

Programación de Computadores

Examen 2002

1.- Escriba un programa que calcule el dígito verificador de una Cédula de Identidad. El dígito verificador puede ser 0,1,2, ..., 8, 9, k. Se obtiene mediante los siguientes pasos (algoritmo).

Primero: Multiplicar los dígitos del número desde el último (o menos significativo) hasta el primero (o más significativo) por los números 2, 3, 4, 5, 6, 7. Si aún faltan dígitos del número por multiplicar, se reinicia la serie desde 2 y hasta 7.

Ejemplo: 7.201.989

entonces:	7	2	0	1	9	8	9
debe ser multiplicado							
por	2	7	6	5	4	3	2

Segundo: Sumar los resultados de cada multiplicación:

$$14+14+0+5+36+24+18=111$$

Tercero: La suma se divide por 11 para obtener el residuo. Si el residuo es 0, entonces el dígito verificador es 0 y el proceso termina.

$$111:11=10$$

1 (residuo)

Cuarto: Si el resto (o residuo) es distinto de 0, al divisor se le resta el resto de la división. El valor de la resta es el dígito verificador, si la resta es 10 el dígito verificador es k.

Nota: las entradas al programas son caracteres.

(1.8 ptos.)

2.- Para la siguiente estructura resuelva:

```
struct _fecha
{
    int dia;
    int mes;
    int anho;
};
typedef struct _fecha fecha;

struct _obrero
{
    int codigo;
    char nombre[25];
    char apellido[25];
    int edad;
    fecha fnac;
    fecha fing; /*fecha de ingreso a la empresa*/
    int coddepto;
};
typedef struct _obrero obrero;

struct _dpto
{
    int coddepto;
    char nomdepto[25];
};
typedef struct _depto depto;

struct _sueldo
{
    int codigo;
    fecha fecpago;
    long monto;
    long extra;
    long total;
};
typedef struct _sueldo sueldo;
```

Asuma que existen arreglos declarados, ya sea global o localmente. El arreglo de sueldo, el cual guarda los sueldos de los ultimos 10 años.

a.- *Indique que departamento posee la planilla de sueldos mayor, es decir, que depto. gasta más en remuneraciones anualmente, tome en cuenta el periodo 2001-2002.*

(1.0 ptos.)

b.- *Muestre un listado los los obreros que entraron a la empresa antes del 20 de Feb. De 1995.*

(0.4 ptos.)

3.- Conocimientos previos:

En C la entrada y salida estandar puede ser dirigida a archivo, esto simplemente usando los caracteres < o >. Por ejemplo, supongamos que se tiene un programa que lee 3 caracteres y los va imprimiendo en pantalla. El ingreso de los caracteres puede haber sido hecha con `getchar()` o `scanf()`. Si en vez de hacer la entrada por teclado, se crea un archivo que contiene los tres caracteres (suponga "abc") y luego el programa se ejecuta de la siguiente manera:

```
$a.out<arch
a
b
c
$
```

Cree un programa al cual se le pasa un archivo que es un programa escrito en C, este contiene ciclos (for y while). Ud. debe determinar si los ciclos estan correctamente escritos (terminados), además indicar la línea donde se produce el error.

Simplificación: tome en cuenta solamente cuando los ciclos estan bien iniciados y terminados.

Recuerde que un archivo termina cuando el caracter leído es la constante EOF.

(1.0 ptos.)

4.- Se tiene una matriz de MAX x MAX, el cual representa un tablero para un juego denominado Nibbles.

Este consiste en mover un objeto (supongamos el símbolo '+'), según las teclas que pulse el usuario (Arriba, Abajo, Derecha, Izquierda).

Existen MAXOBJ objetos que representan Banderas (de 1 a MAXOBJ, en números o letras), el objeto que está en movimiento debe ir tomando cada una de las banderas, pero en el orden correspondiente, de 1 a MAXOBJ.

El jugador finaliza una vez que haya tomado todas las banderas.

El jugador pierde si choca con:

- las paredes (fuera de los rangos de la matriz)
- alguna bandera, si esta no es la que corresponde.
- Su cola (el mismo objeto).

Debe ir mostrando la matriz por cada movimiento hecho por el jugador.

Simplificaciones:

- Puede usar pseudocódigo en caso de funciones muy tediosas de implementar.
- Suponga existe una función obtiene tecla la cual entrega 1, 2, 3 o 4 si es Izq., Der., Arr. y Aba.
- Ud. Especifique le tipo de dato de la matriz.
- En el juego original el objeto va creciendo cada vez que toma una bandera. Para disminuir la dificultad, el objeto siempre va a ser de tamaño 1, sin embargo, igual tiene que implementar el caso de chocar con si mismo.

					I				
			+						
		H							
					F			G	
	E		J						
					D				
				B					C
A			K						

(1.8 ptos.)

Buena Caza

Profesor : Roberto Uribe P.