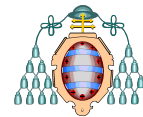




Apellidos:

Nombre:

D.N.I.:



## INTRODUCCIÓN A LA PROGRAMACIÓN - E.P. DE INGENIERÍA DE GIJÓN

25 de Enero de 2022

1. (2 %) Declara en una sola instrucción una constante estática para almacenar la aceleración de la gravedad, que es 9.80665 metros por segundo al cuadrado.

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

```
int x=9, y=2; double f=2.0; boolean a=true; 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
<code>y * a</code>	<input type="checkbox"/> Si <input type="checkbox"/> No			
<code>f * x / y</code>	<input type="checkbox"/> Si <input type="checkbox"/> No			
<code>x / y * f</code>	<input type="checkbox"/> Si <input type="checkbox"/> No			
<code>a == c</code>	<input type="checkbox"/> Si <input type="checkbox"/> No			
<code>c &lt; (char)(c+1)</code>	<input type="checkbox"/> Si <input type="checkbox"/> No			

3. (9 %) Dados dos enteros N y M, tales que  $N < M$ , indicar qué intervalo de valores imprimen EXACTAMENTE los siguientes bucles. Nota: "[ " indica intervalo cerrado (contiene el valor) y "(" intervalo abierto (no contiene el valor)

<pre>int i=N; while ( i &lt; M ) {     if(i%2==0) System.out.println(i);     ++i; }</pre>	<pre>int i; for(i=N-1;i&lt;M+1;i+=1)     System.out.println(i);</pre>

4. (10 %) Diseñar el método público y estático **DescensosBruscos** que, dado un vector no vacío que tiene el valor de cotización de cada día de las acciones de una empresa, el método debe retornar el número días en los que se han detectado descensos mayores del 5% sobre la cotización del día anterior.

Ejemplo. Para el vector {101.3,97.0,90.7,92.1,91.5} el método debe retornar 1, ya que solamente la diferencia de cotización entre el segundo y el tercer día (97.0 → 90.7) presenta un descenso mayor del 5% ( $97.0 \times 0.05 = 4.85$ ).

5. (6 %) Dadas dos variables `f` y `c`, de tipo `int` y una matriz cuadrada no vacía `m` de enteros, escribe la siguientes condiciones para que su valor sea cierto cuando...

... la variable `c` contiene un valor que puede ser un índice válido como columna de la matriz:

... los valores de las variables `f` y `c` indican una posición de la matriz triangular inferior (suponiendo que `f` contiene el índice de la fila y `c` el de la columna):

... el elemento en la fila `f` y columna `c` es impar o múltiplo de 7:

6. (12 %) La Entidad de Ayuda al Estudio subvenciona la matrícula de los alumnos que estudian en España en función de los ingresos de sus familias y de su país de procedencia según la tabla mostrada. Se pide hacer un **programa de consola completo (importando las clases necesarias)** que pida por teclado los ingresos y el país de origen (que se introduce mediante la primera letra del país) y que muestre en la consola el porcentaje que corresponda según los datos introducidos escribiendo un mensaje como el siguiente: **[País=P, Ingresos=27546,34] Se subvenciona un 20% de la matrícula**, si el país es Portugal y los ingresos 27.546,34€.

