**Examen Fundamentos de Informática**

**(Ingeniero Técnico Industrial - Mecánica)**

**13 de septiembre 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 (4.0 p 🡪 1.0 p/pregunta):***

1. Suponiendo que una computadora utiliza ***n=8*** bits para representar números enteros, ¿cómo se representarían internamente los siguientes números decimales: ***+100***, ***-127***, ***+255***, ***-128***? Considerar las siguientes representaciones:
2. Signo y magnitud. b) Complemento a ***1***. c) Complemento a ***2***. d) Sesgada (desplazada).
3. Implementar en ***C*** una función que acepte como argumento un número natural, si este es par lo divide entre ***2*** y si es impar lo multiplica por ***3*** y le suma ***1***, presentando el resultado en pantalla. Este proceso se repite para el nuevo valor obtenido, finalizándose cuando dicho valor sea ***1***. La función deberá devolver a través de su interfaz el nº de iteraciones de este proceso y el valor máximo generado. Ejemplos:

n=22 🡪 22 – 11 – 34 – 17 – 52 – 26 – 13 – 40 – 20 – 10 – 5 – 16 – 8 – 4 – 2 – 1 Nº Iteraciones: 15 Máximo: 52

n=6 🡪 6 – 3 – 10 – 5 – 16 – 8 – 4 – 2 – 1 Nº Iteraciones: 8 Máximo: 16

n=1 🡪 1 Nº Iteraciones: 0 Máximo: 1

1. Implementar una función en ***C*** que lea por teclado y por columnas una matriz de reales (doble precisión) de dimensión máxima ***10x10***. La función leerá por teclado previamente la dimensiones de la matriz (***nf***: nº de filas, ***nc***: nº de columnas), y almacenará los datos leídos en las ***nc*** primeras columnas y ***nf*** primeras filas de un “array” bidimensional que se devolverá como resultado junto con la dimensiones de la matriz leída. Implementar una segunda función para transponer matrices, que acepte como argumento una matriz y devuelva a través del mismo argumento la matriz traspuesta (intercambiar filas por columnas).
2. Considerar los dos archivos binarios siguientes que contienen respectivamente los datos de los clientes de una compañía eléctrica y las lecturas (kWh) de los contadores que se han podido efectuar el último mes:

***Clientes.bin*** ***Lecturas.bin***

NIF: cadena de 12 caracteres (incluido ‘\0’) Nº contador: entero de 4 bytes

Nº contador: entero de 4 bytes Lectura actual: real (simple precisión)

Lectura mes anterior: real (simple precisión)

Suponer que ambos archivos se encuentran clasificados por nº de contador. Implementar una función en C que acepe como argumentos dos cadenas de caracteres con los nombres físicos de dichos archivos y que genere, con el mínimo número de accesos a memoria secundaria, dos archivos de texto (***consumos.txt*** y ***errores.txt***), el primero de ellos que presente en cada línea el NIF del cliente y el consumo efectuado (diferencia entre la lectura actual y la del último mes) y el segundo con los números de los contadores que no se corresponden con ningún cliente registrado. ***Nota***: si no aparece la lectura actual del contador de un cliente (esta no se pudo realizar) se registra como consumo efectuado 0 kWh.

***Ejercicios: Diseño de programas (6.0 p 🡪 3.0 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 para gestionar las plazas de un “parking” de vehículos. Este se ubica en las 6 plantas inferiores de un edificio (planta 0, sótanos -1 a -5), donde cada planta consta de 10 pasillos etiquetados mediante una letra (A..J) y en cada pasillo hay 20 plazas de aparcamiento numeradas de la 1 a la 20. Hay dos modalidades de alquiler de las diferentes plazas, cuyas condiciones y tarifas son las detalladas a continuación:

* Por meses completos (reserva): 50 €/mes. El número máximo de plazas a alquilar en esta modalidad es del 50% de la capacidad del parking, ocupando estas obligatoriamente las 3 plantas inferiores.
* Por tiempo de ocupación: 2 céntimos/mn (1.2 €/hora) las 5 primeras horas; a partir de 5 horas se factura el día completo (24 horas) a 6 €/día.

El programa deberá presentar un menú en pantalla con las siguientes opciones:

* Nueva reserva de plaza: se registrará el nombre, apellidos, NIF y teléfono de contacto de la persona que va a alquilar una plaza, asignándole una plaza libre si la hubiera en ese momento de las destinadas a este fin (3 últimas plantas). Dicha plaza queda reservada y ya no se puede alquilar por tiempo.
* Baja de reserva: se pedirá el NIF y se procederá a anular la reserva de dicha plaza (esta pasa nuevamente a estar disponible para el alquiler por tiempo).
* Entrada de vehículo en parking: se leerá en primer lugar la matrícula del vehículo y se preguntará si tiene plaza reservada, en cuyo caso se leerá el NIF y se hará la comprobación correspondiente localizándose la plaza reservada y el estado de ocupación de la misma. Si se trata de un alquiler por tiempo, se comprobará si hay plazas libres en cuyo caso se le asignará la más próxima a la entrada (planta, pasillo y nº de plaza), registrándose el valor actual del cronómetro de tiempos (ver nota). En ambos casos se registrará la plaza reservada/asignada como ocupada.
* Salida de vehículo del parking: se leerá la matrícula del vehículo y se comprobará si se trata de una reserva o de un alquiler por tiempo. En este último caso se presentará en pantalla el importe de la factura a abonar tras leer el valor del cronómetro de tiempos. En ambos casos el estado de ocupación de la plaza pasa a estar libre.
* Estado ocupación actual: se presentará en pantalla un listado con el total de vehículos actualmente aparcados, desglosando este listado por plantas y por pasillos.
* Listado de plazas reservadas, ordenado alfabéticamente por apellidos y nombre, presentando en cada línea del mismo los apellidos, nombre, NIF y teléfono del cliente junto con la identificación de su plaza de aparcamiento (planta, pasillo y número).
* Fin de la ejecución del programa.

