

PROGRAMACIÓN MODULAR

Realiza el pseudocódigo antes de pasar a Java, al menos en los cuatro primeros ejercicios.

21.- Dados dos valores numéricos reales almacenados en dos variables de nombre a y b, hacer un programa que llame a una función de nombre Cambiar(v1, v2) a la que se le pasan como parámetros dos variables e intercambia sus valores mostrando el resultado.

22.- Escribe un programa con diseño modular que recoja por teclado dos números reales y presente una serie de opciones correspondientes a operaciones aritméticas a realizar con las mismas:

1. Sumar
2. Restar
3. Multiplicar
4. Dividir

En función de la operación elegida, se llama a la función correspondiente que retorna el valor del resultado de dicha operación, que se muestra en el programa principal.

23.- Escribir un programa que utilice una función recursiva para calcular y retornar la potencia de un número real distinto de cero, siendo su exponente entero y positivo.

$$X^n = X * X^{n-1} \quad X^0 = 1$$

24.- Realizar un programa que disponga de una función llamada **esBisiesto** que reciba un año y devuelva cierto si es bisiesto y falso si no lo es.

Realizado en el Ejercicio 13: Los años que sean divisibles por 4 serán bisiestos; aunque no serán bisiestos si son divisibles entre 100 (como los años 1700, 1800, 1900, 2100) a no ser que sean divisibles por 400 (como los años 1600, 2000, 2400).

25.- Construir un programa para un proceso que lea un número entero mayor de 3 y menor de 3000 (se debe comprobar que se encuentre en este rango de valores, pidiendo otro valor en su caso) y nos muestre los números primos que sean menores que dicho número. Para la realización de este algoritmo se debe crear una función **esPrimo**, a la que se le envíe un número entero como parámetro y devuelva un 1 si el número enviado es primo y un 0 en el caso de que no lo sea, o bien devuelva true o false.

Un número es primo si sólo es divisible por sí mismo y por 1.

26.- Realizar un programa que incluya una función llamada **siguientePrimo**. La función recibirá un parámetro que será un número entero y deberá devolver el siguiente número primo que sea mayor al número que se le ha enviado. En el caso de que el parámetro enviado fuera negativo o cero, deberá devolver un 1.

27.- Construir un programa que pida dos números enteros, obligatoriamente positivos, y nos muestre por pantalla los números perfectos que pudiera haber entre los mismos. El programa debe contar con una función llamada **esDivisor** que controle si un número es divisor de otro, devolviendo verdadero o falso en razón de que lo sea o no, y otra que se denomine **esPerfecto**, que reciba un número y que nos indique si es perfecto o no.

Un número es perfecto si la suma de sus divisores, excepto él mismo, es igual al propio número. (6 es perfecto; 1+2+3=6)

28.- Construir un programa que lea un número entero positivo comprendido entre 1 y 15 (debe controlarse este rango) y escriba tantas líneas como indique el número de la forma que se detalla a continuación.

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

29.- Codificar un programa que lea un número entero positivo en el sistema decimal y muestre el número en sistema binario. Utiliza una función **pasaBinario**.

30.- Realiza un programa que proporcione 15 resultados para rellenar una quiniela de fútbol (posibles resultados 1, x, 2). Los resultados deben generarse de forma aleatoria.