

**Prácticas de la Asignatura
METODOLOGÍA Y TECNOLOGÍA DE LA
PROGRAMACIÓN
Primer Cuatrimestre
Curso 2006-2007.**

1º Ingeniero Técnico en Informática de Gestión.

1º Ingeniero Técnico en Informática de Sistemas.

Práctica 1: Programación en C básica (I).

Enunciado

Implementad en C los siguientes algoritmos:

1. Algoritmo que pida un entero que represente un tiempo en segundos, y lo pase a horas, minutos y segundos.
2. Algoritmo que pida los parámetros (a , b , c y d de tipo double) de dos planos, correspondientes a su ecuación implícita ($ax + by + cz + d = 0$), e indique si son coincidentes, paralelos o secantes. En el caso de que sean secantes, hay que indicar si son perpendiculares o no.
3. Algoritmo que pida cuatro números de tipo real e indique cual es el menor y cual es el mayor.
4. Algoritmo que sume los n primeros números enteros.
5. Algoritmo que determine si un número entero dado es perfecto o no. Se dice que un número es perfecto cuando es igual a la suma de sus divisores, excluido él mismo. Por ejemplo 6 es perfecto, ya que $6 = 1 + 2 + 3$.
6. Algoritmo que determine si un año dado es bisiesto o no, teniendo en cuenta que son años bisiestos los múltiplos de 4, excepto los que son también múltiplos de 100 pero no lo son de 400.
7. Algoritmo que descomponga un número entero dado en unidades, decenas, centenas, etc. No se pueden usar cadenas para resolver el problema.

Duración de la práctica: 2 sesiones.

Práctica 2: Programación en C básica (II).

Enunciado

Implementad en C los siguientes algoritmos. Para ello, la solución deberá ser lo más modular posible, usando funciones en aquellos casos en que sea conveniente.

1. Algoritmo que pida un número entero y determine si es primo o no. Para ello implementar una función que determine si un número es primo o no.
2. Algoritmo que descomponga un número par dado, en la suma la suma de dos números primos (conjetura de Goldbach). Para ello usad la función implementada en el problema anterior.
3. Algoritmo que determine si un número entero x dado, es igual al factorial de otro número. En caso contrario determinad el número cuyo factorial más se aproxima a x .
4. Algoritmo que determine si un número es capicúa (se lee lo mismo de derecha a izquierda que de izquierda a derecha).
5. Calcular el valor del coseno de 10° , con un error menor que una cantidad dada por el usuario, usando el desarrollo en serie de Taylor ($x - x^3/3! + x^5/5! - x^7/7! \dots$). Nota: El ángulo hay que pasarlo a radianes y el error cometido es menor que el primer término despreciado.
6. Implementad un programa en C, en el que aparezca un menú con los cinco ejercicios anteriores.

Duración de la práctica: 2 sesiones.

Práctica 3: Punteros y vectores.

Enunciado

Implementad los siguientes programas en C.

1. Programa en el que se declare un entero, un puntero simple a entero y un puntero doble a entero. Por medio del puntero simple se ha de introducir el valor del entero y visualizar por pantalla su raíz cuadrada, y por medio del puntero doble, se ha de asignar un nuevo valor al entero y mostrar su factorial por pantalla.
2. Programa en el cual se declaren dos números de tipo float, se pasen a una función por referencia y se intercambien sus valores. Implementad también una función errónea, que reciba los parámetros por valor, e intente intercambiarlos. Para comprobar la razón de su comportamiento erróneo, mostrar por pantalla los valores y las direcciones de los parámetros actuales y de los formales.
3. Programa que genere aleatoriamente una apuesta de la lotería primitiva (suponemos que los números van desde 0 a 48) y la guarde en un vector de 49 elementos, cuyos valores son ceros y unos. Para almacenar la apuesta, seguiremos el siguiente procedimiento: si el número i ha salido, el elemento $v[i]$ será igual a 1, y si no ha salido, el $v[i]$ será igual a 0. Para generar la apuesta, se usará una función que reciba el vector inicializado a 0, y genere aleatoriamente seis números distintos comprendidos entre 0 y 48.
4. Programa que usando la función implementada en el ejercicio

anterior, genere una combinación ganadora, y n apuestas (n será introducido por el usuario), y contabilice cuantas apuestas tienen 0 aciertos, 1 acierto, 2 aciertos, 3 aciertos, 4 aciertos, 5 aciertos y 6 aciertos, usando un vector auxiliar denominado *aciertos* de 7 elementos. Para facilitar la contabilización de aciertos, contabilizad éstos después de generar cada apuesta. **Nota:** para contabilizar los aciertos usad el producto escalar.

5. Programa en el cual se introduzcan en una función los datos de lluvia mensual, usando un vector de tipo estructura en el cual aparezcan dos campos (cadena de 15 elementos para el nombre y un double para el valor de lluvia), para un año determinado y se obtenga la lluvia anual, la media mensual, la desviación típica, el mes más lluvioso y el mes más seco. Para obtener cada uno de los cinco valores anteriores se implementará una función distinta.

Duración de la práctica: 1 sesión.

Práctica 4: Cadenas y matrices.

Enunciado.

Implementad en C los siguientes programas:

1. Programa que descomponga un número entero dado en unidades, decenas, centenas, etc. Para realizar este ejercicio hay que almacenar el entero en una cadena, y posteriormente trabajar con la cadena para realizar lo pedido.
2. Programa que indique si un número entero es capicúa. Para realizar este ejercicio hay que almacenar el entero en una cadena, y posteriormente trabajar con la cadena para realizar lo pedido.
3. Programa en el cual se declare una matriz de $4 * 3$ de tipo double y sus valores se introduzcan en una función. Después se invocará a una función que calcule la media de los elementos de cada fila y se almacene en un vector de cuatro elementos. Posteriormente se invocará a otra función que obtenga la media de los elementos de las columnas y los almacene en un vector de tres elementos. Para visualizar los vectores resultado, se usará una función para escribir un vector en pantalla. Finalmente se invocará a otra función que calcule la matriz traspuesta de la matriz dada. Para visualizar la matriz traspuesta se usará otra función que permita visualizar por pantalla una matriz.

Duración: 1 sesión.