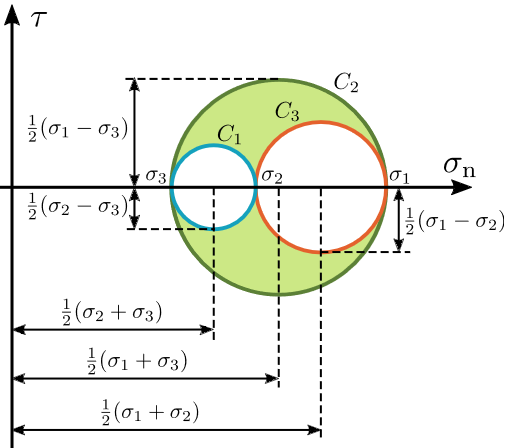
**Fundamentos de Programación (10 de junio de 2019)**

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

**Ejercicio 1 – Programación Estructurada (2 puntos):** construir un programa en ***C*** para determinar los planos en los que las componentes intrínsecas del vector tensión en un punto (tensión normal y tensión tangencial) toman valores dados por teclado, conocido el estado tensional. El vector tensión en un punto, y sus componentes intrínsecas, en el sistema de referencia de las direcciones principales vienen dadas por las siguientes ecuaciones:

siendo **σ1**, **σ2** y **σ3** las tensiones principales (**σ1**≥**σ2**≥**σ3**) y **n=(α,β,γ)** el vector unitario perpendicular al plano con respecto al cual se calcula la tensión, expresado en el sistema de coordenadas de los ejes principales.



Los círculos de Mohr son un método para representar gráficamente el estado tensional de un punto de un sólido con respecto a un plano con vector normal **(α,β,γ)**. Sus componentes intrínsecas **(σn,τ)** se obtienen mediante la intersección de tres circunferencias concéntricas a los círculos **C1**, **C2** y **C3** (ver figura) respectivamente y de ecuaciones:

El programa leerá inicialmente por teclado las tres tensiones principales (debidamente validadas) que representan el estado tensional. Seguidamente solicitará por teclado los valores que toman las componentes normal y tangencial de la tensión en el plano buscado, y comprobará que dichas valores son posibles analizando los círculos de Mohr (el punto **(σn,τ)** debe de estar en la zona sombreada, exterior a los círculos **C1** y **C3**, e interior al círculo **C2**). En tal caso, se calculará el vector unitario del plano solicitado y se imprimirán en pantalla los ángulos que forma con las tres direcciones principales. En caso contrario se imprimirá el correspondiente mensaje de error. Casos especiales a tener en cuenta:

* **σ1**=**σ2**=**σ3** (tensor de tensiones esférico): las tensiones en todas las direcciones son iguales y no tienen componente tangencial (los tres círculos de Mohr se convierten en un punto). El programa no necesita pedir por teclado las componentes intrínsecas de la tensión.
* **σ1**=**σ2** (ó **σ2**=**σ3)**: los tres círculos se reducen a un punto y un círculo, por lo que los únicos valores aceptables de **(σn,τ)** deben de estar sobre la circunferencia C2. En este caso, el programa solo solicitará por teclado la tensión normal debidamente validada, calculará la tensión tangencial e imprimirá en pantalla solamente el ángulo del vector normal al plano con respecto al tercer (ó primer) eje principal.

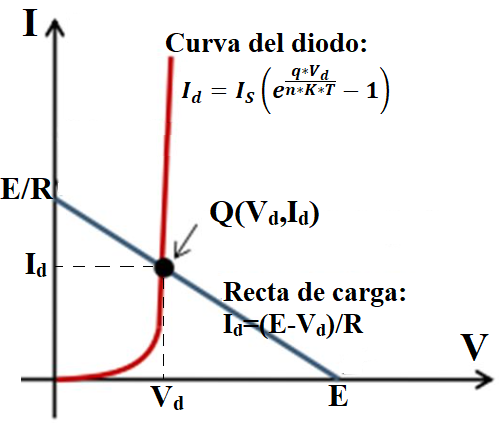
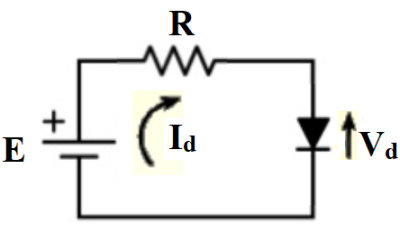
Nota: ecuación de la circunferencia de centro **(xc,yc)** y radio **r**: **(x-xc)2+(y-yc)2=r2**

