Apellidos:

Nombre: D.N.I.:

Introducción a la Programación - E.P. de Ingeniería de Gijón

14 de Enero de 2021

1. (2 %) Escribe una expresión para declarar un vector de números reales cuyo tamaño se lee de un objeto Scanner llamado `teclado`, que supondremos definido previamente.

2. (5 %) Dada la siguiente definición de variables y sus valores iniciales,

```
int x=9, y=3; double f=2.0; char c='a';
```

indica para las siguientes expresiones, si son o no correctas (SI/NO), en caso de resultar incorrectas JUSTIFICA por qué lo son, y en el caso de ser correctas indica el TIPO y el VALOR que producen.

Nota: Los dígitos, al igual que las letras del alfabeto, ocupan posiciones consecutivas en la tabla de códigos.

		Tipo	Valor	Motivo
`f = f / f;`	☐Si ☐No			
`f = 1 / y;`	☐Si ☐No			
`1 / y`	☐Si ☐No			
`(char)(c+y)`	☐Si ☐No			
`c == 'c'`	☐Si ☐No			
`y = c == 'a'`	☐Si ☐No			

3. (12 %) Implementa el método estático `piAprox()` para calcular (y retornar) de forma aproximada el valor de π. Para ello debes utilizar la serie conocida como *Producto de Wallis*:

$$\frac{2}{1}\cdot\frac{2}{3}\cdot\frac{4}{3}\cdot\frac{4}{5}\cdot\frac{6}{5}\cdot\frac{6}{7}\cdot\frac{8}{7}\cdot\frac{8}{9}\cdot\frac{10}{9}\cdot\frac{10}{11}\cdots=\frac{\pi}{2}$$

El método debe recibir un parámetro $n \geq 1$ que indica el número de términos del producto que debemos utilizar para calcular la aproximación. Por ejemplo, si se invoca `double myPi = piAprox(5)` el método debería retornar el resultado de utilizar los 5 primeros términos del producto, es decir,

$$\pi = 2\cdot\left(\frac{2}{1}\cdot\frac{2}{3}\cdot\frac{4}{3}\cdot\frac{4}{5}\cdot\frac{6}{5}\right)$$

4. (12 %) Diseñar el método público y estático `Seguidos` que, dado un vector no vacío de números enteros, calcule *la **longitud** de la subsecuencia más larga de números consecutivos e iguales.*

   Ejemplo. Para el vector 2, 1, 1, 2, 2, 5, 4, 4, 4 el método debe retornar 3 (ya que el número 4 aparece 3 veces seguidas, no hace falta retornar el 4, solamente el 3)

5. (8 %) Escribir un programa completo en el que se pidan por teclado tres números reales a modo de longitudes de lados y se indique si podrían formar un triángulo o no. Además, en caso afirmativo, el programa debe indicar qué tipo de triángulo sería: *equilatero* (tres lados iguales), *isósceles* (dos lados iguales) o *escaleno* (tres lados distintos).

6. (8 %) Un almacén de fontanería ofrece descuentos en función del nivel de compra de los clientes, de acuerdo a la siguiente tabla:

	Herramientas	Materiales
Ocasional	5%	5%
Habitual	10%	20%
Profesional	15%	20%

Diseña el esquema condicional que determine el descuento a aplicar en función del tipo de cliente y el producto comprado. Puedes suponer que partes de una variable `cliente` cuyo valor puede ser `'O'`, `'H'` o bien `'P'`, y de una variable `'producto'` cuyo valor puede ser `'H'` o `'M'`.

7. (16 %) Diseñar el método público y no estático `Alternadas` que dado un String que contiene una palabra devuelva `true` si las vocales y consonantes están alternadas, es decir, que no hay dos vocales, ni dos consonantes seguidas, y `false` en caso contrario. Se permite usar, sin programarlo, el método `public boolean esVocal(char )` que recibe un parámetro de tipo `char` y devuelve cierto si es una vocal y falso en caso contrario. Deben hacerse dos versiones del método `Alternadas`, una de ellas usando el esquema de búsqueda sin emplear `break` ni `return` dentro del bucle.

Ejemplos. "hola" → `true`, "adiós" → `false`, ya que hay 2 vocales seguidas, "palabra" → `false`, hay dos consonantes (b y r) seguidas.

8. (10 %) Escribir el método público y no estático `Abajo` que, dada una matriz de enteros no vacía, la modifica, desplazanado hacia abajo todos los valores de cada columna que son distintos de 0, quedando por tanto abajo los valores distintos de 0 y arriba los ceros.

Antes

0	3	6	1	2
2	3	6	0	0
0	0	0	12	0
0	0	0	0	0

$\Longrightarrow$

Después

0	0	0	0	0
0	0	0	0	0
0	3	6	1	0
2	3	6	12	2

9. (22 %) Implementa la clase `Colchón` para representar colchones. La información que debe recoger cada objeto de esta clase consiste en las dimensiones en cm. (largo, ancho y grosor) así como el tipo de material, que puede ser un valor entero entre 1 y 4 (1=espuma, 2=visco, 3=látex, 4=muelles)

- Constructores: por defecto (con los siguientes valores para cada atributo: largo=190, ancho=90, grosor=21, tipo=1), el de copia, con 1 entero para inicializar el tipo (los otros tres atributos tomarán los valores por defecto antes indicados), con 4 enteros para poder inicializar los 4 atributos.
- Métodos `get()` y `set()`. Ten en cuenta que el largo sólo puede ser 180, 190 o 200, el ancho sólo puede ser 90, 135 o 200, el grosor será de, al menos, 10 cm. y el tipo sólo puede ser 1, 2, 3 o 4.
- Método `calculaVolumen()`, que calcula el volumen del colchón.
- Método `esMásVoluminoso()` que recibe un objeto `Colchón` y devuelve cierto si el objeto con el que se llama al método tiene un volumen mayor que el objeto que se pasa como parámetro, y falso en caso contrario.
- Método `toString()`, que devuelve un String con la información del objeto `Colchón` con el siguiente formato: `"Tipo: espuma. Medidas: 190 cm. x 135 cm. x 32 cm."`. (largo x ancho x grosor)

10. (5 %) Hacer un programa de prueba de la clase `Colchón`. El programa debe crear varios objetos, usando todos los constructores, y emplear esos objetos para probar los métodos `esMásVoluminoso()` y `toString()` .