#include <stdio.h>

#include <string.h>

#include <ctype.h>

#include <stdlib.h>

#ifdef \_WIN32

#include <conio.h> // Para getch() en Windows

#else

#include <termios.h> // Para getch() en Linux/Mac

#include <unistd.h>

#endif

#define MAX\_USUARIO 50

#define MAX\_CONTRASENA 50

#define MAX\_ITEMS 100

#define MAX\_NAME\_LENGTH 50

#define MAX\_TYPE\_LENGTH 20

#define MAX\_SIZE\_LENGTH 2

#define MAX\_COLORS 10

#define MAX\_COLOR\_LENGTH 20

// Estructura para almacenar el color y la cantidad de prendas de ese color

typedef struct {

char color[MAX\_COLOR\_LENGTH]; // Nombre del color

int quantity; // Cantidad de prendas de este color

} ColorQuantity;

// Estructura para las prendas

typedef struct {

char item\_id; // ID de la prenda (letra del alfabeto)

char name[MAX\_NAME\_LENGTH]; // Nombre de la prenda

char type[MAX\_TYPE\_LENGTH]; // Tipo de prenda

char size[MAX\_SIZE\_LENGTH]; // Talla (S, M, L)

int quantity; // Cantidad total disponible

ColorQuantity colors[MAX\_COLORS]; // Colores y cantidades

int color\_count; // Número de colores ingresados

} Item;

// Estructura para el inventario

typedef struct {

Item items[MAX\_ITEMS];

int item\_count;

} Inventory;

// Función para leer una contraseña sin mostrarla en pantalla

void leerContrasena(char \*contrasena, int max\_length) {

int i = 0;

char ch;

while (1) {

#ifdef \_WIN32

ch = \_getch(); // Lee un carácter sin mostrarlo (Windows)

#else

struct termios oldt, newt;

tcgetattr(STDIN\_FILENO, &oldt);

newt = oldt;

newt.c\_lflag &= ~(ICANON | ECHO);

tcsetattr(STDIN\_FILENO, TCSANOW, &newt);

ch = getchar();

tcsetattr(STDIN\_FILENO, TCSANOW, &oldt);

#endif

if (ch == '\r' || ch == '\n') { // Terminar al presionar Enter

contrasena[i] = '\0';

break;

} else if (ch == '\b' && i > 0) { // Manejar retroceso (Backspace)

i--;

printf("\b \b"); // Borrar el asterisco de la pantalla

} else if (i < max\_length - 1) {

contrasena[i++] = ch;

printf("\*"); // Mostrar un asterisco en lugar del carácter

}

}

printf("\n");

}

// Funciones de validación de usuario y contraseña

int validar\_contrasena(const char \*contrasena) {

int tiene\_mayuscula = 0, tiene\_minuscula = 0, tiene\_numero = 0, tiene\_especial = 0;

for (int i = 0; i < strlen(contrasena); i++) {

if (isupper(contrasena[i])) tiene\_mayuscula = 1;

else if (islower(contrasena[i])) tiene\_minuscula = 1;

else if (isdigit(contrasena[i])) tiene\_numero = 1;

else tiene\_especial = 1;

}

return tiene\_mayuscula && tiene\_minuscula && tiene\_numero && tiene\_especial;

}

int validar\_usuario(const char \*usuario) {

int tiene\_alfabetico\_o\_numero = 0;

for (int i = 0; i < strlen(usuario); i++) {

if (isalpha(usuario[i]) || isdigit(usuario[i])) {

tiene\_alfabetico\_o\_numero = 1;

break;

}

}

return tiene\_alfabetico\_o\_numero;

}

// Función para ingresar y validar los datos de la prenda

void ingresarDatos(Item \*p) {

char confirmacion[10]; // Para almacenar la respuesta de confirmación

char agregar\_colores[10]; // Para almacenar la respuesta sobre agregar colores

char ingresar\_talla[10]; // Para almacenar la respuesta sobre ingresar talla

do {

printf("Ingrese el tipo de prenda: ");

scanf("%s", p->type);

printf("Ingrese el genero (Hombre/Mujer): ");

scanf("%s", p->size);

// Preguntar si desea ingresar una talla específica

printf("¿Desea ingresar una talla específica? (si/no): ");

scanf("%s", ingresar\_talla);

if (strcasecmp(ingresar\_talla, "si") == 0) {

printf("Ingrese la talla (S, M, L, etc.): ");

scanf("%s", p->size);

} else {

strcpy(p->size, "Talla única"); // Asignar "Talla única" si no se ingresa talla

}

printf("Ingrese la cantidad total de la prenda: ");

scanf("%d", &p->quantity);

while (p->quantity <= 0) {

printf("Error: Ingrese una cantidad válida (mayor que 0): ");

scanf("%d", &p->quantity);