Distancia entre dos puntos **(x1,y1)** y **(x2,y2):** **d=((x2-x1)2+(y2-y1)2)1/2**

**Ejercicio 2 – Diseño Modular (2 puntos): 2.1)** construir una función en ***C*** que calcule y devuelva el punto de trabajo **Q(Vd,Id)** para la polarización de un diodo con una fuente continua de **E** voltios y una resistencia de **R** ohmios. Considerar que el diodo se encuentra polarizado en directa y en serie con la resistencia, por lo que la corriente es común a la resistencia y al diodo. El punto de trabajo **Q** (intensidad de corriente que atraviesa el diodo **Id** y diferencia de tensión entre sus dos extremos **Vd**) se calcula mediante la intersección de la recta de carga y la curva estática del diodo (fórmula de Shockley). Considerar el siguiente prototipo para dicha función:

void q\_diodo(double e, double r, double n, double t, double \*id, double \*vd);

**2.2)** Construir un programa en ***C*** que presente en pantalla una tabla con los puntos de trabajo de un diodo con n=1.5, a T=273K (0ºC), para valores de E de 5v a 10v (∆E=0.5v) y de R de 20Ω a 100Ω (∆R=10Ω).



Recta de carga:

Modelo de Shockley del diodo:

**Is=10-12A** (corriente de saturación)

**q= 1.602\*10-19C** (carga del electrón)

**n: [1,2]** (coeficiente de emisión)

**k= 1.380648529\*10-23J/K** (cte. de Boltzmann)

**T**: temperatura absoluta de la unión

**Ejercicio 3 – Estructuras de Datos (2 puntos):** Considerar las siguientes estructuras de datos para representar la lista de componentes pasivos de un circuito CA, todos ellos conectados en serie:

#define MAX 100

typedef struct{

char tipo; /\* 'R': resist., 'B': bobina,'C': condens. \*/

union{

double R; /\* resistencia (ohmios) (para Resistencia) \*/

double L; /\* inductancia (Henrios)(para Bobina) \*/

double C; /\* capacitancia (Faradios) (condensador) \*/

};

}tipo\_comp;

typedef tipo\_comp tipo\_array[MAX];

typedef struct{

int n; /\* Nº de componentes registrados en las \*/

tipo\_array v; /\* primeras posiciones del array \*/

}tipo\_lista;

Construir las dos funciones siguientes en C con el prototipo especificado:

|  |  |
| --- | --- |
| void z\_total(tipo\_lista \*l, double f,double \*z, double \*fi); | Calcular la impedancia total del circuito a una frecuencia en hercios dada como argumento. |
| void eliminar(tipo\_lista \*l); | Eliminar de la lista todos los condensadores. |

Nota: impedancias complejas en alterna de cada uno de los componentes pasivos de un circuito:

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Ejercicio 4 – Archivos (2 puntos):** el archivo de texto "enlaces.txt" almacena el conjunto de enlaces de diversas moléculas de **N** átomos cada una. Cada conjunto de enlaces se representa mediante una matriz de adyacencia **NxN**, en las que el elemento **(i,j)** es **1** (si los átomos **i** y **j** están enlazados) y **0** en caso contrario.

Construir un programa que lea el archivo "enlaces.txt" y que calcule los valores medios y el rango de los índices de Zagreb **M1** y **M2** agrupados en función del nº de átomos de la molécula y presente los resultados en pantalla. Nota: considerar solo moléculas con **N** de **3** a **10** átomos, cuya matriz de adyacencia sea válida (simétrica, solo 0 ó 1).

N M1 M2

med min max med min max

3 ... ... ... ... ... ...

.. ... ... ... ... ... ...

10 ... ... ... ... ... ...

**δi**: grado del vértice i (nº de enlaces alrededor de átomo i)

|  |  |
| --- | --- |
|  | "enlaces.txt"  7  0 0 1 0 0 0 0  0 0 1 0 0 0 0  1 1 0 1 0 1 0  0 0 1 0 0 0 0  0 0 0 0 0 1 0  0 0 1 0 1 0 1  0 0 0 0 0 1 0  5  0 1 0 0 0  1 0 1 0 1  0 1 0 1 0  0 0 1 0 0  0 1 0 0 0 |