**Examen Fundamentos de Informática**

**(Ingeniero Técnico Industrial - Mecánica)**

**7 de junio de 2010**

***Nota***: para poder aprobar el examen hay que obtener una calificación mínima del 50% en cada una de las dos partes del mismo (Preguntas breves/sintaxis de C y Diseño de programas)

***Preguntas breves/sintaxis de C (5.0 p 🡪 1.0 p/pregunta):***

1. Describir brevemente las funciones principales de la unidad de control de un procesador central (CPU). ¿Qué información se almacena en el registro de instrucción (IR) y en el contador de programa (PC)?. ¿Para qué se utiliza el registro del puntero de pila (SP)?. Listar las principales diferencias entre la memoria central y la memoria secundaria de una computadora.
2. Implementar en C una función que calcule numéricamente la integral definida de una función cualquiera mediante la regla de Simpson compuesta, dados como argumentos la función, los límites del intervalo de integración ***a*** y ***b*** y el nº de sub-intervalos en que se divide el intervalo de integración ***n***. Nota: Regla de Simpson compuesta para el cálculo de la integral definida de una función ***f(x)*** en el intervalo ***[a,b]***:

b n-1 n-2

∫f(x) dx ≈ (∆x/3) \* (f(a)+f(b)+4\*∑f(xk)+2\*∑f(xk))

a k=1 k=2

(k impar)

(k par)

∆x=(b-a)/n n🡪 múltiplo de 2 (si n es impar sumarle 1)

x0=a

xk=xk-1+∆x k=1,2,...,n

1. Implementar una función en C que lea por teclado una matriz cuadrada de reales de dimensión máxima 10. La función leerá por teclado previamente la dimensión ***n*** de la matriz, y almacenará los datos leídos en las ***n*** primeras filas y columnas de un “array” bidimensional que se devolverá como resultado junto con la dimensión de la matriz cuadrada.
2. Implementar en C una función recursiva que calcule el determinante de una matriz cuadrada de dimensión ***n***, almacenada como una sub-matriz (***n*** primeras filas y ***n*** primeras columnas) en un “array” bidimensional de dimensiones 10x10, mediante el desarrollo del mismo a través de la primera fila mediante sus adjuntos. La función aceptará como argumento un “array” 10x10 de reales y la dimensión de la matriz ***n***, y devolverá a través de su identificador el valor del determinante. Ejemplos de desarrollo por adjuntos de un determinante:

1. Considerar el archivo ***A***, del que se muestran las claves de sus registros:

A 5 2 3 4 7 10 12 15 27 41 2 4 3 6 8 4 11 22 15

1 2 4 6 8 14 19 21 23 16 EOF

Explicar los pasos (pasadas) y fases (partición/fusión) dentro de cada paso que habría que realizar para clasificar dicho archivo según el valor del campo clave (de mayor a menor valor de la clave) utilizando el algoritmo de clasificación por mezcla directa, mostrando los sucesivos estados de los archivos necesarios para conseguir dicha clasificación.

***Ejercicios: Diseño de programas (5.0 p 🡪 2.5 p/ejercicio):*** plantear una solución lo mas modular posible (atendiendo a los criterios de modularidad) y documentar adecuadamente todas las etapas de diseño (diseño preliminar y diseño detallado).

1. Diseñar un programa que realice operaciones básicas sobre expresiones algebraicas polinómicas de grado máximo 10, mediante la selección de las mismas a través de un menú de opciones. Una expresión algebraica está formada por una serie de términos (monomios), cada uno de los cuales está formado por la multiplicación de un número real (coeficiente) por una variable ***x*** elevada a un número entero no negativo (exponente). Ejemplo: ***4x3+5x-5x3+6+2x2-4x3+2***. En primer lugar, el programa solicitará por teclado dos expresiones algebraicas de hasta un máximo de 100 términos, leyéndose término a término el coeficiente y el exponente correspondiente, hasta que el usuario introduzca ambos valores iguales a cero para indicar que no quiere introducir más términos. A continuación el programa simplificará cada expresión ordenando los términos por valores decrecientes de exponente, agrupando términos semejantes (mismo exponente) y eliminando aquellos cuyo coeficiente sea cero. Seguidamente se presentará en pantalla un menú con las siguientes opciones:

+ Sumar expresiones algebraicas.