}

// Preguntar si desea agregar colores

printf("¿Desea agregar los colores de las prendas? (si/no): ");

scanf("%s", agregar\_colores);

if (strcasecmp(agregar\_colores, "si") == 0) {

p->color\_count = 0; // Inicializar el contador de colores

int total\_color\_quantity = 0; // Para validar la suma de las cantidades de colores

while (p->color\_count < MAX\_COLORS && total\_color\_quantity < p->quantity) {

printf("Ingrese el color %d: ", p->color\_count + 1);

scanf("%s", p->colors[p->color\_count].color);

printf("Ingrese la cantidad de prendas de color %s: ", p->colors[p->color\_count].color);

scanf("%d", &p->colors[p->color\_count].quantity);

// Validar que la cantidad no exceda el total

if (total\_color\_quantity + p->colors[p->color\_count].quantity > p->quantity) {

printf("Error: La cantidad total de prendas de colores no puede exceder %d.\n", p->quantity);

continue; // Volver a pedir este color

}

total\_color\_quantity += p->colors[p->color\_count].quantity;

p->color\_count++;

// Preguntar si desea agregar otro color

if (total\_color\_quantity < p->quantity) {

printf("¿Desea agregar otro color? (si/no): ");

scanf("%s", agregar\_colores);

if (strcasecmp(agregar\_colores, "no") == 0) {

break;

}

}

}

// Si no se ingresaron todas las prendas, asignar el resto a "Sin color"

if (total\_color\_quantity < p->quantity) {

strcpy(p->colors[p->color\_count].color, "Sin color");

p->colors[p->color\_count].quantity = p->quantity - total\_color\_quantity;

p->color\_count++;

}

} else {

p->color\_count = 0; // No se agregaron colores

}

// Mostrar los datos ingresados

printf("\nDatos ingresados:\n");

printf("Tipo de prenda: %s\n", p->type);

printf("Género: %s\n", p->size);

printf("Talla: %s\n", p->size);

printf("Cantidad total: %d\n", p->quantity);

if (p->color\_count > 0) {

printf("Colores y cantidades:\n");

for (int i = 0; i < p->color\_count; i++) {

printf("- %s: %d\n", p->colors[i].color, p->colors[i].quantity);

}

} else {

printf("No se ingresaron colores.\n");

}

// Solicitar confirmación

printf("¿Está seguro de que los datos son correctos? (si/no): ");

scanf("%s", confirmacion);

} while (strcasecmp(confirmacion, "si") != 0); // Repetir hasta que el usuario confirme

}

// Función para mostrar las prendas disponibles

void list\_items(Inventory \*inventory) {

printf("Prendas disponibles en el inventario:\n");

for (int i = 0; i < inventory->item\_count; i++) {

printf("ID: %c, Nombre: %s, Tipo: %s, Género: %s, Talla: %s, Cantidad total: %d\n",

inventory->items[i].item\_id,

inventory->items[i].name,

inventory->items[i].type,

inventory->items[i].size,

inventory->items[i].size,

inventory->items[i].quantity);

if (inventory->items[i].color\_count > 0) {

printf("Colores y cantidades:\n");

for (int j = 0; j < inventory->items[i].color\_count; j++) {

printf("- %s: %d\n", inventory->items[i].colors[j].color, inventory->items[i].colors[j].quantity);

}

} else {

printf("No hay colores registrados.\n");

}

printf("\n");

}

}

// Función para buscar una prenda por nombre o tipo

void buscar\_prenda(Inventory \*inventory) {

char criterio[MAX\_NAME\_LENGTH];

int encontrado = 0;

printf("Ingrese el nombre o tipo de la prenda a buscar: ");

scanf("%s", criterio);

for (int i = 0; i < inventory->item\_count; i++) {

if (strstr(inventory->items[i].name, criterio) != NULL || strstr(inventory->items[i].type, criterio) != NULL) {

printf("Prenda encontrada: ID: %c, Nombre: %s, Tipo: %s, Talla: %s, Cantidad: %d\n",

inventory->items[i].item\_id,

inventory->items[i].name,

inventory->items[i].type,

inventory->items[i].size,

inventory->items[i].quantity);

encontrado = 1;

}

}

if (!encontrado) {

printf("No se encontró ninguna prenda con ese nombre o tipo.\n");

}

}

// Función para eliminar una prenda del inventario por ID

void remove\_item(Inventory \*inventory) {

char item\_id;

char confirmation[10]; // Para almacenar respuestas de "si" o "no"

int found = 0;

list\_items(inventory);

printf("Para eliminar una prenda, debe ingresar un ID en letra mayúscula.\n");

printf("Ingrese el ID de la prenda a eliminar (letra del alfabeto en mayúscula): ");

scanf(" %c", &item\_id);

// Verificar si el inventario está vacío

if (inventory->item\_count == 0) {

printf("Error: El inventario está vacío.\n");

return;

}

// Buscar la prenda con el ID especificado

for (int i = 0; i < inventory->item\_count; i++) {

if (inventory->items[i].item\_id == item\_id) {

found = 1;

// Mostrar detalles de la prenda que se está eliminando

printf("Eliminando la prenda:\n");

printf("ID: %c, Nombre: %s, Tipo: %s, Talla: %s, Cantidad: %d\n",

inventory->items[i].item\_id,

inventory->items[i].name,

inventory->items[i].type,

inventory->items[i].size,

inventory->items[i].quantity);

