



FUNDAMENTOS DE COMPUTACIÓN ICM00794



"Impulsando
la sociedad
del
Conocimiento"

[Principal] [Curso] [Material] [Tareas] [Exámenes] [Proyectos] [Políticas] [Soluciones]

EXÁMENES FINALES (2001-2003)

Final III Término 2003 - 2004. Abril 23, 2004

1. En un juego se lanza un dado 5 veces y se gana cuando la suma de puntos obtenida en los dos primeros lanzamientos es igual a la obtenida en los restantes tres.

- Escriba una *función*, denominada juego, que **simule** los cinco lanzamientos del dado y **retorne** dos posibles valores: **1**, si se trata de un jugada ganadora y **0** en caso contrario.

- Escriba un *programa* en C/C++ que permita realizar 1000 jugadas diferentes sucesivas, mediante llamadas a la función juego, y que **muestre por pantalla el porcentaje de jugadas ganadoras**.

2. La raíz cuadrada de un número real **X** mayor que 0 se puede obtener a través de una aproximación **n - ésima** de una función **f** según el método de Newton, el cual establece que:

$$f(1) = X/2$$

$$f(2) = 0.5(f(1) + X/f(1))$$

$$f(3) = 0.5(f(2) + X/f(2))$$

.....

$$f(n) = 0.5(f(n-1) + X/f(n-1))$$

- Escriba una *función* recursiva **f** que reciba dos parámetros: **X** (el número del cual se desea calcular la raíz cuadrada) y **n** (el número de aproximaciones). Esta función debe retornar la raíz cuadrada de **X** para la **n - ésima** aproximación.

- Escriba un *programa* en C/C++ que permita el ingreso de un número real y que, mediante sucesivas llamadas a la función **f**, muestre los resultados del cálculo de la raíz cuadrada de dicho número para cada una de las 10 primeras aproximaciones ($n = 1, 2, 3, \dots, 10$).

3. Dados dos vectores **U** y **V**, se define la distancia de Hamming, **d(U,V)**, como el número de posiciones en las cuales difieren los vectores **U** y **V**.

Por ejemplo, si **U = 20001** y **V = 10103**, **d(U,V) = 3**, ya que existen tres posiciones (la primera, la tercera y la quinta) en las cuales los dos vectores tienen diferente valor.

- Escriba una *función*, denominada **Hamming**, que reciba como parámetros dos cadenas de caracteres y **retorne un entero** que indique la distancia Hamming entre ellas. Si las cadenas no tienen longitudes iguales la función **retornará -1**.

- Escriba un *programa* en C/C++ que pida al usuario el ingreso de una clave (cadena de caracteres), que a continuación lea otra cadena almacenada en un archivo denominado **clave.dat** ya existente en la unidad de disco C y que, después de llamar a la función **Hamming**, muestre por pantalla un mensaje indicando si la cadena ingresada coincide o no con la cadena almacenada en el archivo **clave.dat**.

4. En un almacén se requieren almacenar los datos correspondientes a los diferentes artículos disponibles en un archivo denominado **invent.dat** en la unidad de disco C. Para cada artículo se deben registrar los siguientes datos:

código (entero),
nombre (cadena de hasta 30 caracteres),
precio (número real),
cant (número entero que refleja la cantidad existente).

- Escriba una *función*, denominada **ingresar_datos**, la cual permitirá registrar los datos de varios artículos en el archivo **invent.dat** a través de una estructura. Los códigos de cada registro se irán generando secuencialmente a partir de 1 y los restantes datos serán digitados por el usuario, mientras aquel desee continuar con el ingreso de más artículos. Al final la función retorna el número de artículos ingresados.

- Escriba un *procedimiento*, denominado **venta_item**, que reciba 2 parámetros: el código de un artículo y el número de unidades que serán vendidas. Este procedimiento debe acceder en el archivo **invent.dat** al artículo cuyo código se desea vender y mostrar por pantalla el importe de la venta de tal artículo. Si la cantidad de artículos existentes es menor que la cantidad solicitada para la venta, se debe mostrar un mensaje que indique que la venta no puede ser realizada.