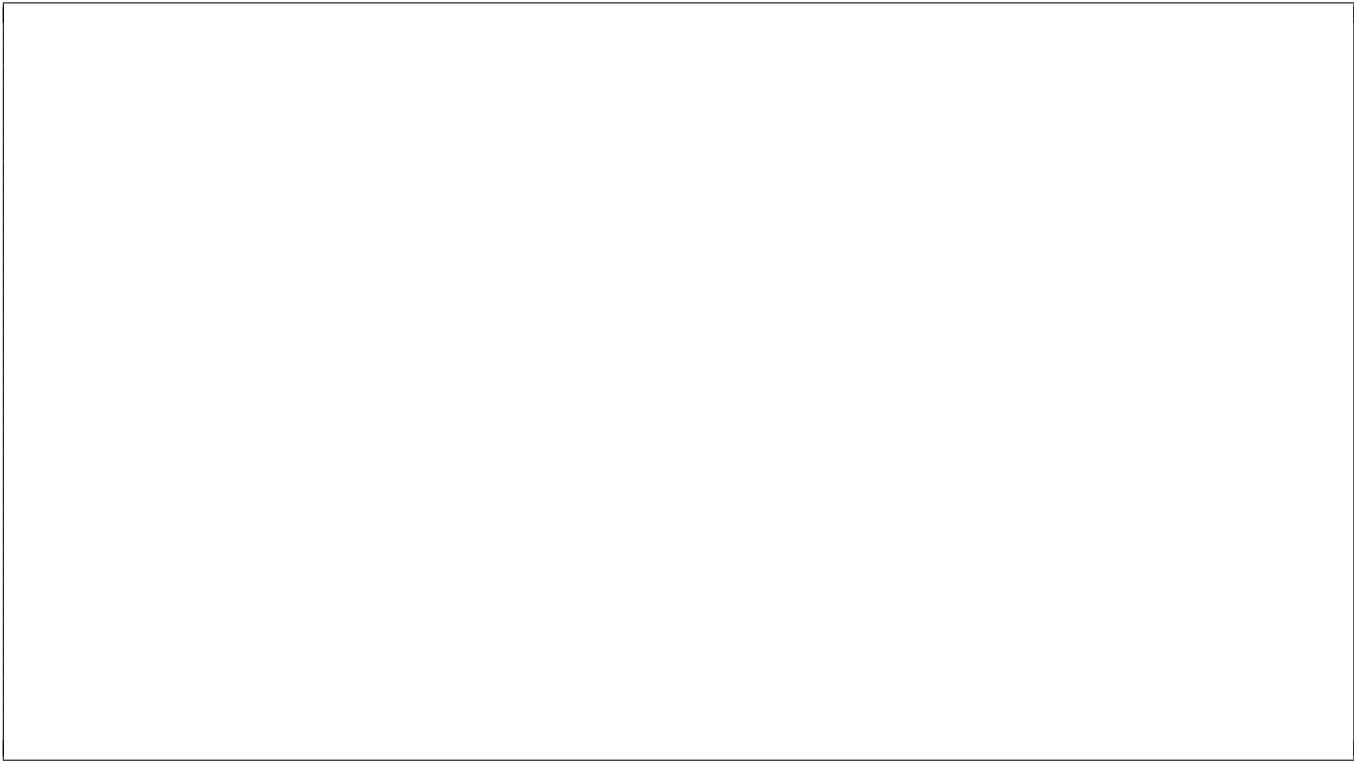
Ingresos	(E)spaña	(P)ortugal	(I)talia	(R)eino Unido
hasta 20.000,0€ (incluido)	50%	50%	50%	20%
entre 20.000,0€ y 45.000,0€	50%	20%	20%	20%
a partir de 45.000,0€ (incluido)	20%	20%	10%	10%

7. (16 %) Diseñar el método público y no estático **tieneCH** que dado un String (en minúsculas) que contiene una palabra devuelva **true** si contiene la subcadena "ch" y **false** en caso contrario. **Restricción:** no se pueden usar ningún método que busque subcadenas dentro de un String. Deben hacerse dos versiones del método **tieneCH**, una de ellas usando el esquema de búsqueda sin emplear **break** ni **return** dentro del bucle.

Ejemplos. "cochinillo" → **true**, "cohibido" → **false**, "marcha" → **true**.

8. (12 %) Escribir el método público y estático **borraDiagIguales** que recibe una matriz **m** de caracteres no vacía y un entero **c** que indica una de las columnas de la matriz. El método debe modificar la matriz colocando el carácter '-' en todas las posiciones que parten de forma diagonal desde la fila 0 y columna **col** y que sean iguales al carácter en (fila=0, columna=**col**). Ejemplo del estado de la matriz antes y después de hacer la llamada: **borraDiagIguales(m,3)**:

Antes de la llamada							Después de la llamada					
y	g	g	b	g	r	⇒	y	g	g	-	g	r
b	g	b	b	b	y		b	g	-	b	-	y
g	g	y	y	r	b		g	g	y	y	r	-
b	b	r	y	y	y		-	b	r	y	y	y



9. (23 %) Implementa la clase **Tenista** para representar la condición física de tenistas. La información que debe recoger cada objeto de esta clase consiste en el nombre del tenista, su peso (en Kg) y su altura (en metros).
- Constructores: por defecto (con los valores "Elsa" para nombre, 70 para el peso y 1.70 para la altura), el de copia y otro con 3 parámetros que se corresponderán con nombre, peso y altura.
  - Métodos `get()` y `set()`. El peso debe estar entre 10 y 200 Kg y la altura entre 0.50 y 2.30 metros.
  - Método `calculaIMC()`, que calcula el Índice de Masa Corporal (IMC) mediante la fórmula  $\text{peso}/\text{altura}^2$ .
  - Método `tieneMásIMC()` que recibe un objeto **Tenista** y devuelve cierto si el objeto con el que se llama al método tiene un IMC mayor que el del objeto que se pasa como parámetro, y falso en caso contrario.
  - Método `toString()`, que devuelve un String con los datos del objeto **Tenista** y su IMC con el siguiente formato: "Elsa, Peso: 70,00 Kg, Altura: 1,70 metros, IMC: 24,22".



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