

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 8

TEMA: Estructuras. Asignación dinámica. Pasaje de Parámetros.

¡ATENCIÓN! LEA LA CONSIGNA DE TRABAJO



- ★ Trabaje de forma ordenada en cada ejercicio, de menor a mayor.
- ★ Identifique entre la información que se brinda, el tipo de dato en cada caso y la presencia de estructuras que se puedan anidar.
- ★ Use reserva dinámica de memoria para los arreglos de tipo struct.
- ★ En los módulos, trabaje el pasaje de parámetros del tipo struct por referencia.

1. Bibliomaniacos

La biblioteca "Bibliomaniacos" desea implementar un sistema para gestionar y hacer seguimiento de los préstamos de sus libros. El sistema debe registrar la información de cada libro junto con su historial de préstamos, detallando los lectores que han tomado el libro en préstamo y las fechas correspondientes.

Datos a registrar:

- Libro:
 - Título del libro
 - Género (policial, suspenso, ciencia ficción, romance, etc)
 - Número de lectores (que han tomado prestado el libro)
 - Lectores que han tomado prestado el libro
- Lector
 - Nombre
 - Edad
 - Fecha de préstamo (compuesta por día, mes y año)

Informes requeridos:

- A. Módulo para identificar el libro con mayor cantidad de lectores.
- B. Módulo para mostrar todos los libros que fueron prestados a socios menores de 21 años.
- C. Módulo para listar los libros de género "policial" que fueron prestados durante el año 2023.

Nota: Implementar el sistema utilizando arreglos dinámicos para almacenar los datos de los libros y los lectores asociados.

Punto de partida:

- a) Un programador ha iniciado el proyecto definiendo la estructura que se utilizará para continuar el desarrollo del sistema:

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

```
typedef struct {
    int dia;
    int mes;
    int anio;
} Fecha;

typedef struct {
    char nombre[MAX];
    short edad;
    Fecha fechaPrestamo;
} Lector;

typedef struct {
    char titulo[MAX];
    int numLectores;
    Lector *lectores;
} Libro;
```

- b) Cree un arreglo dinámico que contenga al menos 4 libros.
- c) Para cada libro, genere arreglos dinámicos de lectores, con una cantidad variable de lectores para cada uno.
- d) Inicialice los datos como crea conveniente.
- e) Pruebe los módulos con los informes requeridos.
- f) Antes de finalizar el programa, asegúrese de liberar la memoria asignada dinámicamente a todos los punteros utilizados.

2. Bancósmico

El banco "Bancósmico" que cuenta con 5 sucursales (Centro, Norte, Sur, Este y Oeste) desea implementar un sistema de gestión para llevar un registro de sus clientes y sus cuentas bancarias. El sistema debe permitir almacenar la información de cada cliente, las cuentas que posee y el historial de movimientos de cada cuenta.

Especificaciones del sistema:

Datos a registrar:

11/11/2024

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

- Cliente:
 - Sucursal
 - DNI
 - Nombre
 - Lista de cuentas bancarias del cliente (arreglo dinámico)
- Cuenta Bancaria:
 - Número de cuenta o CBU
 - Tipo de cuenta (por ejemplo, "Ahorro", "Corriente", "Dólares")
 - Saldo actual

Informes requeridos:

1. Módulo para mostrar el cliente con el mayor número de cuentas bancarias.
2. Módulo para listar las cuentas bancarias con saldo mayor a un valor ingresado por el usuario.
3. Módulo para listar todos los clientes de la sucursal centro que cuentan con caja de ahorro en dólares y con saldo mayor a 10.000.

Requerimientos técnicos:

- Definir estructuras anidadas para representar las relaciones entre cliente y cuenta.
- Usar punteros para manejar la asignación dinámica para el arreglo dinámico principal de clientes y de cuentas bancarias.

Nota: Asegúrese de probar cada módulo del sistema para confirmar que funciona correctamente y que toda la memoria asignada se libera adecuadamente.

3. Nerdvana

Un local de venta de videojuegos llamado "Nerdvana" desea implementar un sistema para gestionar la información de sus clientes, los juegos que venden y las compras realizadas. El sistema debe permitir registrar la información básica de cada cliente, los juegos comprados, y llevar un sistema de recompensas en el que los clientes acumulan puntos por cada compra realizada.

En cada compra, el cliente puede adquirir más de un juego. Los puntos acumulados se calculan en función del monto total de la compra, con una regla de 1 punto por cada 10,000 pesos.

El sistema debe contener la siguiente información:

- Juego:
 - Nombre del juego
 - Género (ej., "Acción", "Aventura", "Deportes")
 - Plataforma (ej., "PC", "PlayStation", "Xbox")
 - Precio

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

- Cliente (estructura principal):
 - DNI
 - Nombre
 - Edad
 - Lista de compras realizadas (arreglo dinámico de Compra)
- Compra:
 - Fecha de compra (día, mes, año)
 - Número de juegos comprados
 - Juegos comprados (arreglo dinámico de Juego)
 - Puntos obtenidos en la compra (calculados en función del precio total de los juegos comprados)

El sistema debe mostrar los siguientes reportes

1. Mostrar el cliente más joven en realizar una compra en Nerdvana
2. Mostrar el cliente que realizó más compras en el local.
3. Mostrar el cliente que ha comprado la mayor cantidad de juegos en total.
4. Mostrar la venta más costosa realizada en la tienda en el año 2024.
5. Listar todos los clientes con sus puntos totales obtenidos.
6. Mostrar cuántos juegos se vendieron en cada plataforma PC, PlayStation y Xbox.
7. Mostrar cuántos juegos de aventura para Xbox se vendieron durante los meses de invierno del 2024 (junio, julio, agosto)

4. Sembrando Aire

Sembrando aire es una empresa de reforestación y quiere contar con un programa que le ayude a gestionar su almacén de árboles, la venta y distribución a sus clientes. La información relevante es la siguiente:

- Datos de la especie:
 - Especie o nombre científico (única para cada una)
 - Categoría (caduca, perenne, conífera, frutal)
 - Nombre vulgar. Depende de la categoría, ejemplo:
 - Árboles de hoja caduca: Arce rojo, Aromo, Acacia espinosa, Castaño de Indias.
 - Árboles de hoja perenne: Encina, Alcornoque, Olivo.
 - Árboles de coníferas: Abeto, Pino, Araucaria.
 - Árboles frutales: Naranjo, Mango, Limonero, Cerezo, Manzano.
- Stock disponible en el almacén.
- Precio
- Pedidos (arreglo cuyo tamaño depende de la cantidad de clientes que hayan realizado pedidos)
 - Datos del cliente (Razón social y teléfono)
 - Datos del empleado que vende (Apellido y nombre)
 - Cantidad

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Su misión será sistematizar la información detallada y brindar informes de seguimiento y control. Para comodidad del usuario muestre el siguiente menú de opciones:

Gestión de operaciones

01. Mostrar todos los árboles/especies disponibles.
02. Mostrar árboles/especies según stock: módulo que muestre los árboles con stock inferior a un número dado como dato.
03. Mostrar todos los pedidos: indicando el monto a cobrar a cada cliente.
04. Agregar pedidos a una especie: si la especie aún no cuenta con pedidos. Controlar si hay stock disponible, caso contrario informar con un mensaje.
05. Actualizar stock de una especie: en caso de mermar controlar que el valor ingresado no sea inferior a los pedidos vigentes en caso de existir.
06. Actualizar el precio de una especie.
07. Salir