

Práctico 4:

Registros en C: Structs

La teoría general para este práctico puede consultarse en los capítulos 6 y 7 de las Notas de Clase de la cátedra.

1. Escriba código C para:

- Definir un registro llamado **persona**, que sirve para guardar los siguientes datos de una persona nombre (string), edad (entero), altura (real), peso(real) y dni (long)
- Definir **Persona** como un tipo utilizando typedef
- Use **struct persona** para declarar una variable **per** y un arreglo **poblacion[10]**
- Use el tipo **Persona** para declarar una variable **per** y un arreglo **poblacion[MAX]**, donde MAX es una constante
- Lea los datos necesarios para completar una persona y los almacene en **per**
- Asigne los valores de los campos de la variable **per** al elemento 3 del arreglo **poblacion**.
- Imprima los campos del elemento 3 del arreglo **poblacion**.

2. Dados los siguientes datos, declarar los registros (structs) que los representen:

- Datos para calcular la medida de superficie de: un rectángulo, un triángulo y un trapecio.
- Un par de coordenadas cartesianas (x, y) donde x e y son dos números reales.
- Una dirección: calle y número, localidad, código postal, número de teléfono.
- Un alumno: nombre (30 caracteres); número de registro (entero de seis dígitos); tipo y número de documento de identidad; dirección; materias (para cada una: nombre, fecha de regularización y de aprobación).
- Un empleado: nombre (30 caracteres); tipo y número de documento de identidad; dirección; fecha de nacimiento; estado civil; cantidad de hijos a cargo; sexo.
- Un aula: tipo de aula deberá incluir: N° de identificación, ubicación (bloque) y tipo (Laboratorio, Conferencia, Teoría-Práctica); cantidad de mesas, cantidad de sillas, cantidad de pizarras, proyector (deberá incluir: marca, color y control remoto (si o no)).

3. Utilizando el registro aula definido en el ejercicio anterior, defina el tipo Aula. Luego realice dos funciones; una función que ingrese los datos de un aula y otra que los muestre por pantalla. Probar el uso combinado de las dos funciones en un programa.

4. Realice un programa que permita cargar en el arreglo **poblacion** definido en el ejercicio 1d los datos de **MAX** personas. Luego debe permitir al usuario ingresar un número de dni y mostrar los datos de la persona según el dni ingresado.

A PARTIR DEL EJERCICIO 5 CONSIDERE:

Debe utilizar una variable para controlar la cantidad de elementos almacenados en el arreglo, de manera tal que no sobreescriba datos ya almacenados luego de sucesivas cargas

5. Hacer un programa que permita opcionalmente: (a) ingresar los datos de aulas en un arreglo mientras el usuario lo desee; (b) mostrar por pantalla los datos de cada uno de las aulas ingresadas. Realizar esto utilizando funciones. Considere que ambas operaciones se realizan en forma aleatoria y más de una vez.

6. Considerando la estructura de empleado definida en el ejercicio 2 e) se pide:

(a) declarar el tipo empleado y la constante MAX con valor 100.

(b) declarar en el main un arreglo reg_empleados de manera tal que pueda almacenar la cantidad MAX de empleados.

(c) realizar una función que permita cargar los datos de un empleado.

(d) realice una función que permita cargar n empleados en el arreglo, donde n es ingresado por el usuario.

(e) realizar una función que dado el tipo y numero de documento muestre por pantalla los datos del mismo, en caso de no encontrarlo en el arreglo debe informarlo por pantalla.

(f) realice un menú de opciones que permita realizar las funcionalidades de los puntos (d) y (e) las veces que desee el usuario, siempre y cuando sea posible.

7. Considerando el tipo Persona trabajado en el practico. Realice un programa que a través de un menú de opciones permita

a. Cargar n personas

b. buscar una persona por dni

c. mostrar cuantas personas están cargadas

Las opciones son utilizadas en forma aleatoria y varias veces.

Ejercicios Complementarios

1. Definir un tipo InfoSalud utilizando struct para mantener el registro de salud de una persona. Los campos del registro deberán incluir el primer nombre de la persona, su apellido, sexo, la fecha de nacimiento (consistiendo de día, mes y año), el peso (en Kg) y la altura (en metros). El programa deberá contar con una función que pida los datos y los almacene en el registro de salud. Además, deberá incluir funciones que calculen y retornen la edad en años y el índice de masa corporal (IMC).

El programa deberá solicitar la información de la persona, crear el registro para la persona y mostrar la información almacenada en dicho registro; luego deberá calcular y mostrar la edad en años de la persona y su índice de masa corporal, así como en qué categoría se encuentra de acuerdo a su IMC. El IBM se calcula por la fórmula

$$\text{IMC} = \text{peso en kg} / \text{altura en metros}$$

Se considera bajo de peso cuando el IMC es menor que 18.5; normal cuando es mayor o igual a 18.5 y menor que 25; con sobrepeso cuando es mayor o igual que 25 y menor a 30, y obeso cuando el IMC es 30 o más.

2. Completar los espacios en blanco con lo que corresponda:

- a. Los elementos que forman parte de un registro son llamados _____.
- b. La palabra clave _____ introduce una declaración de registro en C.
- c. En C, la palabra clave _____ es usada para definir un sinónimo de un tipo de dato previamente definido.

3. Indicar si son verdaderas o falsas las siguientes afirmaciones:

- a. Un registro es una colección de datos relacionados lógicamente, bajo un mismo nombre.
- b. Los registros sólo pueden contener campos que sean todos de un mismo tipo.
- c. Dos structs no pueden compararse usando los operadores == y !=.
- d. Una variable de tipo struct no se puede asignar a otra variable del mismo tipo.
- e. En una declaración de una variable el nombre del struct es opcional.
- f. Los campos o miembros de distintos registros deben tener nombres únicos.
- g. La palabra clave typedef es usada para definir nuevos tipos de datos.

h. Los structs en C son siempre pasados por referencia a las funciones.

4. Dadas las siguientes definiciones y declaraciones en C:

```
typedef struct fecha {  
    int dia, mes, anio;  
} Fecha;  
  
struct emp{  
    char nombre[30];  
    unsigned int nroLegajo;  
    unsigned short int edad;  
    char estadoCivil;  
    Fecha fechaIngreso;  
} empleado;  
  
struct espo{  
    char nombre[30];  
    Fecha fechaNac;  
    unsigned short int edad;  
} esposa;
```

¿Cuáles de los siguientes grupos de sentencias son válidas?

- a. empleado.nombre[10]= esposa.nombre[10];
 empleado.edad = esposa.edad;
- b. emp.nombre[10] = espo.nombre[10];
 emp.fechaIngreso =espo.fechaNac;
- c. empleado.fechaIngreso = {1, 1, 2010};
 d. edad = 26; fechaIngreso.dia = 1; estadoCivil = 'c';
 nombre ="Pedro Rodriguez";
- e. Fecha.dia = 10; Fecha.mes = 8; Fecha.anio = 1910;
- f. struct espo esposo = {"Juan", {10,4,1978}, 35};
 esposa = {"Maria", {1,1,1980}, 33};