**Fundamentos de Programación (8 de septiembre de 2017)**

**(Grados en Ingeniería Mecánica, Eléctrica, Electrónica Industrial y Química Industrial)**

**Ejercicio 1 – Programación Estructurada (2 puntos):** construir un programa en ***C*** que lea por teclado un número entero ***n*** no negativo y menor de 500, y que calcule (de la forma más eficiente posible) y presente en pantalla todos los números combinatorios ***Cn,k*** para valores de ***k*** desde ***0*** hasta ***n***.

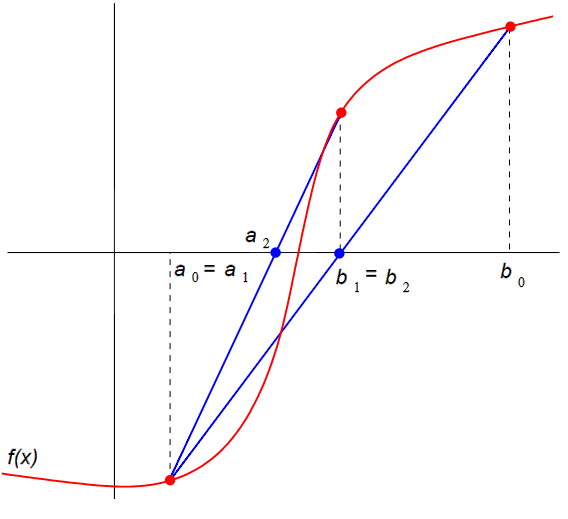
Nota:

**Ejercicio 2 – Diseño Modular (2 puntos):** construir una función en ***C*** que calcule y devuelva una raíz ***r*** de una función ***f:R***🡪***R*** en un intervalo dado ***[a,b]*** utilizando el método de la falsa posición (“regula falsi”). Considerar el siguiente prototipo de dicha función:

/\* Prototipo de función a construir en el ejercicio \*/

int regula\_falsi(double (\*f)(double x),double a,double b,double \*r);

Nota: se dice que ***r*** es una raíz de la función ***f*** si se cumple que ***f(r)=0***. Para buscar la raíz se construirá una sucesión de aproximaciones ***{xi}*** que converja a ***r***. Si la función ***f*** es continua en el intervalo ***[a,b]*** y además se cumple que ***f(a)\*f(b)<0*** (la función toma distinto signo en los extremos del intervalo) entonces existe al menos una raíz en dicho intervalo (Teorema de Bolzano).



En el método de la falsa posición, partiendo del intervalo

inicial ***[a0,b0]=[a,b]***, se va construyendo una sucesión de

intervalos ***[an,bn]*** tales que ***f(an)\*f(bn)<0*** y se va calculando

el punto de corte de la recta que pasa por los puntos ***(an,f(an))***

y ***(bn,f(bn))*** con el eje de abscisas:

Dicho punto divide el intervalo ***[an,bn]*** en dos sub-intervalos del que hay que escoger el que contiene la raíz, reiterándose todo el proceso anterior hasta que ***f(xn+1)*** sea lo suficientemente pequeño (***<1E-10***).

A través del identificador de función se devolverá: ***0*** si cualquiera de los dos extremos del intervalo ya es una raíz (devolviéndose a través del parámetro de salida, y en el caso de ser los dos extremos raíces solo hay que devolver uno de ellos), ***-1*** si no se cumple el Teorema de Bolzano y ninguno de los extremos es una raíz, ***1*** si se cumple el Teorema de Bolzano.

**Ejercicio 3 – Estructuras de Datos (2 puntos):** construir una función en ***C*** que calcule la matriz traspuesta conjugada de una matriz cuadrada de dimensión máxima ***10*** dada como argumento, y que devuelva la misma a través del mismo argumento de entrada. Considerar las siguientes estructuras de datos y prototipo de función:

/\* Estructuras de datos a considerar en el ejercicio \*/

#define MAX 10

typedef struct{

double re;

double im;

}tipo\_complejo;

typedef tipo\_complejo tipo\_array[MAX][MAX];

typedef struct{

int n; // Dimensión de la matriz

tipo\_array a;

}tipo\_matriz;

/\* Prototipo de función a construir en el ejercicio \*/

void traspuesta\_conjugada(tipo\_matriz \*m);

Notas:

Conjugado de un número complejo: otro número complejo obtenido cambiándole el signo a la parte imaginaria: conjugado(a+b\*i)=a-b\*i

siendo ***i*** la unidad imaginaria ()

Matriz traspuesta de una matriz cuadrada: matriz obtenida al intercambiar las filas por las columnas y viceversa.

Ejemplo de traspuesta conjugada de una matriz cuadrada:

**Ejercicio 4 – Archivos (2 puntos):** considerar un archivo de texto de nombre “***datos.txt***” situado en el directorio (carpeta) actual de trabajo que contiene los datos registrados por una estación ITV (Inspección Técnica de Vehículos) de los vehículos que han pasado por la misma en el último semestre, adecuadamente formateados (un vehículo por línea) con la siguiente estructura:

|  |  |
| --- | --- |
| **Archivo “datos.txt”** | **Significado de los datos de cada línea** |
| D 1900 6 0  D 2000 11 101  G 1500 14 10  D 1400 40 0  D 1900 8 0  G 1900 12 100  D 2000 14 111  D 900 4 0  G 1500 21 11  D 1400 4 1  D 1900 6 0  G 1900 11 110  D 2000 14 101  D 900 4 0  G 1500 21 11  … | Col 1: tipo de combustible (G: gasolina, D: diesel)  Col 2: potencia del motor (centímetros cúbicos)  Col 3: antigüedad del vehículo (en años)  Col 4: resultado de la inspección.   * 0: Favorable (no se han detectado defectos) * ≠0: Desfavorable. Resultado de la suma de los siguientes conceptos que indican las deficiencias detectadas:   + 1: fallos en iluminación y acondicionamiento.   + 10: emisiones de humos y frenos.   + 100: pérdidas de líquidos, holguras en dirección, suspensión,… |

Construir un programa en ***C*** que determine e imprima en pantalla:

* Los porcentajes de inspecciones desfavorables debidas a cada una de las tres categorías de deficiencias (aunque no sea exclusivamente).
* Una tabla con la distribución de frecuencias de inspecciones desfavorables tanto para vehículos de gasolina como diesel, agrupando los datos por intervalos de clase de ***4*** años de antigüedad.
* El intervalo de ***4*** años de antigüedad del vehículo en el que se producen más inspecciones desfavorables (moda), tanto para vehículos de gasolina como diesel. Si alguna distribución no fuera mono modal, hay que indicarlo, no siendo necesario en tal caso calcular las distintas modas.

Nota: considerar una antigüedad máxima de ***60*** años y tener en cuenta que la primera inspección técnica del vehículo se produce a los ***4*** años de antigüedad del mismo.