- Escriba un *programa* en C/C++ que llame a la función **ingresar_datos**, que luego pida al usuario el ingreso del código de un artículo y el número de unidades que se desean vender de dicho artículo y que, finalmente, llame al procedimiento **venta_item**.

Final II Término 2003 - 2004. Febrero 10, 2004

Tema No.1

a) Escriba una *función* ordenada en C/C++ que reciba como parámetros **tres números enteros** distintos **a**, **b** y **c**, y los modifique e intercambie internamente de modo que al finalizar la ejecución de la función se cumpla que los contenidos de **a**, **b**, **c** sean tales que **a<b<c**.

Las variables **a**, **b** y **c** utilizadas No deben ser globales. Use punteros.

b) Escriba una *función* **dado_aleatorio** en C/C++ que reciba como parámetro un número entero **n** que representa una de las posibles caras de un dado ($1 \leq n \leq 6$) y simule la cantidad de veces que se lanza el dado (aleatoriamente) hasta que se obtenga el número **n**. Escriba un *programa* de prueba que muestre la cantidad de veces que se necesitan hasta que salga cada una de las caras superiores del dado e indique cuál fue la cara más difícil (que necesitó más intentos para salir).

Tema No.2

Escriba un *programa* en C/C++ que permita almacenar en una matriz las primeras **n** filas, $n < 20$, del triángulo de pascal:

1					
1	1				
1	2	1			
1	3	3	1		
1	4	6	4	1	
1	5	10	10	5	1
...

A partir de la tercera fila, los elementos dentro del triángulo, se obtienen sumando los dos elementos anteriores de la fila anterior. Por ejemplo, $6=3+3$. Muestre por pantalla la matriz generada.

Tema No.3

Los números de Eudoxus se definen mediante la siguiente formulación:

$$\begin{cases} x_i = y_i + y_{i-1}, & i \geq 1 \\ y_i = x_{i-1} + y_{i-1}, & i \geq 1 \\ x_0 = 1 \\ y_0 = 0 \end{cases}$$

- Escriba las *funciones* recursivas **eudoxusX(i)** y **eudoxusY(i)** que permitan generar cada par ordenado (x_i, y_i) .
- Adicionalmente escriba un *programa* de prueba que, utilizando las funciones anteriores, muestre los pares ordenados generados recursivamente para $0 \leq i \leq n$, donde **n** es un valor entero leído previamente.

Tema No. 4

- Escriba una *función* **cuenta_palabra** en C/C++ que reciba una cadena de caracteres y determine **cuántas palabras** contiene. Suponga que las palabras están separadas por un espacio en blanco.

- Escriba un *programa* de prueba en C/C++ que llame a dicha función para determinar **cuántas palabras** contienen las primeras 20 líneas del archivo de texto "A:\datos.txt". El programa también deberá indicar cuál es la palabra más larga de entre las contenidas en esas 20 primeras líneas.

Final I Término 2003 - 2004. Septiembre 02, 2003

TEMA No. 1 (25 puntos)

- Escriba 3 *funciones* en C/C++, denominadas **promedio**, **mayor** y **menor**, las cuales reciben como parámetro un arreglo de 12 elementos de tipo float y retornan, respectivamente, el promedio de los elementos del vector, el mayor de los elementos del vector y el menor de los elementos del vector.

- Escriba un *programa* en C/C++ que almacene en un arreglo las temperaturas medias de los 12 meses del año (datos de tipo float ingresados desde el teclado) y que luego llame a las 3 *funciones* anteriores a fin de mostrar por pantalla la temperatura anual promedio y la diferencia de temperaturas entre el mes mas caluroso y el mes mas frío.

TEMA No. 2 (25 puntos)

- Escriba una *función* recursiva **f** en C/C++ tal que:

$$f(n) = \begin{cases} 1/2, & \text{si } n=0 \text{ ó } n=1 \\ 1/2 * [f(n-1) + f(n-2)], & \text{si } n \text{ es impar mayor que } 1 \\ 1/2 * [f(n-1) - f(n-2)], & \text{si } n \text{ es par mayor que } 1 \end{cases}$$

- Escriba un *programa* en C/C++ que determine el mayor valor de la función **f** para **n=0, 1, 2, 3, 4, 5**

TEMA No. 3 (25 puntos)

- Escriba un *programa* en C/C++ que lea una frase con un máximo de 80 caracteres y la encripte de acuerdo a las siguientes reglas:

1. Recorrer la cadena en grupos de 3 caracteres e intercambiar el primero con el tercero.
2. Si el último grupo tiene menos de 3 caracteres no se modifica.

