

C. F. G. S. Desarrollo de Aplicaciones Web - Distancia - Examen de Programación de la 1º Evaluación. 17-12-2024

“First, solve the problem. Then, write the code.” - John Johnson –

- 1) Un número es narcisista si es igual a la suma de sus dígitos elevados a la potencia del número de dígitos del número.

Por ejemplo, **371** es un número narcisista, ya que:

$$3^3 + 7^3 + 1^3 = 27 + 343 + 1 = \mathbf{371}$$

También lo es, por ejemplo, **93084**:

$$9^5 + 3^5 + 0^5 + 8^5 + 4^5 = 59049 + 243 + 0 + 32768 + 1024 = \mathbf{93084}$$

Hacer un programa que pida un número por teclado y que escriba si es narcisista o no.

Algunos números narcisistas son: 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727, 93084, 548834, 1741725.

- 2) Escribir un programa que pida por teclado un número entero impar n, comprendido entre 3 y 25, ambos incluidos, y que a continuación dibuje una figura como la siguiente (n = 9).

```
* * * * * * * * *  
* * * * * * * *  
* * * * * *  
* * * *  
* * *  
* *
```

Hay que obligar al usuario a que introduzca un valor impar comprendido entre 3 y 25. En el caso de que el valor leído no cumpla estos requisitos, se visualizará el mensaje de error “número incorrecto, tiene que ser impar y comprendido entre 3 y 25”, y se volverá a pedir otro valor hasta que sea válido.

Ejemplo de ejecución del programa (utilizar el mismo formato):

```
Introduce un numero impar entre 3 y 25: 2  
Número incorrecto, tiene que ser impar y comprendido entre 3 y 25.  
Introduce un numero impar entre 3 y 25: 26  
Número incorrecto, tiene que ser impar y comprendido entre 3 y 25.  
Introduce un numero impar entre 3 y 25: 5
```

* * * * *
* * *
*

- 3) Hacer un programa que lea dos números por teclado y que escriba en orden descendente todos los valores comprendidos entre los números leídos (ambos incluidos) que sean divisibles por 5. Tener en cuenta que el primer número leído puede ser menor que el segundo. Separar por comas los números escritos, escribiendo un punto al final.

Ejemplo de ejecución del programa (utilizar el mismo formato):

Introduce el primer numero: 10

Introduce el segundo numero: 54

50, 45, 40, 35, 30, 25, 20, 15, 10.

- 4) Dado el ejercicio: “Leer números por teclado hasta que se introduzca un cero, y después calcular la suma de los valores introducidos”, modificar la solución propuesta, utilizando un bucle for. (tener en cuenta que el primer número leído puede ser el 0)

```
public class Ejercicio55 {  
  
    public static void main(String[] args) {  
        int n;  
        int r = 0;  
        Scanner teclado = new Scanner(System.in);  
  
        System.out.print("Introduce numeros: ");  
        n = teclado.nextInt();  
        while (n != 0) {  
            r += n;  
            n = teclado.nextInt();  
        }  
        System.out.println("El resultado de la suma es: " + r);  
    }  
}
```

- 5) Hacer un programa que pida por teclado una cadena de caracteres, que tiene que estar formada por parejas de dígito + carácter. A continuación, escribirá cada carácter en mayúsculas tantas veces como indique el número que lo precede. No es necesario comprobar que la cadena leída tiene el formato correcto.

El programa tiene que ejecutarse hasta que se teclee fin, tanto en mayúsculas como minúsculas.

Ejemplo de ejecución del programa (utilizar el mismo formato):

Introduce una cadena formada por parejas de dígito + carácter (o 'fin' para salir): 2q3B

Resultado: QQQBBB

Introduce una cadena formada por parejas de dígito + carácter (o 'fin' para salir): 1p

Resultado: P

Introduce una cadena formada por parejas de dígito + carácter (o 'fin' para salir): 5H1f2b

Resultado: HHHHHFBBB

Introduce una cadena formada por parejas de dígito + carácter (o 'fin' para salir): fin

Programa finalizado.

Puntuación de cada ejercicio:

- Ejercicio 1: 2 puntos
- Ejercicio 2: 2,5 puntos
- Ejercicio 3: 2 puntos
- Ejercicio 4: 1 punto
- Ejercicio 5: 2,5 puntos

Criterios de corrección y cómo entregar el examen:

Aunque un programa no funcione, entregarlo igual, ya que puede ocurrir que esté bien planteado y falle por un pequeño detalle. De la misma manera, si se pide hacer un programa con una determinada sentencia para implementar un bucle y no os sale, hacerlo de otra. Evidentemente no puntuará lo mismo, pero al menos podéis completar el ejercicio.

Hay que subir la solución del examen como tarea, en formato de procesador de texto en un único fichero, que incluya el código fuente completo de cada uno de los ejercicios. La tarea está antes de la UD1 con el nombre “Examen de a 1^a evaluación”. Nombrar al documento con vuestro nombre y apellidos.

AYUDA PARA EL EXAMEN

A la hora de programar, netbeans nos ayuda a escribir código de manera más rápida mediante decenas de atajos (shortcuts). NetBeans incluye un archivo pdf con la lista completa de dichos atajos. Esta lista se encuentra dentro del directorio de instalación de NetBeans en una carpeta llamada “nb”. Allí encontraremos 2 pdfs:

shortcuts.pdf

shortcuts_mac.pdf

El primero contiene los atajos para PC (Windows/Linux), el segundo para los usuarios de Mac.

Acordaros dónde está el fichero shortcuts.pdf para poder consultarla el día del examen. Os recomiendo que os familiaricéis con el uso de estos atajos. De todas maneras, os pongo a continuación algunos de los atajos más útiles:

1. sout + Tab, genera la línea: System.out.println("");
2. psvm genera un método main

```
public static void main(String[] args) {  
}
```

3. sw + Tab, genera una sentencia switch

```
switch (var) {  
    case val:  
  
        break;  
    default:  
        throw new AssertionError();  
}
```

4. for, while, if +ctrl+spacio dan sugerencias para construir cualquiera de estas estructuras de control

Tampoco es necesario saberse de memoria todos los métodos de cada una de las clases que tiene el lenguaje. Para eso podemos consultar la API del lenguaje y los IDEs nos ayudan. Si no recordamos como se calcula el valor absoluto de un valor, simplemente escribiendo Math. (clase utilizada en los apuntes), nos aparecen todos los métodos de la clase, con lo que podemos localizar el que nos interesa utilizar. De la misma manera, si después de una variable de tipo String pongo el punto también me aparecen todos los métodos (extraer un carácter, convertir a mayúsculas.....).

Math.|

 E	double
 PI	double
 IEEEremainder (double d, double d1)	double
 abs (double d)	double
 abs (float f)	float
 abs (int i)	int
 abs (long l)	long
 acos (double d)	double
 asin (double d)	double
 atan (double d)	double
 atan2 (double d, double d1)	double
 cbrt (double d)	double
 ceil (double d)	double

String nombre;

nombre.|

 charAt (int i)	char
 codePointAt (int i)	int
 codePointBefore (int i)	int
 codePointCount (int i, int il)	int
 compareTo (String string)	int
 compareToIgnoreCase (String string)	int
 concat (String string)	String
 contains (CharSequence cs)	boolean
 contentEquals (CharSequence cs)	boolean