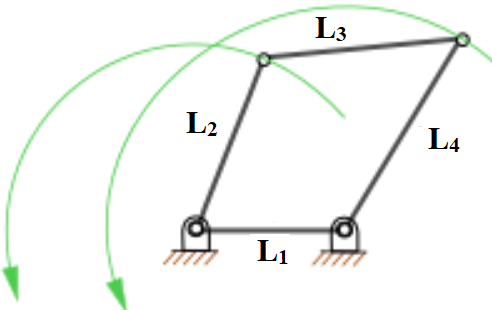
**Fundamentos de Programación (12 de septiembre de 2019)**

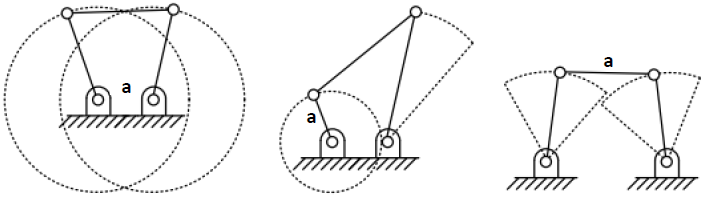
**(Grados en Ingeniería Mecánica, Eléctrica, Electrónica Industrial y Química Industrial)**

**Ejercicio 1 – Programación Estructurada (2 puntos):** construir un programa en ***C*** que lea por teclado una lista de cuadriláteros articulados, y que calcule e imprima en pantalla el porcentaje de los mismos que se

comportan como doble manivela, manivela-balancín y doble balancín. Para cada cuadrilátero, el programa solicitará que se introduzcan por teclado en orden las longitudes de los cuatro lados (**l1**,**l2**,**l3**,**l4**), correspondiendo la primera a la barra fija. A continuación solo analizará aquellos cuadriláteros con lados de longitudes diferentes (**a<b<c<d**) y comprobará para los mismos la desigualdad de Grashof**: b+c>a+d**. En caso de cumplirse, el mecanismo se comporta como:



* Doble manivela si **a** es el elemento fijo.



* Manivela-balancín si **a** es contiguo al elemento fijo.
* Doble balancín si **a** es opuesto al elemento fijo.

El programa terminará de leer datos por teclado cuando

alguna de las cuatro longitudes introducidas sea 0 ó

negativa, imprimiendo a continuación en pantalla los

valores de los porcentajes de cada grupo.

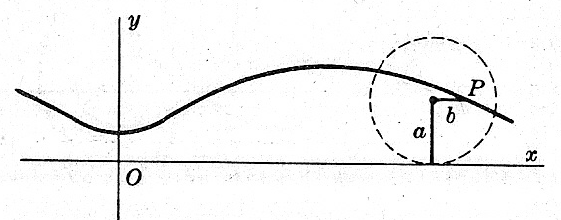
**Ejercicio 2 – Diseño Modular (2 puntos): 2.1)** construir una función en ***C*** que calcule y devuelva el área con su signo encerrada por una función ***f(x)***, el eje de las abscisas y las rectas verticales ***x=a*** y ***x=b*** (integral definida), utilizando la regla de Simpson 3/8 compuesta con ***n*** subintervalos (***n*** múltiplo de ***3***) y con el prototipo siguiente:

/\* h=(b-a)/n xi=a+i\*h i=0,1,2,...,n \*/

double simpson(double (\*f)(double x), double a, double b, int n);

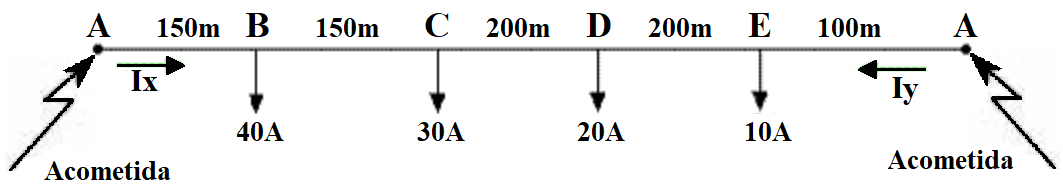
**2.2)** Construir un programa que, haciendo uso de la función anterior, calcule e imprima en pantalla el área de un periodo de una trocoide de radio **a=1** y **b=0.5,** usando **999** subintervalos.

