```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#define MAX_ITEMS 50

// Structure for menu items
struct MenuItem {
    char name[50];
    float price;
};

// Structure for ordered items
struct OrderItem {
    char name[50];
    int quantity;
    float price;
};

// Global menu with 8 Starbucks-style items in USD
struct MenuItem menu[MAX_ITEMS] = { // why is this written this way
    {"Espresso", 3.50},
    {"Cappuccino", 4.00},
    {"Latte", 4.50},
    {"Americano", 3.00},
    {"Mocha", 5.00},
    {"Macchiato", 4.75},
    {"Frappuccino", 5.50},
    {"Hot Chocolate", 3.75}
};
int menuCount = 8; // updated count

// Current order storage
struct OrderItem currentOrder[MAX_ITEMS];
int currentOrderCount = 0;
float currentOrderTotal = 0;

// Customer info
char customerName[50];
char customerPhone[20];
int customerCounter = 1; // Customer numbering

// Additional order info
```

```
char paymentMethod[20];
int satisfaction; // 1-5
char orderDate[20]; // dd-mm-yyyy

// Display cafe menu
void displayMenu() {
    printf("\n------ Cafe Menu ------\n");
    for (int i = 0; i < menuCount; i++) {
        printf("%d. %-20s $%.2f\n", i + 1, menu[i].name, menu[i].price);
    }
    printf("-----------------------\n");
}

// Take new order
void takeOrder() {
    currentOrderTotal = 0;
    currentOrderCount = 0;

    printf("\nCustomer #%d\n", customerCounter);
    printf("Enter customer name: ");
    scanf(" %[^\n]", customerName);
    printf("Enter phone number: ");
    scanf(" %[^\n]", customerPhone);

    // Get order date
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);
    sprintf(orderDate, "%02d-%02d-%04d", tm.tm_mday, tm.tm_mon + 1, tm.tm_year + 1900);

    displayMenu();

    char more;
    int index = 0;

    do {
        int choice, qty;

        printf("Enter item number: ");
        scanf("%d", &choice);

        if (choice < 1 || choice > menuCount) {
            printf("Invalid choice. Try again.\n");
            continue;
        }
```

```
        printf("Enter quantity: ");
        scanf("%d", &qty);

        strcpy(currentOrder[index].name, menu[choice - 1].name);
        currentOrder[index].quantity = qty;
        currentOrder[index].price = menu[choice - 1].price;
        index++;
        currentOrderCount = index;

        while(1) {
            printf("Add more items? (y/n): ");
            scanf(" %c", &more);
            if(more == 'y' || more == 'Y' || more == 'n' || more == 'N') break;
            printf("Invalid option. Please enter 'y' or 'n'.\n");
        }

    } while (more == 'y' || more == 'Y');
}

// Cancel current order
void cancelOrder() {
    int confirm;
    if(currentOrderCount == 0){
        printf("\nNo order to cancel.\n");
        return;
    }
    printf("\nDo you want to cancel the current order? (1: Yes / 0: No): ");
    scanf("%d", &confirm);
    if(confirm == 1){
        for(int i=0; i<MAX_ITEMS; i++){
            currentOrder[i].quantity = 0;
        }
        currentOrderCount = 0;
        currentOrderTotal = 0;
        printf("Order canceled successfully.\n");
    } else {
        printf("Order not canceled.\n");
    }
}

// Remove/reduce item quantity from current order
void modifyOrder() {
    if (currentOrderCount == 0) {
```

```
        printf("\nNo current order to modify!\n");
        return;
    }

    printf("\n--- Current Order ---\n");
    for (int i = 0; i < currentOrderCount; i++) {
        printf("%d. %-12s x %-3d = $%.2f\n", i + 1, currentOrder[i].name, currentOrder[i].quantity,
currentOrder[i].price * currentOrder[i].quantity);
    }
    printf("---------------------\n");

    int choice, qty;
    printf("Enter item number to modify: ");
    scanf("%d", &choice);
    if(choice < 1 || choice > currentOrderCount){
        printf("Invalid choice.\n");
        return;
    }

    printf("Enter quantity to remove (current quantity: %d): ", currentOrder[choice - 1].quantity);
    scanf("%d", &qty);

    if(qty >= currentOrder[choice - 1].quantity){
        for(int i = choice - 1; i < currentOrderCount - 1; i++){
            currentOrder[i] = currentOrder[i + 1];
        }
        currentOrderCount--;
        printf("Item removed.\n");
    } else {
        currentOrder[choice - 1].quantity -= qty;
        printf("Quantity reduced. New quantity: %d\n", currentOrder[choice - 1].quantity);
    }
}

// Complete current order and save to file
void completeOrder() {
    if(currentOrderCount == 0){
        printf("\nNo order to complete!\n");
        return;
    }

    // Ask payment method after order is complete
    printf("Enter payment method (Cash/Card/Other): ");