- Para encriptar la frase diseñe una *función* que reciba como parámetro dicha frase y la encripte. El programa debe mostrar la frase encriptada.

TEMA No. 4 (25 puntos)

- Escriba un programa en C/C++ que cumpla con los siguientes requisitos:

1. Lea desde el teclado y almacene en un archivo a:\carros.dat los siguientes datos de un total de **NUM_CARS**

(constante entera global) vehiculos:

Placa: (Cadena de 6 caracteres)
Año: (entero)
Tipo: (A: auto, C: camioneta)
Marca: (cadena de 10 caracteres)
Color: (cadena de 10 caracteres)
Precio: (dato real en dólares)

2. Muestre por pantalla las placas de autos cuyos datos están almacenados en a:\carros.dat , tales que su año sea mayor a 1995, su color sea rojo y su precio menor a 6000 dólares.

Final II Término 2002 - 2003. Febrero 13, 2002

TEMA No. 1 (20 puntos)

- Defina una *función* llamada **perfecto(x)** que retorne **1** si **x** es un número perfecto o **0** en caso de que no lo sea. Un número perfecto es aquel que es igual a la suma de todos sus divisores, con excepción del mismo.

Ej: 6 es perfecto, porque sus divisores son: 1, 2, 3 (6 no se considera).

- Para probar la *función* genere en el *programa* principal **m** pares de números aleatorios con valores entre 1 y **n** inclusive (**m** y **n** deben ser previamente pedidos por teclado), y muestre cuántas de estas parejas tienen su suma igual a un número perfecto.

TEMA No. 2 (35 puntos)

El archivo a:\DATOS1.TXT contiene la información de los clientes del banco A en el siguiente formato: char cedula[10], char nombre[60], int cuenta (valor lógico que indica si tiene una cuenta corriente, lo contrario significa que tiene cuenta de ahorros).

El archivo a:\DATOS2.TXT contiene la información de los clientes del banco B en el siguiente formato: char cedula[10], char nombre[60], int prestamo (valor lógico que indica si tiene un préstamo, lo contrario significa que no tiene deudas).

Elabore un programa que

- Lea los 2 archivos desde el disquete. (10 pts)
- Seleccione los clientes que tienen cuenta de ahorros y préstamo. (15 pts)
- Muestre un reporte por pantalla con la cédula y el nombre de los clientes que tienen cuenta de ahorros y préstamo. (10 pts).

Nota: Dentro de un mismo archivo no existen clientes repetidos. El número de clientes máximo es 100, pero el número puede ser menor de acuerdo a si el cliente tiene o no préstamo o cuentas en el banco y puede ser leído en el archivo antes de leer el primer cliente.

TEMA No. 3 (20 puntos)

Una compañía de telecomunicaciones tiene sospechas de que sus datos están siendo interceptados y desea transmitirlos de manera segura usando algoritmos de encriptación de tal forma que solo al pueda entender el destinatario. Toda su información se transmite como enteros de 4 dígitos. El asesor de seguridad informática le recomienda seguir los siguientes pasos:

- A cada digito súmele siete. Al resultado de esta suma, divídalo para 10 y extraígale el residuo. El valor resultante reemplaza al digito original
- Intercambie el primer digito con el tercero y el segundo con el cuarto.

Escriba una *función* **encripta** que reciba numero entero de 4 dígitos (entre 1000 y 9999) y que retorne el mismo numero entero encriptado correspondientemente

Ejemplo: **encripta(1254)** --> 2189

Escriba un *Programa* que pida al usuario una clave de acceso (entero de 4 dígitos), verifique que sea válida en rango). Encripte la clave de acceso y la muestre en pantalla.

TEMA No. 4 (25 puntos)

En el juego del "Imitador" pueden participar dos personas para probar su memoria en el que por medio de turnos alternos se prueba la capacidad de recordar una secuencia numérica entera (0-9) de cada jugador.

