evitar dar saltos de alegría. Desde aquel momento no hay semana que no pase por allí a comprar algo de marisco para recordar, junto a otros amigos, el sabor de su tierra.

Pero últimamente sospecha que los dueños del cocedero no son trigo limpio. Cuando prepara sus famosos mejillones en salsa, muchas veces se encuentra que al ir a comer uno está solo la concha. Al principio pensaba que el mejillón se habría soltado y alguno de sus invitados tendría la suerte de comer dos en uno. Pero tras semanas quejándose de su mala suerte se ha planteado que quizá no sea tal. Es posible que en el cocedero estén, intencionadamente, quedándose con algunos mejillones y metiendo conchas vacías. ¡Es el negocio redondo! Cobran conchas a precio de mejillón, y éstos los venden al doble de precio, sin la concha, por otro lado.

Para poner a prueba su teoría, las próximas semanas va a anotar cuántos mejillones se comen realmente con cada concha, sabiendo que al cocinarlos siempre quita una de las dos conchas del mejillón.

Entrada

La entrada comienza con un número que indica cuántos días ha comprado mejillones Anduriña para comprobar si hay o no engaño. Por cada uno, aparecen cuántos mejillones había en cada concha. La lista acaba con un -1 indicando que no quedan más (es decir, el -1 indica que se ha terminado el caso de prueba, y el programa debe dar una respuesta, y pasar al siguiente caso de prueba). Solo se debe controlar que la entrada sea correcta en el número de casos.

Salida

Por cada día de su experimento, el programa generará una línea dando su veredicto. Escribirá "Justo" si el número de conchas era el mismo que el de mejillones, "Suerte" si había más mejillones que conchas y "Timo" si, como sospecha, había menos mejillones que conchas y hay algo turbio en el cocedero. Al finalizar el programa, se mostrará un mensaje indicando cuantos días han sido Justos, cuantos ha tenido Suerte, y cuántos le han timado.

Ejemplo ejecución:

ENTRADA	SALIDA
4 (número de casos de prueba)	
0 2 1 2 1 1 -1	SUERTE
0 1 -1	TIMO
1 1 -1	JUSTO
2 1 -1	SUERTE
	JUSTO - - > 1 SUERTE - - > 2 TIMO - - > 1

¡Buen trabajo!

