#include <stdio.h>

#include <string.h>

#include <stdbool.h>

#define MAX\_ITEMS 100 // Maximum commodity quantity

#define MAX\_NAME\_LENGTH 100 // Maximum name length

#define MAX\_CATEGORY\_LENGTH 50 // Class maximum length

// Set yes or no

typedef enum {

IN\_STOCK,

OUT\_OF\_STOCK

} AvailabilityStatus;

//Construction variable

typedef struct {

char name[MAX\_NAME\_LENGTH];

char category[MAX\_CATEGORY\_LENGTH];

int quantity;

float price;

AvailabilityStatus status;

} GroceryItem;

GroceryItem inventory[MAX\_ITEMS];

int itemCount = 0; // Change in commodity quantity

// Additional goods

void addGroceryItem() {

if (itemCount >= MAX\_ITEMS) {

printf("Stock is full, cannot add more items.\n");

return;

}

// Create a new item

GroceryItem newItem;

printf("Please enter the product name: ");

fgets(newItem.name, MAX\_NAME\_LENGTH, stdin);

newItem.name[strcspn(newItem.name, "\n")] = '\0';

printf("Please enter the product category: ");

fgets(newItem.category, MAX\_CATEGORY\_LENGTH, stdin);

newItem.category[strcspn(newItem.category, "\n")] = '\0';

printf("Please enter the quantity of goods: ");

scanf("%d", &newItem.quantity);

printf("Please enter the price of the item: ");

scanf("%f", &newItem.price);

getchar();

newItem.status = newItem.quantity > 0 ? IN\_STOCK : OUT\_OF\_STOCK;

// Add new items to inventory

inventory[itemCount++] = newItem;

printf("Product added successfully!\n");

}

void listAllItems() {

if (itemCount == 0) {

printf("There are no goods in stock.\n");

return;

}

printf("\nList of all products:\n");

for (int i = 0; i < itemCount; i++) {

printf("Trade name: %s\n", inventory[i].name);

printf("Commodity category: %s\n", inventory[i].category);

printf("Commodity quantity: %d\n", inventory[i].quantity);

printf("Commodity price: %.2f\n", inventory[i].price);

printf("Stock status: %s\n", inventory[i].status == IN\_STOCK ? "In stock" : "Out of stock");

printf("----------------------------------\n");

}

}

void updateQuantity() {

char itemName[MAX\_NAME\_LENGTH];

printf("Please enter the name of the product for which you want to update the quantity: ");

fgets(itemName, MAX\_NAME\_LENGTH, stdin);

itemName[strcspn(itemName, "\n")] = '\0';

// Search for goods

for (int i = 0; i < itemCount; i++) {

if (strcmp(inventory[i].name, itemName) == 0) {

int newQuantity;

printf("Please enter the new quantity of item %s: ", itemName);

scanf("%d", &newQuantity);

getchar();

// Update item quantity and stock status

inventory[i].quantity = newQuantity;

inventory[i].status = newQuantity > 0 ? IN\_STOCK : OUT\_OF\_STOCK;

printf("Item quantity updated successfully！\n");

return;

}

}

printf("This item was not found.\n");

}

void removeGroceryItem() {

char itemName[MAX\_NAME\_LENGTH];

printf("Please enter the name of the product you want to delete: ");

fgets(itemName, MAX\_NAME\_LENGTH, stdin);

itemName[strcspn(itemName, "\n")] = '\0';

for (int i = 0; i < itemCount; i++) {

if (strcmp(inventory[i].name, itemName) == 0) {

// Delete the item and adjust the inventory array

for (int j = i; j < itemCount - 1; j++) {

inventory[j] = inventory[j + 1];

}

itemCount--;

printf("Item '%s' deleted successfully！\n", itemName);

return;

}

}

printf("This item was not found.\n");

}

// Display menu

int main() {

int choice;

do {

printf("\nGrocery Inventory Management System\n");

printf("1. Add Grocery Item\n");

printf("2. List All Grocery Items\n");

printf("3. Update Quantity\n");

printf("4. Remove Grocery Item\n");

printf("5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

getchar(); // Consume the newline character

switch (choice) {

case 1:

addGroceryItem();

break;

case 2:

listAllItems();

break;

case 3:

updateQuantity();

break;

case 4:

removeGroceryItem();

break;

case 5:

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

break;

default:

printf("Invalid choice! Please try again.\n");

}

} while (choice != 5); // Continue the menu loop until the user chooses to exit

return 0;

}