

Práctica 8: Tipos de datos estructurados. Registros (2 sesiones)

Programación 1. Grado en Ingeniería Multimedia

5 de diciembre de 2016

Objetivos:

- Conocer la importancia del tipo de dato registro (struct).
- Declarar, definir y saber utilizar el tipo struct.
- Diseñar e implementar tipos compuestos mediante anidación de arrays y registros.

1.

Implementar un programa en C que gestione números complejos.

- a. Debes utilizar una estructura de datos para almacenar un número complejo (TComplejo). Los números complejos tienen parte real y parte imaginaria.
- b. Mostrará un menú con 4 opciones: leer número, mostrar número, sumar dos complejos y restar dos complejos
- c. Debes implementar una función para leer un número complejo

`void LeerComplejo(TComplejo &n)`

Por tanto debes leer dos números, el primero será la parte real y el segundo la parte imaginaria de n

- d. Debes implementar una función para escribir un número complejo

`Void MostrarComplejo (TComplejo n)`

Debes procurar mostrar los números al estilo tradicional:

$2+3i$

En donde 2 es la parte real y el 3 es la parte imaginaria

- e. Usa preferiblemente una función para calcular la suma y otra para la res:

`TComplejo Suma(TComplejo n1, TComplejo n2)`

`TComplejo Resta(TComplejo n1, TComplejo n2)`

2.

La FIFA se está planteando gestionar la información de las selecciones y jugadores del próximo mundial de fútbol a celebrar en el 2018. De las selecciones le interesa disponer de la siguiente información:

- Nombre del país
- Número de convocados
- Seleccionador
- Si ha sido campeona del mundo

En el caso de los jugadores le interesa almacenar los siguientes datos:

- Nombre
- Selección a la que pertenece
- Goles marcados
- Posición en el campo (portero, defensa, mediocampo o delantero)

Teniendo en cuenta estos requerimientos, debes diseñar e implementar un programa que contenga un menú con las siguientes opciones:

1. Leer desde teclado la información de una nueva selección.
2. Leer desde teclado la información de un nuevo jugador.
3. Mostrar un listado con los datos de todas las selecciones.
4. Mostrar un listado con los datos de todos los jugadores de una selección (se pedirá el nombre de la selección por teclado).
5. Mostrar el "pichichi" del mundial (aquel jugador (o jugadores) que hayan marcado más goles). El listado mostrará: Nombre jugador, Selección a la que pertenece y número de goles marcados.

3.

Un supermercado quiere gestionar la información acerca de sus productos y sus proveedores. De cada producto le interesa conservar los siguientes datos:

- Código
- Nombre
- Proveedor
- Precio
- Stock actual
- Stock mínimo
- Stock máximo

En el caso de los proveedores le interesa almacenar la siguiente información:

- CIF
- Nombre
- Teléfono
- Dirección
- Sector (cárnico, frutas, verduras, pescadería o droguería)

Un proveedor podría suministrar diversos tipos de productos.

Teniendo en cuenta estos requerimientos, debes diseñar e implementar un programa que contenga un menú con las siguientes opciones:

1. Dar de alta un nuevo producto. Se debe comprobar que el proveedor existe (ya está dado de alta).
2. Dar de alta un nuevo proveedor.
3. Mostrar un listado con los productos de un mismo proveedor (se debe pedir el nombre del proveedor).
4. Mostrar un listado con los datos de todos los proveedores.
5. Mostrar un listado de pedidos a realizar. Los productos que se deben pedir son aquellos en el que el stock actual es menor que el stock mínimo. Se deberá mostrar: código del producto, proveedor y cantidad a pedir. La cantidad a pedir será la suficiente para que el stock del producto sea igual al stock máximo.

4.

Para poder gestionar los datos de sus empleados, una empresa nos ha pedido un programa que deberá almacenar para cada trabajador la siguiente información:

- nombre
- teléfono,
- edad,
- número de hijos,
- fecha de ingreso en la empresa,
- volumen de ventas de los últimos 4 meses (vector).

Nuestro programa tendrá un menú con una serie de opciones:

1. Introducir un nuevo empleado (tendremos que guardar cuantos empleados hemos introducido hasta ahora)
2. Listar los datos de todos los empleados cuya edad esté comprendida entre dos edades.
3. Obtener el empleado del mes, que será aquel cuyo volumen de ventas en los 4 últimos meses sea el mayor.
4. Ordenar los empleados por el volumen total de los 4 últimos meses.
5. Listar por pantalla las nóminas calculadas de todos los empleados. El listado debe tener el siguiente formato:

Nombre	Fijo	C. Trienios	IRPF	Nómina
Pepito Grillo	900 €	160 €	15%	901€
Juanito González	900€	240€	11%	1014€
...

Para calcular la nómina hay que tener en cuenta la siguiente información:

- Todos los empleados perciben un salario fijo de 900€.
- Además cobran una cantidad de 80€ por cada trienio (tres años de antigüedad en la empresa).
- A cada empleado se le aplica una retención sobre el salario que depende del nº de hijos que tiene. Se aplican los siguientes porcentajes:

Nº hijos	Retención
0	18%
1	15%
2	11%
3 ó más	10%

- El salario final se calcula como:
 $(s. \text{ fijo} + 80 * \text{trienios}) * (1 - \text{IRPF})$.
- Para simplificar los cálculos, ten en cuenta simplemente el año de la fecha de alta en la empresa y el año actual con el fin de calcular el número de trienios.

5.

Implementa un programa en lenguaje C que permita gestionar las pociones de un taller de magia. ¡Los magos están desesperados con tantas pociones! Debes ayudarlos para que no se equivoquen. Se desea almacenar para cada poción:

- Número de la poción: se rellenará automáticamente empezando en 0.
- Utilidad de la poción: "amor", "exámenes", "salud", etc.
- Ingredientes que lleva: puede contener un número variable de ingredientes, como máximo 10 (pueden estar repetidos), los cuales

se codificarán mediante un identificador entero, de la siguiente forma:

Identificador	Ingrediente
1	Bigote de gato
2	Escama de dragón
3	Diente de diablo
4	Veneno de serpiente

- Cantidad de la poción que hay en el frasco (medida en ml).
- Si se ha terminado la poción del frasco o no.

El programa tendrá un menú con las siguientes opciones:

1. **Introducir una nueva poción.** El almacén de los brujos es reducido, por lo que como máximo podrán almacenar 100 pociones. Características:

- a. El número de poción no se pedirá al usuario, se escribirá automáticamente (basándote en la posición del array), empezando en 0.
- b. Inicialmente una poción está como “no terminada”, como es lógico.

2. **Utilizar poción,** sabiendo su número, basándonos en:

- a. Ten en cuenta que el número de poción está relacionado con la posición en el array, no es necesario que la busques por todo el array.
- b. Se pedirá la cantidad que se desea utilizar (siempre que sea menor que la cantidad que queda en el frasco) y se decrementará del frasco. Si se llegase a terminar la poción, marcar una poción como “terminada”.

3. **Listar las pociones que todavía quedan** (las que no se han terminado).

4. **Finalizar la ejecución del programa.**

La ejecución del programa consistirá en ir seleccionando cualquiera de las 3 primeras opciones del menú, en cualquier orden y número de veces, hasta que se elija la opción 4, en cuyo caso el programa finalizará.