

PARTE I - Complete el programa (26 puntos)

Complete las secciones indicadas con "\_\_\_\_" con el código correspondiente. Complete las secciones indicadas con "\_\_\_\_(\*)\_\_\_\_" con una o más líneas de código. El resto de las líneas deben quedar sin modificar.

1.1 Completar el código de manera tal que la función mostrar\_archivo muestre por pantalla el contenido completo del archivo de texto "numeros.csv"

```
void mostrar_archivo()
{
    int caract, num;
    FILE *ptrArchivo = _____;
    if (ptrArchivo != NULL)
    {
        caract=fgetc(ptrArchivo);
        while (_____)
        {
            if (caract!=EOF)
            {
                _____(*)_____;
                fscanf(_____, "%d\n", &num);
                printf(" %d\n", num);
            }
        }
        _____;
    }
    fclose(ptrArchivo);
}
```

```
typedef struct t_nodo
{
    Ejemplo archivo:
    {
        int pedro,1\n
        stru juan manuel,2\n
        stru ...
    } t_nodo
```

1.2 Completar el código de manera tal que la función agregue un entero al final de una lista doblemente enlazada.

```
_____ agregar(t_nodo* lista, int valor)
{
    t_nodo aux;
    if (_____)
    {
        *lista = nuevo_nodo(valor);
    }
    else
    {
        for(_____)
        {
            aux->sig = nuevo_nodo(valor);
            _____(*)_____
        }
    }
}
```

PARTE II - Opción múltiple (26 puntos, solamente si están todas correctas)

Indique la opción correcta (sólo una por cada apartado)

2.1

```
int retorno(int valor)
{
    if( valor != (valor&&valor) )
    {
        retorno(valor&valor);
        if(valor!=1)
            return 144;
    }
    return 266;
}
```

¿Cuál de los siguientes valores pasados como parámetro podría hacer que la función retorno no retorne?

- a) 1 b) 0 c) 2 d) Cualquiera de los anteriores  
e) Ninguno de los anteriores

2.3

```
unsigned char a;
a= ~( (~0) << (1<<2));
```

¿Cuánto termina valiendo la variable a?

- (A) 240 (B) 1 (C) 9 (D) 15 (E) 14 (F) -240 (G) 4

2.2

```
#include <stdlib.h>
int main()
{
    int a[]={1,3,5,7,9,0};
    int *uno=NULL, dos=0;
    uno = a;
    dos= *(uno + *uno);
    printf("dos: %d\n", dos);
    return 0;
}
```

¿Cual es el valor que imprime el programa?

- a) 1 b) 2 c) 3 d) 4 e) error de compilación  
f) Ninguna de las anteriores

2.4

```
.
.
.
if ( (a << b) > a )
    printf("rojo");
else
    printf("negro");
```

Suponiendo que tanto a como b son int y tienen valores enteros positivos distintos de cero, ¿qué muestra el programa?

- A) Siempre muestra "rojo" b) Siempre muestra "negro" c) Depende de los valores de a y b  
d) Depende sólo del valor de a  
e) Depende sólo del valor de b  
f) Ninguna de las anteriores

PARTE III - Programación (48 puntos)

Una red social dispone de los siguientes archivos:

Archivo	Contenido
Usuarios.dat	Registros: ID_usuario (int) nombre (char[60]) sexo(char) ('M' o 'F') edad(int)
Amigos.txt (*)	Archivo CSV con: ID_usuario (int) ID_usuario (int)

(\*) El archivo Amigos.txt indica qué usuario es amigo de qué otro en la red social. La relación de amistad es bidireccional, es decir que si el usuario 101 es amigo de 203, también 203 es amigo de 101, pero en este archivo sólo aparece un renglón que lo indique, con los números de usuario en cualquier orden.

Realizar las siguientes funciones:

- Hacer una función que reciba un ID\_usuario, un char que representa el sexo (siendo F Femenino, M Masculino y X cualquiera de los dos) y un árbol (que podría estar vacío) y agregue a dicho árbol todos los amigos de ese ID\_usuario del sexo recibido como parámetro. En el árbol no deben quedar usuarios repetidos.
- Hacer una función que reciba un ID\_usuario y que, utilizando la función del punto anterior, genere y retorne un árbol con “los amigos de los amigos” de dicho ID\_usuario, que sean del sexo opuesto.

---

Condiciones para todos los ejercicios:

- Para aprobar el examen se debe sumar un mínimo de 65 puntos.
  - Declarar todos los “struct” y “typedef” que utilice para resolver el ejercicio.
  - No desperdiciar memoria. Liberar todos los espacios que se hayan reservado y que no se utilicen.
  - Se pueden programar funciones adicionales, respetando los puntos anteriores.
  - Considere que las estructuras recibidas como parámetro por las funciones pueden estar vacías (nulas).
  - Si utiliza pilas y colas deben respetar el protocolo y el método elegido para su implementación a lo largo de todo el examen.
-