

Fundamentos de Programación -CC112

Práctica Dirigida 9

Estructuras, puntero a estructura, paso de estructura como paso de función, archivos

Estructuras

1 ¿Cuál es la salida del siguiente código?. Explica tu respuesta.

```
#include <iostream>
using namespace std;
struct st {
    int A;
    char CH;
};
int main()
{
    struct st s[] = {{ 10, 'A' }, { 20, 'B' } };
    int X, Y;
    X = s[0].A + s[0].CH;
    Y = s.A + s.CH;
    cout << (X * Y);
    return 0;
}
```

2 ¿Cuál es la salida del siguiente código?. Explica tu respuesta.

```
#include <iostream>
using namespace std;
struct st1 {
    int A = 10;
    struct st2 {
        char ch = 'A';
    } S;
} SS;
int main()
{
    struct st1* PTR;
    int X = 0;
    X = PTR->A + PTR->S.ch;
```

```

    cout << X;
    return 0;
}

```

3 ¿Cuál es la salida del siguiente código?. Explica tu respuesta.

```

#include <stdio.h>
struct resultado{
    char sub[20];
    int marcas;
};

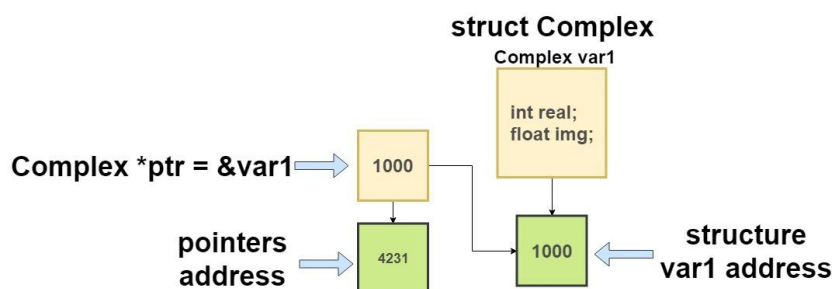
void main()
{
    struct resultado res[] = { {"Matematica",100},
                                {"Ciencia",90},
                                {"Ingles",85}
    };
    printf("%s ", res.sub);
    printf("%d", (*(res+2)).marcas);
}

```

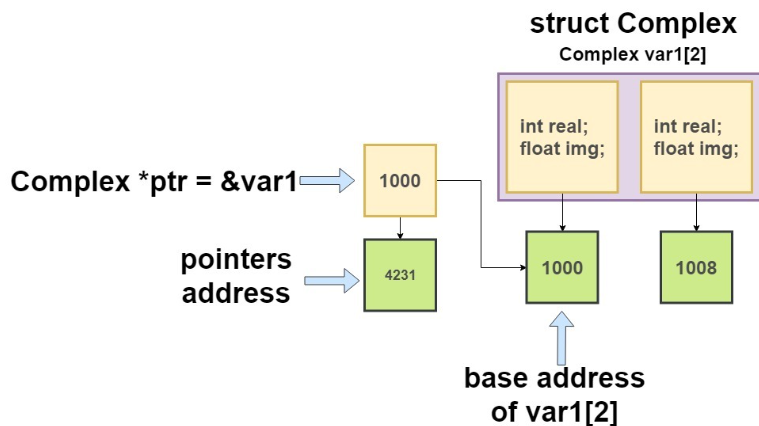
4 Escribe un programa para acceder a los miembros de la estructura utilizando los operadores de puntero a miembro, * y ->.*.

5 Implementa en lenguaje C++ un programa que maneje un arreglo de estructuras para almacenar las fechas de cumpleaños de 20 amigos. Los datos a almacenar sobre cada amigo son: nombre, día, mes y año. El programa debe indicarnos cuántos de nuestros amigos cumplen los años un día y mes determinados introducidos por teclado.

6 Escribe el código apropiado para la siguiente figura:



7 Escribe el código apropiado para la siguiente figura:



8 Implementa un programa para mantener una base de datos con información de accesorios o partes de equipos informáticos. Para esto deberás incluir lo siguiente:
El tipo estructura accesorio cuyos campos son el número de partes, nombre, cantidad (stock).

Define el arreglo inventario que tendrá un tamaño límite de 100 accesorios.

En la función **main** se presentará un menú que pedirá el ingreso de un código de operación, luego se realizará alguna de las siguientes tareas:

- Insertar: intenta añadir un nuevo accesorio, si fuera posible. Muestra un mensaje de error si el número de parte ya existe.
- Buscar: Solicita un número de accesorio y lo busca en la base de datos. Si la ubica, imprime el nombre y stock, sino imprime un mensaje de error.
- Actualizar: Solicita un número de accesorio y lo busca en la base de datos. Si el número de parte no existe muestra un mensaje de error, sino le pide al usuario ingresar la cantidad y realiza el cambio.
- Listar: Presenta todos los accesorios adquiridos. Los accesorios se mostrarán en el orden que fueron ingresados. El menú se presentará de forma interactiva hasta que el usuario presione el código 5.

