

## Enunciado:

En una pinturería tienen el stock de latas de pintura en un archivo de texto, y se desea implementar un sistema informático para poder realizar consultas.

El formato del archivo de stock (stock.txt) es una secuencia de productos agrupados por Marca, Color, y tamaño de la lata en litros (puede haber repeticiones, indicando que hay más de una lata de esa marca, color y tamaño). Una parte del archivo de stock podría ser:

---

```
ALBA
ROJO
1

COLORIN
BLANCO
0.5

POLACRIN
NEGRO
4

COLORIN
BLANCO
0.5
```

---

Se pide realizar un programa en lenguaje C que realice las siguientes acciones

- 1) Lea el archivo de texto y cargue cada uno de los productos (cada lata) en una lista enlazada utilizando punteros y memoria dinámica. Cada nodo de la lista enlazada debe crearse utilizando un tipo de dato “struct” y memoria dinámica.
  - 2) Cada producto (lata) se almacene en un tipo de dato “struct” con la marca y color en un arreglo char de tamaño máximo 20 y el tamaño de la lata en una variable float utilizando memoria dinámica.
  - 3) Presente en pantalla un menú con las siguientes opciones:
    - a. Agregar producto: El usuario ingresa una Marca, Color y Tamaño de lata y la misma se agregar a la lista enlazada  
Entrada:  
    “Ingrese la Marca:” Alba  
    “Ingrese el Color:” Blanco  
    “Ingrese el Tamaño de la lata:” 4  
Salida:  
    “Producto Agregado” (y se agrega a la lista enlazada)
    - b. Remover producto: El usuario ingresa una Marca, Color y Tamaño de lata y la misma se quita de la lista enlazada. Mostrar un cartel “Producto no encontrado” si el producto no se encontraba en la lista
-

Entrada:

“Ingrese la Marca:” Alba

“Ingrese el Color:” Blanco

“Ingrese el Tamaño de la lata:” 4

Salida:

“Producto Removido” (y se quita de la lista enlazada)

- c. Verificar Disponibilidad: El usuario ingresa un Color y el sistema muestra una lista de Marcas y Tamaños de lata de ese color o un mensaje “Color no disponible” si no hay latas de ese color (puede repetirse datos de salida si hay mas de una lata de la misma marca, color y tamaño)

Entrada:

“Ingrese el Color:” Blanco

Salida:

ALBA, 4 litros

COLORIN, 1 litro

ALBA, 0.5 litros

- d. Verificar Color y Cantidad: El usuario ingresa un color y un tamaño de lata y el sistema muestra una lista de Marcas de ese color y cantidad o un mensaje de “Stock no disponible” si no hay una lata de ese color y tamaño. Cada marca debe aparecer una sola vez, aunque haya más de una lata de una marca con ese color y tamaño

Entrada:

“Ingrese el Color:” Blanco

“Ingrese Tamaño de lata:” 1

Salida:

“Marcas Disponibles: ALBA, COLORIN”

- e. Total Color: Muestra una lista de Marcas, el total de pintura en litros, y la cantidad de latas de un color ingresado por el usuario. Cada marca debe aparecer una sola vez, aunque haya mas de una lata de esa marca y color

Entrada:

“Ingrese el Color:” Blanco

Salida:

Marcas: ALBA, COLORIN

Total Color: 12,5 litros

Cantidad de Latas: 4

- f. Total Marca: Muestra una lista de los colores disponibles y la cantidad de latas de una marca ingresada por el usuario. Cada color debe aparecer una sola vez aunque haya mas de una lata de esa marca y color

Entrada:

“Ingrese la Marca”: Alba

Salida:

Colores: AZUL, BLANCO, NEGRO

Cantidad de latas: 5

- g. Total por Marca y Color: Muestra la cantidad total de pintura en litros y la cantidad total de latas de una Marca y Color ingresado por el usuario.  
Entrada:  
    “Ingrese la Marca”: Alba  
    “Ingrese el Color”: Rojo  
Salida:  
    “Total Color: 15 litros”  
    “Cantidad de latas: 4”
  - h. Salir: Al elegir esta opción se borran todos los elementos de la lista enlazada, liberando la memoria dinámica utilizada y se finaliza la ejecución del programa
- 4) Excepto en los casos (3.a) y (3.b) Luego de presentar la información por pantalla, se debe dar la opción de generar un archivo de texto con la información presentada. Si el usuario elige generar el archivo, el sistema debe solicitar el nombre y generar el archivo con ese nombre, la extensión .TXT y el mismo contenido mostrado en pantalla (entrada y salida).
- 5) Implementar la lista enlazada con las siguientes estructuras en un archivo header (.H):

```
typedef struct lista {  
    p_nodo nodo;  
    int cantidad_nodos;  
} lista, *p_lista;  
  
typedef struct nodo {  
    p_lata elemento;  
    p_nodo siguiente;  
} nodo, *p_nodo;  
  
typedef struct lata {  
    char Marca[20];  
    char Color[20];  
    float tamanio;  
} lata, *p_lata;  
  
enum status = {status_ok, status_fail};
```

Y funciones en un archivo .C para:

- a. Crear una lista
  - b. Crear un nodo
  - c. Verificar si la lista esta vacia
  - d. Agregar un elemento a la lista (puede ser al principio o al final)
  - e. Borrar un elemento
-

### **Pautas:**

- El trabajo es individual o en grupos de 2 personas. Indicar nombre, apellido y padron en un comentario al principio del archivo .C que tiene el programa principal.
- No debe quedar memoria dinámica asignada luego de la finalización del programa.
- Se debe entregar todos los archivos .C y .H necesarios para compilar el sistema (excepto librerías estándar de C).
- El archivo de entrada tiene el texto en mayúscula, se recomienda utilizar la función toupper() con la entrada de usuario antes de hacer las comparaciones de texto con strcmp().
- Fecha de Entrega: 23.59hs del miércoles 19/06/2024 via Campus