\* Multiplicar expresiones algebraicas.

/ Dividir expresiones algebraicas (método de Ruffini: la segunda expresión debe ser de la forma ***x-a***)

0 Finalizar operaciones con estas dos expresiones

Tras la selección de la opción del menú, el programa realizará la acción pertinente (si ello es posible), presentando la expresión resultante (sin simplificar y simplificada) en la pantalla, y volviendo a presentar a continuación el menú de opciones. Tras seleccionar la opción 0, el programa preguntará si se desean efectuar mas operaciones con otras expresiones diferentes, y en caso de respuesta afirmativa se volverá a repetir todo el proceso anterior. Ejemplos:

|  |  |  |
| --- | --- | --- |
| **Suma de expresiones** | **Producto de expresiones** | **División método de Ruffini** |
| Exp1: 4x2-3x3+2+x-2x2-x  Exp2: 7x-2x2-5x-x+1 | Exp1: 4x2+x-2x2-x+1  Exp2: 7x-2x2-5x-x+1 | Dividendo: 5x4+8x3+13x+4  Divisor: x+3 |
| 4x2-3x3+2+x-2x2-x  +  7x-2x2-5x-x+1  =  -3x3+2x2+2  +  -2x2+x+1  =  -3x3+2x2+2-2x2+x+1  =  -3x3+x+3 | 4x2+x-2x2-x+1  \*  7x-2x2-5x-x+1  =  2x2 +1  \*  -2x2+x+1  =  -4x4+2x3+2x2-2x2+x+1  =  -4x4+2x3+x+1 | * Se disponen los coeficientes del dividendo y el término independiente del divisor (a) en una tabla:   5 8 0 13 4  -3 -15 21 -63 150  5 -7 21 -50 154   * Se multiplica el primer coeficiente del dividendo por el valor de ***a*** y se coloca el resultado en la siguiente columna. Se suman los valores de la columna siguiente y se repite el proceso hasta realizar la suma de la última columna. * El último término es el resto de la división. El resto de coeficientes corresponde al polinomio cociente, de un grado inferior al del dividendo:   Cociente: 5x3-7x2+21x-50  Resto: 154 |

1. Una imagen digital consta de una matriz rectangular de píxeles, cada uno de los cuales representa una pequeña área de la imagen completa; de cada pixel se registra un conjunto de valores denominados bandas. La imagen se almacena en memoria secundaria en dos archivos: uno de ellos contiene los diferentes valores de cada pixel para cada una de las bandas, y el segundo archivo (mismo nombre y extensión .cfg) contiene información necesaria para poder leer la imagen. La estructura interna de cada archivo es la que se especifica a continuación:

*Imagen* *Imagen.cfg*

Tipos tipo\_imagen: archivo\_s de real tipo\_cfg: archivo\_s de tipo\_reg\_inf

tipo\_reg\_inf: registro de

lineas: entero {1..1024}

columnas: entero {1..1024}

bandas: entero {1..10}

formato:cadena3 {BSQ,BIL,BIP}

Fin\_registro

cadena3: cadena[4]

Diseñar un programa que permita cambiar el formato de un archivo que contiene los datos de una imagen digital por otro especificado. El programa leerá por teclado el nombre del archivo de datos de la imagen así como el nuevo formato entre una lista de 3 opciones: BSQ, BIL y BIP, procediéndose a modificar in situ (si ello fuera pertinente) el formato de los datos.

Formatos de almacenamiento de una imagen:

* BSQ (“Band Sequential Format”: Formato Secuencial de Bandas): cada línea de datos va seguida inmediatamente por la siguiente línea de la misma banda espectral. Las siguientes bandas se almacenan de manera similar.
* BIP (“Band Interleaved by Pixel Format”: Formato de Bandas Intercaladas por Pixel) : la imagen almacena el primer pixel de todas las bandas en orden secuencial, seguido por el segundo pixel de todas las bandas, seguido por el tercer pixel para todas las bandas,... Al terminar los píxeles de una fila se continúa con los píxeles de la siguiente fila.
* BIL (“Band Interleaved by Line Format”: Formato de Bandas Intercaladas por Línea): la imagen almacena la primera línea de la primera banda, seguida por la primera línea de la segunda banda, seguida por la primera línea de la tercera banda,... Las líneas subsecuentes para cada banda se intercalan de manera similar.