

[Página Principal](#) / [Mis cursos](#) / [131_15021482_ZSIFC02_MP0485_B](#) / [Examen tercera evaluación](#) / [Examen tercera evaluación 2023 \(copia\)](#)

Pregunta **1**

Sin finalizar

Puntúa como 1,50

Trabajar con bytes en fichero bmp

Asociado automáticamente a la pregunta hay un fichero bmp que puedes direccionar con

File archivo = new File("unaimagen.bmp");

La estructura en bytes de un fichero bmp es la siguiente

Bytes	Información
0, 1	Tipo de fichero "BM"
2, 3, 4, 5	Tamaño del archivo
6, 7	Reservado
8, 9	Reservado
10, 11, 12, 13	Inicio de los datos de la imagen
14, 15, 16, 17	Tamaño de la cabecera del bitmap
18, 19, 20, 21	Anchura (píxels)
22, 23, 24, 25	Altura (píxels)
26, 27	Número de planos
28, 29	Tamaño de cada punto
30, 31, 32, 33	Compresión (0=no comprimido)
34, 35, 36, 37	Tamaño de la imagen
38, 39, 40, 41	Resolución horizontal
42, 43, 44, 45	Resolución vertical
46, 47, 48, 49	Tamaño de la tabla de color
50, 51, 52, 53	Contador de colores importantes

Se quiere consultar el tamaño de archivo y su dimensión en píxeles para obtener la información como se observa en el test ejemplo. Lógicamente la respuesta sólo se considera correcta si los datos se recuperan de la cabecera del fichero.

Por ejemplo:

Resultado
El tamaño del fichero BMP es: 818058 Kbytes La anchura de la imagen BMP es: 640 píxeles La altura de la imagen BMP es: 426 píxeles

Respuesta: (sistema de penalización: 0 %)

1 ||

Comprobar

Pregunta **2**

Sin finalizar

Puntúa como 1,50

Ordenando artículos

Completa el código de forma que se listamos todos los artículos cuya cantidad es menor de 26 unidades. El listado debe de estar ordenado por cantidad, en caso de empate de cantidad por precio, y en caso de empate de cantidad y precio por nombre.

Ten en cuenta que una data class provee automáticamente a la clase de un toString() sobrescrito que es el que se usa en la impresión de los tests (no tienes que escribirlo tú, por lo tanto).

```
data class Artículo(val id: Int, val nombre: String, val precio: Double, val cantidad: Int)
val listaArticulos= listOf(
    Artículo(1, "Coca Cola", 8.0, 10),
    Artículo(2, "Pepsi", 8.0, 10),
    Artículo(3, "Fanta", 7.0, 10),
    Artículo(4, "Sprite", 6.0, 20),
    Artículo(5, "Manzanita", 5.0, 25),
    Artículo(6, "7up", 6.0, 30),
    Artículo(7, "Mirinda", 10.0, 25),
)

fun main() {
    //Ordenar e imprimir
}
```

ENVÍA: el código anterior pero incluyendo la ordenación e impresión en main

Por ejemplo:

Resultado
Artículo(id=3, nombre=Fanta, precio=7.0, cantidad=10) Artículo(id=1, nombre=Coca Cola, precio=8.0, cantidad=10) Artículo(id=2, nombre=Pepsi, precio=8.0, cantidad=10) Artículo(id=4, nombre=Sprite, precio=6.0, cantidad=20) Artículo(id=5, nombre=Manzanita, precio=5.0, cantidad=25) Artículo(id=7, nombre=Mirinda, precio=10.0, cantidad=25)

Respuesta: (sistema de penalización: 0 %)

1 ||

Comprobar

Pregunta **3**

Sin finalizar

Puntúa como 3,50

FISGEITOR

La empresa de seguridad "MuySeguris" ha desarrollado un sistema de alertas de emergencia llamado "EmergenciaPro". Este sistema permite registrar diferentes tipos de unidades de emergencia para notificarlas en caso de detectar una alerta.

El sistema "EmergenciaPro" utiliza el patrón Observer para notificar a las unidades de emergencia registradas sobre diferentes tipos de alertas: Alerta Roja y Alerta Naranja. Cada tipo de alerta tiene un conjunto específico de unidades de emergencia registradas.

Si te fijas en el test, hay dos tipos de alertas: naranjas y rojas. Si el código de la alerta comienza por "Naranja" se trata de una alerta Naranja, en caso contrario será alerta roja ya que suponemos siempre datos correctos. Una vez comprobado el tipo de la alerta se imprime un mensaje que describe la alarma y a continuación se avisa a los observadores registrados en ese tipo de alarma que cuando reciben el aviso imprimen "que van en camino".

De los test y la explicación, se deduce que Fisgeitor es la clase observada (u observable, como quieras describirla).

Las clases observadoras para simplificar las escribimos sin atributos, exceptuando la clase Psicologos que deben de tener un atributo de tipo String llamado *nombreGabinete* que identifica el grupo de psicologos asignados a la emergencia. Como aprecias en el test, este atributo se inicializa a través del argumento del constructor.

SE ENVIA: Clases necesarias que permitan cumplir los test, También necesitaras añadir los import que utilizan tus clases.