MENU

- (1) Insertar
- (2) Buscar
- (3) Actualizar
- (4) Listar
- (5) Salir

Elija el código de operación.

9 Declara el dato de tipo estructura Estudiante conformada por los miembros:

- nombre, una cadena de capacidad 20 para guardar el primer nombre de un alumno del curso de Fundamentos de Programación.
- nota del tipo `int` para almacenar la nota del laboratorio calificado 5 del alumno.

Ingresa la cantidad de estructuras n a generar. Luego, declara e inicializa un arreglo de n estructuras del tipo Estudiante e imprime dicho arreglo.

Finalmente, imprime la mayor nota.

Ejemplo

Ingresa la cantidad de estudiantes a ingresar:

Nombre del estudiante: kapu

Nombre del estudiante: chalito

Nombre del estudiante: tomo

	Nombre	Nota
Nro 0:	kapu	1
Nro 1:	chalito	4
Nro 2:	tomo	9

Nota máxima: 9

10 Considera

```
struct Tnodo {
    string palabra;
    int conteo;
    Tnodo * izquierda;
    Tnodo* derecha;
};
```

Escribe una función para ingresar nuevas palabras en un árbol de Tnodo. Escribe una función para escribir un árbol de Tnodo. Escribe una función para escribir un árbol de Tnodo con las palabras en orden alfabético.

Modifica Tnodo para que almacene (solo) un puntero a una palabra arbitrariamente larga almacenada como un arreglo de caracteres usando new.

11 Estás ejecutando un sitio web y estás tratando de realizar un seguimiento de cuánto dinero gana por día con la publicidad. Declara una estructura publicitaria que realice un seguimiento de cuántos anuncios has mostrado a los lectores, en qué porcentaje de anuncios hicieron clic los usuarios y cuánto ganastes en promedio por cada anuncio en el que se hizo clic. Lee los valores para cada uno de estos campos del usuario. Pasa la estructura publicitaria a una función que imprima cada uno de los valores y luego calcula cuánto ganastes ese día.

Archivos

- 1 Escribe un programa en C++ para leer y escribir operaciones de archivos con funciones de manejo de archivos.
2. Escribe un programa en C++ para almacenar o ingresar datos en un archivo usando funciones de archivos.
3. Escribe un programa en C++ para recuperar información de un archivo usando funciones de archivo.
- 4 Escribe un programa en C++ para leer y mostrar un archivo usando funciones de archivos
5. Escribe un programa en C++ para juntar dos archivos en un tercer archivo usando funciones de archivos.
- 6 Escribe un programa en C++ para cifrar y descifrar archivos usando funciones de archivos.
- 7 Escribe un programa en C++ para contar dígitos, letras y espacios usando el manejo de archivos.
- 8 Escribe un programa en C++ para contar palabras, líneas y el tamaño total usando el archivo manejo.
- 9 Escribe un programa en C++ para leer un archivo de texto y escribirlo en otro archivo de texto utilizando funciones de archivos.
- 10 Escribe un programa en C++ para contar la ocurrencia de una palabra usando el manejo de archivos.
- 11 Escribe un programa en C++ para manipular punteros de archivos mediante el manejo de archivos.
- 12 Escribe un programa que almacene registros de empleados en un archivo. El programa debería poder leer los registros y mostrarlos en la pantalla.
- 13 Escribe un programa que lea una matriz $m \times n$ $A = (a_{ij})$ de un archivo de texto con m líneas es una lista de n entradas dobles separadas por un espacio, luego emite el sistema homogéneo $Ax = 0$ en forma algebraica explícita con m líneas de texto con el siguiente formato:

$$a_{i1} * x[1] + \dots a_{in} * x[n].$$

El nombre del archivo que contiene la matriz y el número entero n debe leerse desde la línea de comando.

Por ejemplo, para el archivo de texto:

1 -2.3 3 0.0

-9.123 2 -50 -1

debes obtener el siguiente resultado:

$$x[1] - 2.3 * x[2] + 3 * x[3] = 0$$

$$-9.123 * x[1] + 2 * x[2] - 50 * x[3] - x[4] = 0$$

14 Escribe un programa que escriba la segunda palabra (y su longitud) en un arreglo de caracteres. Las palabras están separadas por espacios, puntuación y tabulaciones.

15 Escribe un programa que lea un archivo de texto e imprima la suma de los números que se encuentran en la segunda columna (una palabra está en la segunda columna de un archivo si su primer carácter es el primer carácter que no es un espacio después del primer carácter de espacio en el línea).