*#include* <stdio.h>

*#include* <string.h>

*#define* MAX\_ITEMS 50

struct InventoryItem {

    char name[50];

    float price;

    int quantity;

};

void manageInventory(struct InventoryItem *inventory*[], int \**numItems*) {

    int choice;

    printf("\n1. Add new item\n");

    printf("2. Update item quantity\n");

    printf("3. Display inventory\n");

    printf("4. Exit\n");

    printf("Enter choice => ");

    scanf("%d", &choice);

*switch* (choice) {

*case* 1:

*// Add new item to the inventory*

*if* (\**numItems* < MAX\_ITEMS) {

                printf("Enter the name of the new item => ");

                scanf("%s", *inventory*[\**numItems*].name);

                printf("Enter the price of %s => RM", *inventory*[\**numItems*].name);

                scanf("%f", &*inventory*[\**numItems*].price);

                printf("Enter the quantity of %s => ", *inventory*[\**numItems*].name);

                scanf("%d", &*inventory*[\**numItems*].quantity);

                (\**numItems*)++;

                printf("Item added to the inventory.\n");

            } *else* {

                printf("Inventory is full. Cannot add more items.\n");

            }

*break*;

*case* 2:

*// Update item quantity*

*if* (\**numItems* > 0) {

                char itemName[50];

                int updatedQuantity;

                printf("Enter the name of the item => ");

                scanf("%s", itemName);

*for* (int i = 0; i < \**numItems*; i++) {

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

                        printf("Enter the new quantity for %s => ", *inventory*[i].name);

                        scanf("%d", &updatedQuantity);

*inventory*[i].quantity = updatedQuantity;

                        printf("Quantity updated for %s.\n", *inventory*[i].name);

*break*;

                    }

*if* (i == \**numItems* - 1) {

                        printf("Item not found in the inventory.\n");

                    }

                }

            } *else* {

                printf("Inventory is empty. Cannot update quantity.\n");

            }

*break*;

*case* 3:

*// Display inventory*

*if* (\**numItems* > 0) {

                printf("\nCurrent Inventory:\n");

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

                printf("%-20s%-15s%-10s\n", "Item Name", "Price(RM)", "Quantity");

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

*for* (int i = 0; i < \**numItems*; i++) {

                    printf("%-20s%-15.2f%-10d\n", *inventory*[i].name, *inventory*[i].price, *inventory*[i].quantity);

                }

            } *else* {

                printf("Inventory is empty.\n");

            }

*break*;

*case* 4:

*// Exit function*

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

*break*;

*default*:

            printf("Invalid choice. Please enter a valid option.\n");

    }

}

int main() {

    struct InventoryItem inventory[MAX\_ITEMS];

    int numItems = 0;

    int menuChoice;

*do* {

*// Display main menu*

        printf("\nMain Menu:\n");

        printf("1. Manage Inventory\n");

        printf("2. Exit\n");

        printf("Enter choice => ");

        scanf("%d", &menuChoice);

*switch* (menuChoice) {

*case* 1:

*// Manage inventory*

                manageInventory(inventory, &*numItems*);

*break*;

*case* 2:

*// Exit program*

                printf("Exiting the program. Thank you :>\n");

*break*;

*default*:

                printf("Invalid choice. Please enter a valid option.\n");

        }

    } *while* (menuChoice != 2);

*return* 0;

}