// Confirmar eliminación

printf("¿Está seguro de que desea eliminar esta prenda? (si/no): ");

scanf("%s", confirmation);

// Comparar la respuesta sin distinguir mayúsculas y minúsculas

if (strcasecmp(confirmation, "si") == 0) {

// Desplazar los elementos restantes hacia la izquierda

for (int j = i; j < inventory->item\_count - 1; j++) {

inventory->items[j] = inventory->items[j + 1];

}

inventory->item\_count--;

printf("Prenda con ID %c eliminada con éxito.\n", item\_id);

} else {

printf("Eliminación cancelada.\n");

}

break;

}

}

// Si no se encontró la prenda

if (!found) {

printf("Error: No se encontró la prenda con ID %c.\n", item\_id);

}

}

// Función para mostrar el menú y permitir que el usuario elija acciones

void mostrarMenu() {

printf("\nSelecciona una opción:\n");

printf("1. Ingresar nueva prenda\n");

printf("2. Mostrar datos de la prenda\n");

printf("3. Buscar prenda\n");

printf("4. Eliminar prenda\n");

printf("5. Salir\n");

}

// Función principal

int main() {

char usuarioRegistrado[MAX\_USUARIO];

char contrasenaRegistrada[MAX\_CONTRASENA];

char usuarioIngresado[MAX\_USUARIO];

char contrasenaIngresada[MAX\_CONTRASENA];

int opcion;

int registroCompleto = 0;

Inventory inventory;

inventory.item\_count = 0; // Inicializa el contador de artículos

do {

printf("==== Menú ====\n");

printf("1. Registrarse\n");

printf("2. Iniciar sesión\n");

printf("3. Salir\n");

printf("Seleccione una opción: ");

scanf("%d", &opcion);

getchar(); // Consumir el salto de línea restante

if (opcion == 1) { // Registro de usuario

printf("=== Registro ===\n");

do {

printf("Ingrese un nombre de usuario: ");

fgets(usuarioRegistrado, MAX\_USUARIO, stdin);

usuarioRegistrado[strcspn(usuarioRegistrado, "\n")] = '\0'; // Eliminar el salto de línea

if (!validar\_usuario(usuarioRegistrado)) {

printf("El nombre de usuario debe contener al menos un carácter alfabético o un número.\n");

}

} while (!validar\_usuario(usuarioRegistrado));

do {

printf("Ingrese una contraseña: ");

leerContrasena(contrasenaRegistrada, MAX\_CONTRASENA);

if (!validar\_contrasena(contrasenaRegistrada)) {

printf("La contraseña debe contener al menos una letra mayúscula, una letra minúscula, un número y un carácter especial.\n");

}

} while (!validar\_contrasena(contrasenaRegistrada));

registroCompleto = 1;

printf("¡Registro exitoso! Ahora puede iniciar sesión.\n\n");

} else if (opcion == 2) { // Inicio de sesión

if (!registroCompleto) {

printf("Primero debe registrarse antes de iniciar sesión.\n\n");

continue;

}

printf("=== Iniciar sesión ===\n");

printf("Ingrese su nombre de usuario: ");

fgets(usuarioIngresado, MAX\_USUARIO, stdin);

usuarioIngresado[strcspn(usuarioIngresado, "\n")] = '\0'; // Eliminar el salto de línea

printf("Ingrese su contraseña: ");

leerContrasena(contrasenaIngresada, MAX\_CONTRASENA);

// Validar usuario y contraseña

if (strcmp(usuarioRegistrado, usuarioIngresado) == 0 &&

strcmp(contrasenaRegistrada, contrasenaIngresada) == 0) {

printf("¡Inicio de sesión exitoso! Bienvenido, %s.\n\n", usuarioRegistrado);

do {

mostrarMenu();

printf("Ingrese su opción: ");

scanf("%d", &opcion);

if (opcion == 1) {

Item nuevaPrenda;

printf("Ingresar nueva prenda:\n");

ingresarDatos(&nuevaPrenda);

nuevaPrenda.item\_id = 'A' + inventory.item\_count; // Asignar un ID único a la prenda

inventory.items[inventory.item\_count] = nuevaPrenda;

inventory.item\_count++;

printf("Prenda agregada con éxito.\n");

} else if (opcion == 2) {

list\_items(&inventory);

} else if (opcion == 3) {

buscar\_prenda(&inventory);

} else if (opcion == 4) {

remove\_item(&inventory);

} else if (opcion == 5) {

printf("Saliendo...\n");

break;

} else {

printf("Opción no válida. Intente de nuevo.\n");

}

} while (opcion != 5);

} else {

printf("Nombre de usuario o contraseña incorrectos.\n\n");

}

} else if (opcion == 3) {

printf("Saliendo del programa.\n");

} else {

printf("Opción no válida. Intente de nuevo.\n\n");

}

} while (opcion != 3);

return 0;

}