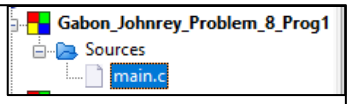
PROBLEM 8 (PROGRAM 1)




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

#define MAX_PARTS 50

typedef enum {
    CPU,
    GPU,
    RAM,
    STORAGE,
    MOTHERBOARD
} Category;
struct Part {
    char name[50];
    Category category;
    float price;
    int quantity;
};
struct Shop {
    struct Part inventory[MAX_PARTS];
    int partCount;
};
void addPart(struct Shop *shop) {
    if(shop->partCount >= MAX_PARTS) {
        printf("Inventory full!\n");
        return;
    }

    struct Part newPart;
    printf("Enter part name: ");
    scanf(" %[^\n]s", newPart.name);

    printf("Select category (0-CPU, 1-GPU, 2-RAM, 3-Storage, 4-Motherboard): ");
    scanf("%d", (int*)&newPart.category);

    printf("Enter price: $");
    scanf("%f", &newPart.price);

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

```c
    shop->inventory[shop->partCount] = newPart;
    shop->partCount++;
    printf("Part added successfully!\n");
}
void displayInventory(struct Shop *shop) {
    char *categories[] = {"CPU", "GPU", "RAM", "Storage", "Motherboard"};

    printf("\nCurrent Inventory:\n");
    printf("%Name\t\t\tCategory\tPrice\t\tQuantity\n");
    printf("-------------------------------------------------------\n");

    for(int i = 0; i < shop->partCount; i++) {
        printf("%-20s\t%-10s\t$%-10.2f\t%d\n",
            shop->inventory[i].name,
            categories[shop->inventory[i].category],
            shop->inventory[i].price,
            shop->inventory[i].quantity);
    }
}
void updateStock(struct Shop *shop) {
    char searchName[50];
    printf("Enter part name: ");
    scanf(" %[^\n]s", searchName);

    for(int i = 0; i < shop->partCount; i++) {
        if(strcmp(shop->inventory[i].name, searchName) == 0) {
            printf("Enter new quantity: ");
            scanf("%d", &shop->inventory[i].quantity);
            printf("Stock updated successfully!\n");
            return;
        }
    }
    printf("Part not found!\n");
}
int main() {
    struct Shop shop = {.partCount = 0};
    int choice;

    do {
```

```c
        printf("\n=====Computer Parts Shop Program=====\n");
        printf("1. Add New Part\n");
        printf("2. Display Inventory\n");
        printf("3. Update Stock\n");
        printf("0. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);

        switch(choice) {
            case 1: addPart(&shop); break;
            case 2: displayInventory(&shop); break;
            case 3: updateStock(&shop); break;
            case 0: printf("Goodbye!\n"); break;
            default: printf("Invalid choice!\n");
        }
    } while(choice != 0);
    return 0;
}
```

```
=====Computer Parts Shop Program=====
1. Add New Part
2. Display Inventory
3. Update Stock
0. Exit
Enter choice: 1

Enter part name: AMD Ryzen 7 5800X
Select category (0-CPU, 1-GPU, 2-RAM, 3-Storage, 4-Motherboard): 0
Enter price: $299.99
Enter quantity: 15
Part added successfully!
```

```
=====Computer Parts Shop Program=====
1. Add New Part
2. Display Inventory
3. Update Stock
0. Exit
Enter choice: 1

Enter part name: NVIDIA RTX 3080
Select category (0-CPU, 1-GPU, 2-RAM, 3-Storage, 4-Motherboard): 1
Enter price: $699.99
Enter quantity: 8
Part added successfully!
```