Nota: suponer que se dispone de los dos módulos siguientes:

***Procedimiento leer\_matricula(mat: cadena10 (S))*** 🡪 ***mat***: nº de la matrícula del vehículo que pasa por el sensor.

***Procedimiento leer\_cronometro(mn: entero S))*** 🡪 ***mn***: minutos transcurridos desde que se inicializó el cronómetro.

1. Diseñar un programa para factorizar matrices cuadradas de dimensión máxima ***10*** utilizando la descomposición ***LU***. Factorizar una matriz ***A*** consiste en descomponerla en dos matrices ***L*** y ***U*** cuyo producto sea la matriz original: ***A=L\*U***; en la descomposición ***LU*** la matriz ***L*** es una matriz triangular inferior (los elementos por encima de la diagonal principal son ***0***) y la matriz ***U*** es una matriz triangular superior (los elementos por debajo de la diagonal principal son ***0***). Si la matriz cuadrada ***A*** tiene todos sus menores principales no nulos, entonces se puede descomponer de esta forma. El programa leerá por teclado la dimensión de la matriz (máximo ***10***) y los elementos de la misma, realizará si ello es posible la factorización ***LU*** y presentará en pantalla las dos matrices ***L*** y ***U*** (o un mensaje indicando que no se puede realizar esta descomposición). Nota: el menor principal ***i*** de una matriz es el determinante de la matriz formada por la ***i*** primeras filas e ***i*** primeras columnas de dicha matriz.

|  |  |  |
| --- | --- | --- |
| **Ejemplo: Descomposición LU** | | |
| 1 0 2 -1 1 0 0 0 1 0 2 -1 (f1)  3 1 1 0 = 0 1 0 0 \* 3 1 1 0 (f2)  -1 1 -6 2 0 0 1 0 -1 1 -6 2 (f3)  0 -1 1 1 0 0 0 1 0 -1 1 1 (f4) | | La matriz inicial se descompone inicialmente en el producto de la matriz identidad por si misma­: A=L\*U, con L=I y U=A  Objetivo: convertir U en una matriz triangular superior. |
| 1 0 0 0 1 0 2 -1  0 1 0 0 3 1 1 0 (f2)-3\*(f1)/1  0 0 1 0 -1 1 -6 2 (f3)-(-1)\*(f1)/1  0 0 0 1 0 -1 1 1 (ya vale 0) | | Empezamos haciendo ceros los elementos no nulos de la columna 5 a partir de la fila 2, sustituyendo cada fila por la combinación lineal de filas indicada |
| 1 0 0 0 1 0 2 -1  -3 1 0 0 0 1 -5 3  1 0 1 0 0 1 -4 1 (f3)-1\*(f2)/1  0 0 0 1 0 -1 1 1 (f4)-(-1)\*f2/1 | | Seguimos haciendo ceros los elementos no nulos de la columna 6 a partir de la fila 3, sustituyendo cada fila por la combinación lineal indicada. Nótese que la primera columna no se modifica. |
| 1 0 0 0 1 0 2 -1  -3 1 0 0 0 1 -5 3  1 -1 1 0 0 0 1 -2  0 1 0 1 0 0 -4 4 (f4)-(-4)\*f3/1 | | Terminamos haciendo cero el elemento no nulo de la columna 6 a partir de la fila 4, sustituyendo cada fila por la combinación lineal indicada y sin modificar las dos primeras columnas. |
| 1 0 0 0 1 0 2 -1  -3 1 0 0 0 1 -5 3  1 -1 1 0 0 0 1 -2  0 1 4 1 0 0 0 -4 | | Para obtener la matriz L cambiamos el signo a los elementos del triángulo inferior de la primera matriz. |
| 1 0 0 0 1 0 2 -1  L = 3 1 0 0 U = 0 1 -5 3  -1 1 1 0 0 0 1 -2  0 -1 -4 1 0 0 0 -4 | |  |
| **Ejemplo: Cálculo del determinante de una matriz mediante el método de Gauss** | | |
| 2 0 4 -2  3 1 1 0 (f2)-3\*(f1)/2  -1 1 -6 2 (f3)-(-1)\*(f1)/2  0 -1 1 1 (ya es cero) | Objetivo: convertir la matriz en una matriz triangular sustituyendo cada fila por la suma de la propia fila con una combinación lineal de las otras filas. Empezamos haciendo cero todos los elementos no nulos de la primera columna, excepto el de la primera fila. (\*) | |
| 2 0 4 -2  0 1 -5 3  0 1 -4 1 (f3)-1\*(f2)/1  0 -1 1 1 (f4)-(-1)\*(f2)/1 | Seguimos haciendo cero todos los elementos no nulos de la segunda columna, excepto el de la segunda fila.(\*) | |
| 2 0 4 -2  0 1 -5 3  0 0 1 -2  0 0 -4 4 (f4)-(-4)\*(f3)/1 | Continuamos haciendo cero todos los elementos no nulos de la tercera columna, excepto el de la tercera fila.(\*) | |
| 2 0 4 -2  0 1 -5 3  0 0 1 -2 det=2\*1\*1\*(-4)=-8  0 0 0 -4 | El determinante se obtiene multiplicando todos los elementos de la diagonal principal. | |
| (\*) Si este último fuera nulo, habría que buscar una fila posterior que tuviera un valor no nulo en dicha columna e intercambiar las filas (si no se encuentra dicha fila posterior entonces el determinante vale cero). | | |