Por ejemplo:

Test	Resultado
<pre>Fisgeitor fisgeitor = new Fisgeitor(); UnidadEmergencia policia = new Policia(); UnidadEmergencia ambulancia = new Ambulancia(); UnidadEmergencia bomberos = new Bomberos(); UnidadEmergencia geos = new Geos(); UnidadEmergencia psicologos = new Psicologos("TeEscuchamos"); // Registrar unidades de emergencia en la alerta naranja fisgeitor.registrarUnidadEmergenciaNaranja(policia); fisgeitor.registrarUnidadEmergenciaNaranja(ambulancia); fisgeitor.registrarUnidadEmergenciaNaranja(bomberos); // Registrar unidades de emergencia en la alerta roja fisgeitor.registrarUnidadEmergenciaRoja(policia); fisgeitor.registrarUnidadEmergenciaRoja(ambulancia); fisgeitor.registrarUnidadEmergenciaRoja(bomberos); fisgeitor.registrarUnidadEmergenciaRoja(geos); fisgeitor.registrarUnidadEmergenciaRoja(psicologos); // Simular detección de alertas fisgeitor.generarAlerta("Naranja3k001"); fisgeitor.generarAlerta("Roja05001");</pre>	<pre>----- Se ha detectado una alerta Naranja3k001. Policía en camino. Ambulancia en camino. Bomberos en camino. ----- ¡¡peligro!! alerta Roja05001. Policía en camino. Ambulancia en camino. Bomberos en camino. Geos en camino. Psicologos en camino.</pre>
<pre>Fisgeitor fisgeitor = new Fisgeitor(); UnidadEmergencia policia = new Policia(); UnidadEmergencia ambulancia = new Ambulancia(); UnidadEmergencia bomberos = new Bomberos(); UnidadEmergencia geos = new Geos(); UnidadEmergencia psicologos = new Psicologos("TeEscuchamos"); fisgeitor.registrarUnidadEmergenciaRoja(policia); fisgeitor.registrarUnidadEmergenciaRoja(ambulancia); fisgeitor.registrarUnidadEmergenciaRoja(bomberos); fisgeitor.registrarUnidadEmergenciaRoja(geos); fisgeitor.registrarUnidadEmergenciaRoja(psicologos); // Eliminar una unidad de emergencia de la alerta roja Psicologos psicologos2 = new Psicologos("TeEscuchamos"); fisgeitor.eliminarUnidadEmergenciaRoja(psicologos2); // Simular detección de alerta después de eliminar una unidad fisgeitor.generarAlerta("Roja9H007");</pre>	<pre>----- ¡¡peligro!! alerta Roja9H007. Policía en camino. Ambulancia en camino. Bomberos en camino. Geos en camino.</pre>

Respuesta: (sistema de penalización: 0 %)

1 ||

Comprobar

Pregunta **4**

Sin finalizar

Puntúa como 3,50

Pulpo a fartar

Este año en mi pueblo estamos de aniversario, es la XXX *Feira do pulpo*. Para celebrar los 30 añitos del ala, se va invitar a la gente a comer *pulpo a fartar*. Bueno, es una forma de hablar, en realidad se podrá comer pulpo gratis pero con cierto orden. La gente que viene a la feria, a través de un complicado sistema de tickets, se organiza en colas. Cada cola de personas comensales se asigna a un caldeiro de pulpo. Una diestra pulpeira va picando en tiempo real el pulpo y entregando cada ración picada al primero de la cola.

Nos centramos aquí exclusivamente en lo que pasa con una cola de un caldeiro. De cada caldeiro se obtienen n platos de pulpo. La cola es fija en el sentido que siempre está formada por las mismas personas. Cuando una persona se le asigna un plato de pulpo se vuelve a poner de último en la cola y se va comiendo el pulpo a toda velocidad y de pie para seguir progresando en la cola de espera con la esperanza de conseguir un plato adicional. Osea, que la cola funciona circularmente hasta que del caldeiro ya no salen más platos y se acaba el funcionamiento de la cola..

Las pulpeiras nunca mezclan en el mismo plato patas y cabezas. Al que le toca un plazo de cabeza de pulpo, mala suerte y haber si tiene mejor suerte para la siguiente. Lo que pretendemos aquí es obtener una lista ordenada ascendentemente por nombre de todos los comensales de la cola que al menos tuvieron que comer un plato de cabezas

ENTRADA:

Cada caso tiene dos líneas. En la primera línea se indica una tira de caracteres p y c sin separación entre ellas para representar plato de patas(p) o cabezas(c). En la segunda línea se indican los nombres de los comensales de la cola separados por un espacio en blanco. Por lo tanto, esta segunda línea muestra el estado de la cola justo antes de comenzar a repartirse los platos de pulpo. El comensal más a la izquierda en la línea será el primero de la cola, el siguiente el segundo, etc.

Suponemos que todos los nombres de la entrada son únicos.

SALIDA:

Listado ordenado por nombre ascendetemente de los comensales de la cola que al menos tuvieron que comerse un plato de cabezas.

Ten en cuenta que si nadie come cabeza la salida será una lista vacía. Y ten en cuenta también que puede ocurrir que a la misma persona le hubieran tocado varios platos de cabeza pero sólo se lista su nombre una vez, por lo tanto, la salida nunca se va a contener nombres duplicados.

Observa detalles adicionales de la ENTRADA/SALIDA en los test de prueba.

Por ejemplo:

Entrada	Resultado
pppcpppppcpcpp yo tu el	[el, yo]
pppppppppppppppppppppp yo tu el	[]
ppppcppc jose ana telma	[ana]
cc z x y	[x, z]

Respuesta: (sistema de penalización: 0 %)

1 ||