

Problemas de introducción a Java (v1)

Aviso para los que ya conocéis Java: soy consciente de que algunos problemas pueden solucionarse con una simple llamada a métodos de la biblioteca de Java. El objetivo es resolverlos usando los métodos explicados en clase y/o en los apuntes. A lo largo de la asignatura ya iremos introduciendo más métodos de la biblioteca estándar.

En todos los problemas intentaremos añadir código que nos permita comprobar, hasta cierto punto, que las funciones que hemos programado realizan correctamente la funcionalidad que tienen encomendada (lo que se conoce como **código de pruebas**).

Problema 1.

Diseñad e implementad un programa tal que pida al usuario un número entero mayor que 1 y, como respuesta, indique si el entero dado es perfecto o no. Si el número introducido no es mayor que 1, se mostrará un mensaje de error.

Un número es perfecto si es la suma de todos sus divisores (excepto de él mismo). Por ejemplo 6 es perfecto pues sus divisores son 1, 2 y 3, que suman 6. Otros números perfectos son 496 y 8128.

Definid una función auxiliar siguiente para comprobar dicha propiedad:

```
1 public boolean isPerfect(int n) {  
2     ¿?  
3 }
```

Problema 2.

Inspirándoos en el ejemplo 3 de los apuntes, definid una función que devuelva el elemento máximo de un array de enteros que, como mínimo, tiene un elemento.

```
1 public int max(int[] numbers) {  
2     ¿?  
3 }
```

Para el programa principal podéis usar la siguiente estructura para el run:

```
4 public void run() {  
5     int[] nums = new int[] {4, 12, -5, 3};  
6     int max = max(nums);  
7     println("El resultado es " + max);  
8 }
```

Que debería escribir "El resultado es 12" en la pantalla.

Probad con diferentes vectores y, por ejemplo, uno de tamaño mínimo.

Problema 3.

Diseñad e implementad un método tal que como parámetro de entrada reciba un array de caracteres y como resultado el array consistente en devolver los mismos elementos y en el mismo orden que el array original excepto el primero y el último.

```
1 public char[] removeExtremes(char[] chars) {  
2     ¿?  
3 }
```

Por ejemplo, si el vector de entrada fuera { 'a', 'b', 'c', 'd' } el resultado sería { 'b', 'c' }.

Si el array tiene una longitud menor o igual que dos, se devuelve el array vacío.

Haced un programa que pruebe con arrays de diferentes longitudes y escriba tanto el array de entrada como el resultado.

Para la función que escribe el array os podéis inspirar en el println del ejemplo 4 de los apuntes.

Problema 4.

Diseñad e implementad una función que reciba un vector de caracteres y como resultado devuelva el vector que contenga los mismos caracteres originales pero sin los que son espacios en blanco.

```
1 public char[] removeSpaces(char[] chars) {  
2     ¿?  
3 }
```

Por ejemplo, si el vector de entrada fuera { ' ', 'a', ' ', 'c', 'd', 'e' }, el resultado sería { 'a', 'c', 'd', 'e' }.

Haced un programa que pruebe con arrays de diferentes longitudes y escriba tanto el array de entrada como el resultado.

Para la función que escribe el array os podéis inspirar en el `println` del ejemplo 4 de los apuntes.

Problema 5.

Diseñad e implementad un método para multiplicar matrices de enteros. El método recibirá como parámetros dos matrices y devolverá como resultado otra matriz. Es decir:

```
1 public int[][] matrixMultiplication(int[][] left,
2                                     int[][] right) {
3     ¿?
4 }
```

Simplificaciones y pistas:

- Dentro de la función podéis suponer que las matrices son de las dimensiones que tienen que ser (el número de columnas de `left` ha de coincidir con el número de filas de `right`).
- Haced un programa principal que cree un par de matrices constantes (aseguraros de que son de las dimensiones correctas).
- Podéis usar el método `print` definido en el ejemplo de los apuntes (o programar uno por vosotros mismos).

Problema 6.

Diseñad e implementad un método tal que como parámetro de entrada reciba una cadena de caracteres `str` y como resultado devuelva la cadena consistente en eliminar los caracteres primero y último de `str`. Es decir:

```
1 public String removeExtremes(String str) {
2     ¿?
3 }
```

Tened en cuenta qué devolvéis si la cadena está vacía o consiste solamente en un carácter.

Haced un programa que pida un texto al usuario y muestre el texto sin los extremos.

Problema 7.

Diseñad e implementad un método tal que como parámetro de entrada reciba una cadena de caracteres `str` y como resultado devuelva la

cadena consistente en eliminar todos los espacios de `str`. Es decir:

```
1 public String removeSpaces(String str) {  
2     ¿?  
3 }
```

Haced un programa que pida un texto al usuario y muestre el texto sin espacios.