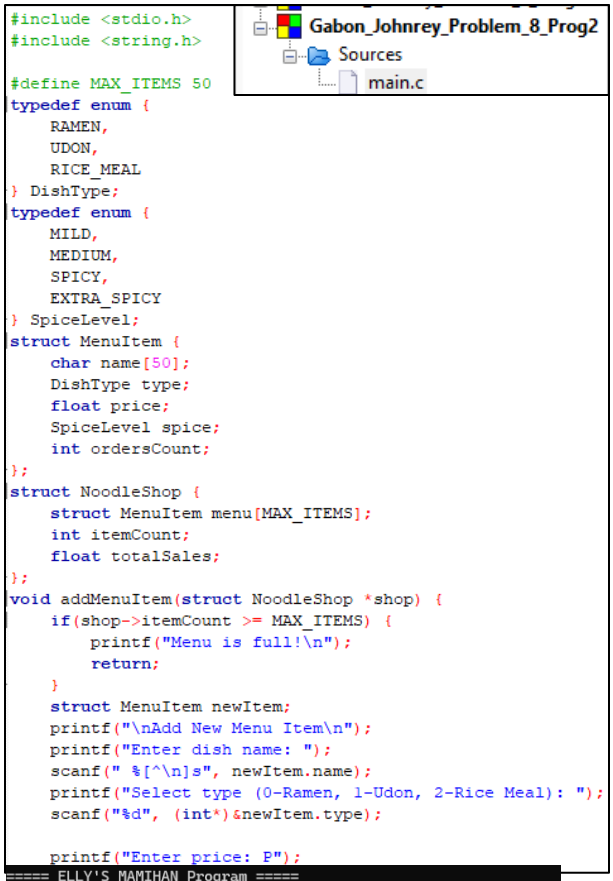
```
=====Computer Parts Shop Program=====
1. Add New Part
2. Display Inventory
3. Update Stock
0. Exit
Enter choice: 2


Current Inventory:
Name                    Category        Price           Quantity
-------------------------------------------------------
AMD Ryzen 7 5800X       CPU             $299.99         15
NVIDIA RTX 3080         GPU             $699.99         8
```

```
=====Computer Parts Shop Program=====
1. Add New Part
2. Display Inventory
3. Update Stock
0. Exit
Enter choice: 3

Enter part name: NVIDIA RTX 3080
Enter new quantity: 5
Stock updated successfully!
```

```
=====Computer Parts Shop Program=====
1. Add New Part
2. Display Inventory
3. Update Stock
0. Exit
Enter choice: 2


Current Inventory:
Name                    Category        Price           Quantity
-------------------------------------------------------
AMD Ryzen 7 5800X       CPU             $299.99         15
NVIDIA RTX 3080         GPU             $699.99         5
```

```
=====Computer Parts Shop Program=====
1. Add New Part
2. Display Inventory
3. Update Stock
0. Exit
Enter choice: 0

Goodbye!

Process returned 0 (0x0)   execution time : 97.276 s
Press any key to continue.
```

PROBLEM 8 (PROGRAM 2)




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

#define MAX_ITEMS 50
typedef enum {
    RAMEN,
    UDON,
    RICE_MEAL
} DishType;
typedef enum {
    MILD,
    MEDIUM,
    SPICY,
    EXTRA_SPICY
} SpiceLevel;
struct MenuItem {
    char name[50];
    DishType type;
    float price;
    SpiceLevel spice;
    int ordersCount;
};
struct NoodleShop {
    struct MenuItem menu[MAX_ITEMS];
    int itemCount;
    float totalSales;
};
void addMenuItem(struct NoodleShop *shop) {
    if(shop->itemCount >= MAX_ITEMS) {
        printf("Menu is full!\n");
        return;
    }
    struct MenuItem newItem;
    printf("\nAdd New Menu Item\n");
    printf("Enter dish name: ");
    scanf(" %[^\n]s", newItem.name);
    printf("Select type (0-Ramen, 1-Udon, 2-Rice Meal): ");
    scanf("%d", (int*)&newItem.type);

    printf("Enter price: P");
    scanf("%f", &newItem.price);

    printf("Select spice level (0-Mild, 1-Medium, 2-Spicy, 3-Extra Spicy): ");
    scanf("%d", (int*)&newItem.spice);

