**Fundamentos de Programación**

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

**26 de junio de 2012**

***Ejercicios 1, 2 y 3 (2 p/ejercicio):*** implementar en C solamente lo que se le indica en el enunciado (no hay que construir el programa completo).

***Ejercicio 4 (4.0 p):*** construir un programa en C lo más modular posible (atendiendo a los criterios de modularidad) y documentar el diseño preliminar con el diagrama de módulos (estructura del programa), la definición en C de las nuevas tipologías de datos y los prototipos de las funciones en C, y el diseño detallado con las definiciones de las respectivas funciones en C.

**Ejercicio 1:** Construir en C una función que acepte como argumento una lista de temperaturas (de la tipología especificada a continuación) y que devuelva a través de su identificador el rango intercuartílico de dicha distribución de datos (diferencia entre el tercer y primer cuartil: Q3 - Q1).

|  |  |
| --- | --- |
| /\* Definición de tipos de datos \*/  #define MAX 1000  typedef float tipo\_vector[MAX];  typedef struct{  tipo\_vector v;  int n;  } tipo\_lista; | Nota: dado un conjunto ***ordenado*** de datos, se definen los cuartiles como los tres valores que dividen dicho conjunto en cuatro partes iguales:  25%  25%  25%  25%  Min Q1 Q2 Q3 Max  Estos valores pueden formar parte del conjunto de datos o bien ser valores medios ponderados de dos datos consecutivos. |

**Ejercicio 2:** Implementar en C las estructuras de datos necesarias para representar de forma eficiente los siguientes objetos de información:

Un punto bidimensional: (x,y)

Un triángulo en el plano, dadas las coordenadas de sus tres vértices

Una línea bidimensional, considerando cualquiera de las siguientes posibilidades de especificación:

|  |  |
| --- | --- |
| 1. Dados dos puntos diferentes de la misma: **p1** y **p2**   **(x2,y2)**  **(x1,y1)**  Ecuación de la recta: y=[(y2-y1)/(x2-x1)]\*(x-x1)+y1 | 1. Dada la pendiente de la recta (**m**) y la ordenada en el origen (**b**):   **θ m=tg(θ)**  **b**  Ecuación de la recta: y=m\*x+b |
| 1. Dada la pendiente de la recta **m** y un punto **p1** de la misma   Ecuación de la recta: y=m\*(x-x1)+y1 | 1. Dados los coeficientes **a**, **b** y **c** del polinomio de su representación implícita:   Ecuación de la recta: a\*x+b\*y+c=0 |
| 1. Dado un valor de abscisa **x0** (para el caso especial de rectas verticales):   Ecuación de la recta: x=x0 | 1. Dado un valor de ordenada **y0** (para el caso especial de rectas horizontales):   Ecuación de la recta: y=y0 |
| 1. Dados un punto **p0** y los incrementos horizontal y vertical **∆x** **∆y** de su representación paramétrica:   Ecuaciones de la recta: x=x0+u\*∆x  y=y0+u\*∆y | **Nota**:  m=(y2-y1)/(x2-x1)= ∆y/∆x=-a/b  m=∞ (recta vertical) |

Diseñar e implementar en C las siguientes funciones:

* Una función que acepte como argumento una línea bidimensional (en cualquiera de las representaciones anteriores) y la devuelva en representación implícita.
* Una función que calcule y devuelva el punto de intersección de dos líneas dadas como argumentos. Dicha función deberá devolver a través de su identificador un código que indique el resultado del cálculo (1: líneas secantes, 0: líneas paralelas/coincidentes). Nota: dos líneas son paralelas si tienen la misma pendiente.

**Ejercicio 3:** Utilizando las estructuras de datos definidas en el ejercicio anterior, construir una función en C que calcule y devuelva el orto centro de un triángulo dado como argumento.

Nota 1: el ortocentro de un triángulo es el punto de intersección de sus tres alturas; cada altura se define como el segmento perpendicular a cada arista (o prolongación de esta) que pasa por el vértice opuesto.

Nota 2: dos líneas son perpendiculares si o bien el producto de sus pendientes vale -1 (m1\*m2=-1) o bien una es horizontal y la otra vertical.

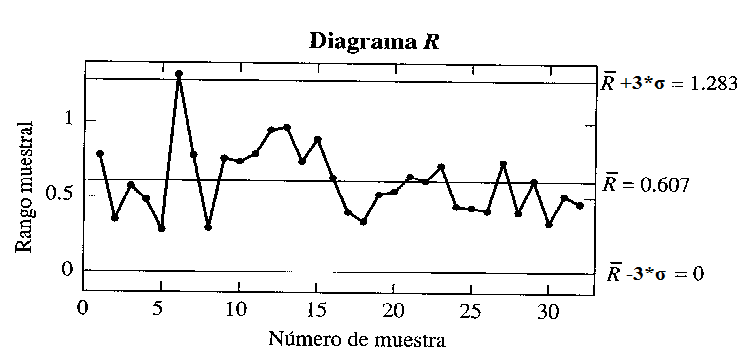
**V2**

**V1**

**V3**

**Ejercicio 4:** Construir un programa que realice un control estadístico de calidad de un determinado proceso industrial mediante un diagrama  e indique si dicho proceso se encuentra en un estado de control según las reglas de la compañía ***Western Electric***. El programa leerá inicialmente de un archivo de texto los datos de las muestras (subgrupos racionales) tomadas de la variable a controlar en un determinado período. La primera línea del archivo contiene el nº de datos de cada muestra ***n*** (todas las muestras tienen el mismo nº de datos). Considerar que como máximo puede haber 100 muestras de hasta 10 datos por muestra.

|  |  |
| --- | --- |
| Archivo de datos: ***muestras.txt*** | Cálculos de medias y rangos de cada muestra:    Rango=Maximo-Mínimo |
|  |  |



Seguidamente se calculará la media aritmética de cada muestra así como su rango (máximo-mínimo), y se eliminarán aquellas muestras cuyo rango esté fuera del intervalo Rmedio±3σrango.(diagrama ***R***)

Finalmente se aplicarán las reglas de ***Western Electric*** a las medias muestrales (diagrama ) y se presentará en pantalla un informe indicando si el proceso está en control o que regla no se cumple, especificando los números de las muestras involucrados.

|  |  |
| --- | --- |
| Media de cada muestra: = ∑**xi**/**n**  Media de las medias: =∑**i**/**N**  Desv. estándar: σ=(∑(**i**-)2/(**N-1))1/2**  N: nº de muestras | Las reglas de ***Western Electric***: un proceso está fuera de control si se cumple alguna de las siguientes condiciones:   1. Un punto cualquiera del diagrama  está fuera de los límites ±3σ. 2. Dos de tres puntos consecutivos del diagrama  están por encima del límite superior +2σ, o dos de tres puntos consecutivos del diagrama están por debajo del límite inferior -2σ. 3. Cuatro de cinco puntos consecutivos del diagrama están por encima del límite superior +σ, o cuatro de cinco puntos consecutivos del diagrama están por debajo del límite inferior -σ. 4. Ocho puntos consecutivos del diagrama caen del mismo lado que la recta central |