Se empieza mostrando un numero aleatorio entero. El primer jugador solo deberá digitar este numero, luego se muestra un segundo numero aleatorio, se cambia el turno y el otro jugador deberá digitar el primero y el segundo numero aleatorio. Se muestra un nuevo numero aleatorio y el siguiente jugador deberá digitar los números aleatorios anteriores mas el último mostrado. El juego continua hasta que uno de los dos participantes se equivoca en la secuencia y se declara al otro participante como ganador.

El juego presenta en pantalla para cada jugada lo siguiente:

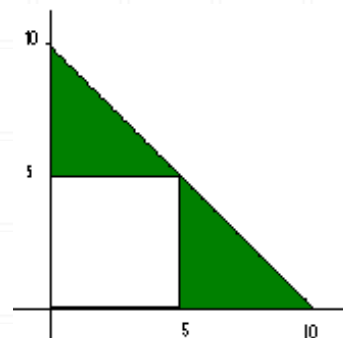
- El jugador que tiene el turno,
- La cantidad de números de la secuencia a digitar y El orden del dígito que se esta esperando por teclado.

Final II Término 2001 - 2003. Febrero 5, 2002

1.- Considere las funciones de variable real: $f(x)=xe^x+1$ y $g(x)=x\text{Sen}x$.

- ◆ Escriba dos funciones en C/C++ que definan las reglas de correspondencia para f y g respectivamente.
- ◆ Elabore un programa principal en C/C++ que permita encontrar el menor y el mayor valor de la función $(f \circ g)(x) = 0.1, 0.2, 0.3, \dots, 2.9, 3.0$

2.- Escriba un programa en C/C++ que genere aleatoriamente valores enteros para 100 pares ordenados (x, y) tales que $0 \leq x \leq 10, 0 \leq y \leq 10$. El programa deberá escribir **CUANTOS** puntos se encuentren dentro del área sombreada mostrada en la figura:



3.- El flujo de efectivo F_k se evalúa recursivamente considerando el flujo neto anterior F_{k-1} y las tasas i_1, i_2 tal como se indica:

$$F_k = \begin{cases} (1+i_1)F_{k-1} + C_k, & F_{k-1} \geq 0, k > 0 \\ (1+i_2)F_{k-1} + C_k, & F_{k-1} < 0, k > 0 \\ 0, & k = 0 \end{cases}$$

En la fórmula, C representa un arreglo de los valores del flujo para n años que operan en un determinado negocio (los valores negativos indican desembolsos) Ej:

-500	300	600	-200	300	-200	-300	350
C_0	C_1	C_2	C_3	C_4	C_n

- Escriba una función recursiva para poder utilizar F_k .
- Escriba un programa principal que lea i_1, i_2 , y el arreglo C_k , de n elementos y llame a la función anterior para calcular F_k .

- Posteriormente, en el mismo programa asigne a i_1 , el valor de 0.1 y pida por teclado 2 valores para i_2 . Evalúe F_K para los casos e identifique si hubo un cambio de signo.

Nota.- Considere las variables *float* C[20], i_1 , i_2 como globales.

4.- En una clínica existen 100 habitaciones. Los pacientes al ingresar a la clínica se les asigna una habitación, mientras que al darles de alta se deja libre la habitación.

a) Crea dos estructuras de datos en C/C++, para lo cual se conoce lo siguiente:

- Habitación: Código (entero), Ocupada (1 carácter, S=Sí/N=No)
- Paciente: Cédula (10 caracteres), Nombre (50 caracteres), código de habitaciones (entero).

b) Escriba un programa en C/C++ que permita ingresar la información de pacientes y habitaciones, con el siguiente menú:

- Ingresar habitaciones
- Ingresar y asignar pacientes a las habitaciones
- Desasignar pacientes de las habitaciones

Nota.- Las restricciones son las siguientes · No se puede asignar un paciente a una habitación que está ocupada · No se puede desasignar un paciente de una habitación que no la ocupe.

Actualizado: 20/01/2009

Revisión: 7

©2008 ICM-ESPOL

Comentarios:
edelros@goliat.espol.edu.ec