Trocoide: curva descrita por un punto situado a una distancia **b** del centro de una circunferencia de radio a a medida que ésta rueda sin resbalar sobre el eje **X**. Utilizada en el diseño de los dientes de los engranajes. **x = a \* ϕ - b \* sen(ϕ)**



**y = a – b \* cos(ϕ)**

**Ejercicio 3 – Estructuras de Datos (2 puntos):** Considerar las siguientes estructuras de datos para representar una línea eléctrica alimentada por ambos extremos a la misma tensión:



#define MAX 20 /\* Nº máximo de dispositivos receptores \*/

typedef struct{

double i; /\* Intensidad corriente receptor(Amperios) \*/

double fdp; /\* factor de potencia (cos φ) \*/

double r; /\* Distancia al receptor anterior (metros) \*/

}tipo\_receptor;

typedef tipo\_receptor tipo\_array[MAX];

typedef struct{

double d; /\* Longitud total de la línea (metros) \*/

int tipo; /\* 0: cc, 1: ca monofásica, 3: trifásica \*/

double un; /\* Tensión nominal de la línea (voltios) \*/

double cdt; /\* Caida de Tensión máxima (%) \*/

char mat; /\* Material del conductor: 'C':Cu, 'A':Al \*/

int n; /\* Nº de receptores conectados a la línea \*/

tipo\_array v; /\* Características de los receptores \*/

}tipo\_linea2e;

typedef struct{

double ix; /\* Intensidad de corriente de alimentación izq. \*/

double iy; /\* Intensidad de corriente de alimentación der. \*/

int n; /\* Nº de receptor del punto de mínima tensión \*/

}tipo\_carga;

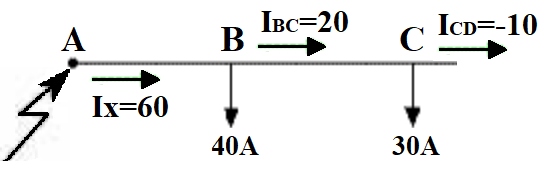
Construir las siguientes funciones en C con el prototipo especificado:

|  |  |
| --- | --- |
| void insertar(tipo\_linea2e  \*l); | Insertar un nuevo receptor en la línea. La función pedirá por teclado el nº del receptor anterior así como su distancia en relación al mismo y el valor de la carga (validando adecuadamente todos los datos), y procederá a insertar el mismo en la estructura de datos. |
| void carga(tipo\_linea2e l,  tipo\_carga \*c); | Calcular la carga que soporta cada uno de los extremos y el punto de mínima tensión (aquel que recibe corriente por los dos lados). |

*Nota 1*: La intensidad en un punto de alimentación (A ó B) se calcula tomando la suma de las intensidades en las derivaciones con relación al otro punto de alimentación y dividiendo por la longitud total:

Ejemplo:

*Nota 2*:El punto de mínima tensión se determina a partir de las corrientes ***Ix*** ó ***Iy***, restando sucesivamente las corrientes que se deriven en cada nudo de la línea, hasta encontrar un valor negativo, y ese nudo es el punto de mínima tensión. Basándonos en la Ley de Nudos de



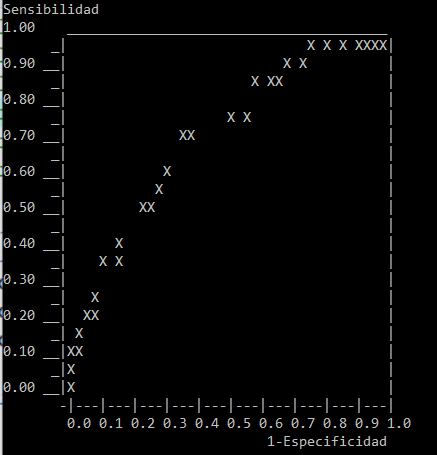
Kirchoff, vamos restando de izquierda a derecha las intensidades

a **Ix**, hasta el primer resultado negativo. Esta intensidad negativa

debe coincidir, tanto si la calculamos de izquierda a derecha

como de derecha a izquierda.

**Ejercicio 4 – Archivos (2 puntos):** construir un programa que represente en pantalla la curva ROC correspondiente a la evaluación de una clasificador inteligente para la detección de anemia ferropénica a partir del volumen corpuscular medio (VCM), y que calcule además el área bajo dicha curva.



Una curva ROC ("Receiver Operating Characteristic") es una representación gráfica de la sensibilidad frente a uno menos la especificidad para un sistema clasificador binario según se varía el umbral de discriminación.

El programa leerá inicialmente los datos del archivo de texto "datos.txt", donde cada línea del mismo contiene dos datos: estado de anemia ferropénica según existencia de depósitos de Fe en médula ósea (0: con anemia, 1: sin anemia) y VCM, y determinará los valores extremos de la variable predictora (mínimo y máximo de VCM). Seguidamente dividirá el rango de la misma en **N=100** intervalos y para cada valor de la misma construirá la matriz de confusión resultante de clasificar valores de VCM mayores o iguales al mismo como **1** e inferiores como **0**, y calculará para cada matriz de error la sensibilidad y especificidad. Finalmente, con los **100** puntos obtenidos, construirá la curva ROC, imprimiéndola en un rectángulo de cuadrículas de pantalla de **20** de alto y **40** de ancho, presentando un asterisco si hay al menos un dato de la curva en dicha cuadrícula

"datos.txt"

0 52.0

0 81.0

...

1 78.1

1 101.3

0 65.8

Matriz de confusión:

**pij**: nº de casos clasificados en la clase **i** que realmente son de la clase **j**