    newItem.ordersCount = 0;

    shop->menu[shop->itemCount] = newItem;
    shop->itemCount++;
    printf("Menu item added successfully!\n");
}

void displayMenu(struct NoodleShop *shop) {
    char *types[] = {"Ramen", "Udon", "Rice Meal"};
    char *spiceLevels[] = {"Mild", "Medium", "Spicy", "Extra Spicy"};

    printf("\n===== ELLY'S MAMIHAN MENU =====\n");
    printf("Name\t\t\tType\t\tSpice Level\tPrice\n");
    printf("-----------------------------------------------------\n");
    for(int i = 0; i < shop->itemCount; i++) {
        printf("%-20s\t%-10s\t%-10s\tP%.2f\n",
            shop->menu[i].name,
            types[shop->menu[i].type],
            spiceLevels[shop->menu[i].spice],
            shop->menu[i].price);
    }
}
void placeOrder(struct NoodleShop *shop) {
    char dishName[50];
    printf("Enter dish name: ");
    scanf(" %[^\n]s", dishName);

    for(int i = 0; i < shop->itemCount; i++) {
        if(strcmp(shop->menu[i].name, dishName) == 0) {
            shop->menu[i].ordersCount++;
            shop->totalSales += shop->menu[i].price;
            printf("Order placed successfully!\n");
            printf("Total sales: P%.2f\n", shop->totalSales);
            return;
        }
    }
    printf("Dish not found!\n");
}
int main() {
    struct NoodleShop shop = {.itemCount = 0, .totalSales = 0};
    int choice;
    do {
        printf("\nELLY'S MAMIHAN Management System\n");
        printf("1. Add Menu Item\n");
        printf("2. Display Menu\n");
        printf("3. Place Order\n");
        printf("0. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);
        switch(choice) {
            case 1: addMenuItem(&shop); break;
            case 2: displayMenu(&shop); break;
            case 3: placeOrder(&shop); break;
            case 0: printf("Goodbye!\n"); break;
            default: printf("Invalid choice!\n");
        }
    } while(choice != 0);
    return 0;
}
```

```
===== ELLY'S MAMIHAN Program =====
1. Add Menu Item
2. Display Menu
3. Place Order
0. Exit
Enter choice: 1


Add New Menu Item
Enter dish name: Mami with Ramen
Select type (0-Ramen, 1-Udon, 2-Rice Meal): 0
Enter price: P150.00
Select spice level (0-Mild, 1-Medium, 2-Spicy, 3-Extra Spicy): 1
Menu item added successfully!
```

```
===== ELLY'S MAMIHAN Program =====
1. Add Menu Item
2. Display Menu
3. Place Order
0. Exit
Enter choice: 1


Add New Menu Item
Enter dish name: Overload Mami
Select type (0-Ramen, 1-Udon, 2-Rice Meal): 2
Enter price: P280.00
Select spice level (0-Mild, 1-Medium, 2-Spicy, 3-Extra Spicy): 2
Menu item added successfully!
```

```
===== ELLY'S MAMIHAN Program =====
1. Add Menu Item
2. Display Menu
3. Place Order
0. Exit
Enter choice: 2


===== ELLY'S MAMIHAN MENU =====
Name                    Type            Spice Level     Price
-----------------------------------------------------
Mami with Ramen         Ramen           Medium          P150.00
Overload Mami           Rice Meal       Spicy           P280.00
```

```
===== ELLY'S MAMIHAN Program =====
1. Add Menu Item
2. Display Menu
3. Place Order
0. Exit
Enter choice: 3

Enter dish name: Overload Mami
Order placed successfully!
Total sales: P280.00
```

```
===== ELLY'S MAMIHAN Program =====
1. Add Menu Item
2. Display Menu
3. Place Order
0. Exit
Enter choice: 0

Goodbye!

Process returned 0 (0x0)   execution
Press any key to continue.
```