**Ejercicio 1: Este ejercicio se responderá en esta hoja de examen. [10 puntos]**

Definir las constantes y tipos de datos apropiados para:

a) Almacenar en memoria principal un máximo de 100 Clientes. De cada Cliente se quiere almacenar su nombre, domicilio y un campo que indique si es habitual.

**[2,5 Puntos]**

b) Almacenar en memoria la ocupación de un hotel. El hotel tiene 10 plantas y 7 habitaciones por planta. De cada habitación queremos saber el número de personas que la ocupan.

**[2,5 Puntos]**

c) Almacenar la información de un mensaje de WhatsApp. El mensaje consta de remitente, destinatario, hora de envío y cuerpo del mensaje.

**[2,5 Puntos]**

d) Almacenar la información de los 60 mejores expedientes académicos de la ETSII. Lo único que nos interesa de los alumnos es su número de expediente. **[2,5 Puntos]**

**Ejercicio 2: [20 puntos]**

Escribir un método que reciba dos números enteros positivos y devuelva si son **amigos**. Dos números son amigos si la **suma de sus divisores** propios (distintos de ellos mismos) **son iguales** (se puede crear otro método auxiliar).

Por ejemplo los números (220, 284) sí son amigos.

**Ejercicio 3: [15 puntos]**

Escribir un **método recursivo** que reciba un número entero positivo y permita sumar los dígitos de un número.

Ejemplo: Entrada 1173 Resultado: 12

**Ejercicio 4: [40 puntos]**

Se quiere realizar un programa que realice distintas operaciones sobre arrays uni y bidimensionales.

Se tiene la siguiente declaración de tipos de datos



Se pide:

1. **MediaArray:** Realizar un método que reciba el *array* definido arriba y la clase Scanner y pida tantos datos como el usuario quiera y los vaya introduciendo en el array. El método devolverá la media de los datos introducidos. Para ello después de introducir un dato preguntará al usuario que quiere hacer (“1- Insertar más datos, 2 -Salir”). **[10 Puntos]**
2. **MaxArray:** Realizar un método reciba el *array* y devuelva el valor máximo que contiene. **[5 Puntos]**

c) **DesplazarArray:** Realizar un método que reciba un *array* y modifique el array desplazando sus elementos hacia la izquierda 1 posición. Así el segundo elemento pasaría a la primera posición, el tercer elemento pasaría a la segunda posición y así sucesivamente. El primer elemento pasaría a ocupar la última posición. No se puede usar otro array auxiliar. **[10 Puntos]**

Ejemplo:

Si inicialmente el array = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 cuando finaliza el subprograma el array = 1, 2, 3, 4, 5, 6, 7, 8, 9, 0.

d) **Matriz:** Realizar un método que reciba la *matriz* y el valor *fmatriz* ambos del enunciado y rellene sus elementos con los valores que se muestran en el ejemplo (deben usarse bucles y variables para hacerlo; a continuación el método mostrará el contenido de la matriz como en el ejemplo **[10 Puntos]**

Ej:

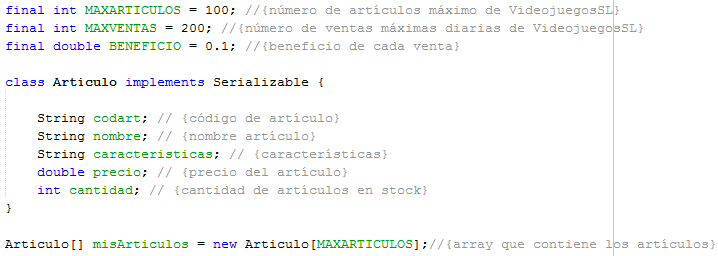


f) Escribe las instrucciones necesarias para realizar las llamadas a los métodos de los apartados a), b) y mostrar el valor que devuelve cada método correctamente al usuario por pantalla. **[5 puntos]**

**Ejercicio 5: [15 puntos]**

En la empresa **VideojuegosSL** necesitan almacenar los datos de sus artículos y sus ventas en sendos ficheros. Para ello se desean incorporar dos métodos que almacenen el array de artículos y el array de ventas a su programa de gestión. Las opciones que debe incluir el nuevo programa serán:

Los tipos de datos son los siguientes:





a) **Guardar las ventas realizadas en fichero de texto**. Realizar un método que reciba el *array de ventas* y escriba en un fichero de texto denominado “*ventas.txt*” una línea por cada venta, donde aparecerá el código de artículo, el precio del artículo (con dos decimales), el coste (precio de venta al público con dos decimales), la cantidad y el coste total como se muestra en el ejemplo. **[7,5 puntos]**

**Ejemplo** de cómo quedará el fichero de texto:

3455 5.12 5.63 20 112.64

5585 45.10 49.61 1 49.61

5585 7.20 7.92 5 39.60

b) **Guardar el stock de almacén en fichero binario**: Realizar un método que almacene el *array de artículos* de la ferretería en un fichero binario (*articulo.dat*). Se debe almacenar únicamente los datos de stock (solo los artículos existentes en el almacén) **[7,5 puntos]**