

Ejercicios de Informática Industrial

2014-2015

1. Calcular el máximo de un array de números
2. Calcular el factorial de un número.
3. Ordenar un vector de números.
4. Detectar cuando una palabra introducida es un palíndromo (se lee igual de izquierda a derecha, que de derecha a izquierda). Por ejemplo: Ana, arenera, oso, radar, reconocer, rotor, salas, seres, somos, sometemos, etc.
5. Introducir dos matrices y multiplicarlas.
6. Realizar un programa que nos detecte si los caracteres que introducimos son letras, números u otra cosa.
7. Convertir una cadena de caracteres en minúscula a mayúscula.
8. Escribir un programa que permita realizar distintas operaciones matemáticas sobre un par de números: sumar, restar, multiplicar, dividir, etc. El programa pedirá que se introduzcan los números con los que se va a operar así como la operación que se desea aplicar. Las funciones que implementan las operaciones tienen que estar implementadas en un fichero distinto de donde se encuentra la función main y deben contar además con el correspondiente fichero de cabecera.
9. Realizar el mismo ejercicio que el anterior pero ahora los números para operar se le pasarán como parámetros al programa.
10. Crear una estructura/clase que represente cada uno de los alumnos de la asignatura. Debe existir al menos los siguientes campos: nombre, apellidos, nie y nota. Crear un array/lista donde se vayan almacenando todos los alumnos para posteriormente calcular la nota media de todos; también se podrán listar los que tienen sobresaliente, notable, aprobado o suspenso.
11. Hacer un programa que pida al usuario una cadena de caracteres y detecte si es o no un número entero, y, si lo es, que lo calcule.
12. Lo mismo que el anterior, pero incluyendo la posibilidad de detectar números decimales.
13. Difícil. Hacer un programa que almacene los nodos contiguos de una red, con las distancias entre ellos, y que calcule la ruta más rápida entre dos lugares. Se puede hacer con parte de la red de transportes de la Comunidad de Madrid.

14. Hacer un programa que calcule si un número es primo o no, y que, si no lo es, lo descomponga en factores.
15. Hacer un juego en el que, un tablero de diez por diez se llene aleatoriamente de unos y ceros. Luego, cada elemento mantendrá su valor (uno o cero) hasta la generación siguiente si más de dos tercios de sus vecinos son iguales a él, cambiará si menos de un tercio de sus vecinos son iguales a él, y tendrá un 50% de probabilidad de mantener su valor (y el restante 50% de cambiarlo) si entre un tercio y dos tercios de sus vecinos son iguales a él. Se mostrará por pantalla el valor de los elementos en forma de matriz.
16. Sencillo. Hacer un programa que pida los elementos de una matriz, de tamaño prefijado, y que la muestre por pantalla.
17. Lo mismo, pero con un tamaño de la matriz variable, que se pregunta previamente al usuario.
18. Hacer un programa que detecte si un número de teléfono es fijo o móvil, y, si es fijo, que diga de qué provincia es (en una primera aproximación, se puede hacer que distinga si es de Barcelona, de Madrid, o de otra provincia).
19. Sencillo. Hacer un programa que pida las dimensiones de un paralelepípedo, y a partir de ello, calcule su área y su volumen.
20. Hacer un programa que pida el nombre, primer apellido y segundo apellido de un conjunto de personas, que los almacene, y que luego permita ordenarlos alfabéticamente por cualquiera de los tres campos, o sacar los que empiecen por una determinada letra de algún campo.
21. Difícil. Lo mismo que el anterior, pero con un número variable de personas (memoria dinámica) y la posibilidad de encontrar cualquier cadena de caracteres en cualquier parte de cualquiera de los campos.
22. Difícil. Hacer un programa que detecte rimas en un texto.
23. Difícil. Hacer lo mismo que el anterior, pero además, contando sílabas.
24. Hacer un programa que detecte tantos números como haya en un texto, reserve la memoria necesaria para ellos, los calcule como enteros y los guarde.
25. La sucesión o serie de Fibonacci es la sucesión de números:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Cada número se calcula sumando los dos anteriores a él.

- El 2 se calcula sumando (1+1)
- Análogamente, el 3 es sólo (1+2),
- Y el 5 es (2+3),
-etc..

Realice un programa que cree un vector de números de Fibonacci e imprímalos por pantalla. Para crear el vector escriba una función **fibonacci(x,y,v,n)**

Donde **x** e **y** son dos enteros que me indicarán los dos números iniciales del vector

$$v[0] = x \quad v[1] = y$$

Cada elemento del vector es la suma de los dos anteriores.

v es el vector que vamos a crear y **n** el número de elementos del vector.

26. Al programa anterior, añada una nueva función que ordene en forma decreciente los números del vector de Fibonacci creado e imprímalos por pantalla.

27. Compruebe si un número entero positivo corresponde a un número de la serie de Fibonacci. Para esto use la fórmula de Binet:

Si **N** es un número de Fibonacci entonces $5 \cdot N^2 + 4$ ó $5 \cdot N^2 - 4$ son cuadrados perfectos.

28. Fácil: Escribir una función que acepta un límite superior y un límite inferior, calcular los cuadrados de cada uno de los números enteros entre el superior y el límite inferior [inclusive], y luego imprimir la suma de todas las cuadrados. Hacerlo a través de la recursividad.

Eazy: Write a function to accept an upper and a lower limit, calculate the squares of each of the integer number between the upper and the lower limit [inclusive], and then print the sum of all the squares. Do so using recursion.

29. Medio: Aceptar un número entero, junto con su base [binario, octal, decimal o hexadecimal], validar las entradas, a continuación, convertir el número en una base diferente, conforme a lo solicitado por el usuario, y luego imprimir el número de convertidos.

Medium: Accept an integer number along with its base [Binary, octal, decimal or hexadecimal], validate the inputs, then convert the number into a different base, as requested by the user, and then print the converted number.

30. Difícil: Escribir una función para aceptar un número entero [hasta cinco dígitos], y imprimir el número de palabras.

Eg: (a) 65 – Sixty Five.

(b) 12345 – Twelve Thousand Three Hundred and Forty Five.

Difficult: Write a function to accept an integer number[Up to five digits long], and display the number in words.

Eg: (a) 65 – Sixty Five.

(b) 12345 – Twelve Thousand Three Hundred and Forty Five.

31. Difícil: Escribir un programa para aceptar cualquier fecha en el siglo 21 y la salida del día de la semana de esta fecha cae. Validar la entrada del número de días en un mes, año bisiesto, el número de meses en un año, etc.

Difficult: Write a program to accept any date in the 21st century and output the day of the week this date falls on. Validate the input for number of days in a month, leap year, number of months in a year, etc.