    scanf(" %[^\n]", paymentMethod);
```

```
    // Ask customer satisfaction after order is complete
    printf("Rate our services (1-5): ");
    scanf("%d", &satisfaction);
    if (satisfaction < 1) satisfaction = 1;
    if (satisfaction > 5) satisfaction = 5;

    currentOrderTotal = 0;
    printf("\n------ Receipt ------\n");
    printf("Date: %s\n", orderDate);
    printf("Customer: %s\n", customerName);
    printf("Phone: %s\n", customerPhone);
    printf("Payment Method: %s\n", paymentMethod);
    printf("Customer Satisfaction: %d/5\n", satisfaction);

    for(int i = 0; i < currentOrderCount; i++){
        float itemTotal = currentOrder[i].price * currentOrder[i].quantity;
        printf("%-12s x %-3d = $%.2f\n", currentOrder[i].name, currentOrder[i].quantity, itemTotal);
        currentOrderTotal += itemTotal;
        currentOrder[i].quantity = 0; // reset for next order
    }
    printf("---------------------\n");
    printf("Total: $%.2f\n", currentOrderTotal);
    printf("Thank you, %s, for your order!\n\n", customerName);
    printf("Order completed for Customer #%d.\n", customerCounter);
    printf("----------------------------\nNext customer, please.\n");

    // Save to file
    FILE *fp = fopen("cafe_order.txt", "a");
    if(fp != NULL){
        fprintf(fp, "------ Receipt ------\n");
        fprintf(fp, "Date: %s\n", orderDate);
        fprintf(fp, "Customer: %s\n", customerName);
        fprintf(fp, "Phone: %s\n", customerPhone);
        fprintf(fp, "Payment Method: %s\n", paymentMethod);
        fprintf(fp, "Customer Satisfaction: %d/5\n", satisfaction);
        for(int i = 0; i < currentOrderCount; i++){
            fprintf(fp, "%-12s x %-3d = $%.2f\n", currentOrder[i].name, currentOrder[i].quantity,
currentOrder[i].price * currentOrder[i].quantity);
        }
        fprintf(fp, "---------------------\nTotal: $%.2f\n", currentOrderTotal);
        fprintf(fp, "Thank you, %s, for your order!\n\n", customerName);
        fprintf(fp, "----------------------------\n\n");
        fclose(fp);
```

```
    }

    currentOrderTotal = 0;
    currentOrderCount = 0;
    customerCounter++;
}

// Add new menu item
void addMenuItem() {
    if(menuCount >= MAX_ITEMS){
        printf("Menu is full! Cannot add more items.\n");
        return;
    }

    char name[50];
    float price;

    printf("Enter new item name: ");
    scanf(" %[^\n]", name);
    printf("Enter item price: ");
    scanf("%f", &price);

    strcpy(menu[menuCount].name, name);
    menu[menuCount].price = price;
    menuCount++;

    printf("Item '%s' added to menu with price $%.2f\n", name, price);
}

// Remove menu item
void removeMenuItem() {
    if(menuCount == 0){
        printf("Menu is empty! Nothing to remove.\n");
        return;
    }

    displayMenu();
    int choice;
    printf("Enter item number to remove: ");
    scanf("%d", &choice);

    if(choice < 1 || choice > menuCount){
        printf("Invalid choice.\n");
        return;
```

```
    }

    for(int i = choice - 1; i < menuCount - 1; i++){
        menu[i] = menu[i + 1];
    }
    menuCount--;

    printf("Item removed from menu.\n");
}

int main() {
    int choice;

    printf("====================================\n");
    printf("       /\\_/\\\n");
    printf("      ( ^.^ )    LUCKY CAT CAFE     \n");
    printf("       > ^ <\n");
    printf("====================================\n");

    while(1){
        printf("\nMenu Options:\n");
        printf("1. Take Order\n");
        printf("2. Remove/Reduce Item from Order\n");
        printf("3. Cancel Current Order\n");
        printf("4. Remove Menu Item\n");
        printf("5. Add Menu Item\n");
        printf("6. Complete Current Order\n");
        printf("7. Exit\n");
        printf("Choose an option: ");
        scanf("%d", &choice);

        switch(choice){
            case 1:
                takeOrder();
                break;
            case 2:
                modifyOrder();
                break;
            case 3:
                cancelOrder();
                break;
            case 4:
                removeMenuItem();
                break;
```

```
            case 5:
                addMenuItem();
                break;
            case 6:
                completeOrder();
                break;
            case 7:
                printf("\nThat's a wrap for today at Lucky Cat Cafe! Thank you for your service. Have
a great evening!\n");
                exit(0);
            default:
                printf("Invalid option. Try again.\n");
        }
    }

    return 0;
}
```