PROBLEM 7 (PROGRAM 1)

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

#define MAX_PRODUCTS 5

struct Product {
    char name[30];
    float price;
    int quantity;
};
struct Store {
    struct Product items[MAX_PRODUCTS];
    int count;
};
// initialized features methods
void addProduct(struct Store *store);
void showProducts(struct Store *store);
void updateStock(struct Store *store);
float calculateTotalValue(struct Store *store);

int main() {
    struct Store store = {.count = 0};
    int choice;

    do {
        printf("\n1. Add Product\n");
        printf("2. Show Products\n");
        printf("3. Update Stock\n");
        printf("4. Calculate Total Value\n");
        printf("0. Exit\n");
        printf("Choice: ");
        scanf("%d", &choice);

        switch(choice) {
            case 1: addProduct(&store); break;
            case 2: showProducts(&store); break;
            case 3: updateStock(&store); break;
            case 4: printf("Total inventory value: $%.2f\n",
                           calculateTotalValue(&store)); break;
            case 0: printf("Goodbye!\n"); break;
            default: printf("Invalid choice!\n");
        }
    } while(choice != 0);

    return 0;
}
void addProduct(struct Store *store) {
    if(store->count >= MAX_PRODUCTS) {
        printf("Store is full!\n");
        return;
    }

    printf("Enter product name: ");
    scanf(" %[^\n]s", store->items[store->count].name);
    printf("Enter price: ");
    scanf("%f", &store->items[store->count].price);
    printf("Enter quantity: ");
    scanf("%d", &store->items[store->count].quantity);

    store->count++;
}
void showProducts(struct Store *store) {
    printf("\nProduct List:\n");
    printf("Name\t\tPrice\tQuantity\n");
    for(int i = 0; i < store->count; i++) {
        printf("%-15s$%.2f\t%d\n",
            store->items[i].name,
            store->items[i].price,
            store->items[i].quantity);
    }
}
void updateStock(struct Store *store) {
    char searchName[30];
    int newQuantity;

    printf("Enter product name: ");
    scanf(" %[^\n]s", searchName);

    for(int i = 0; i < store->count; i++) {
        if(strcmp(store->items[i].name, searchName) == 0) {
            printf("Enter new quantity: ");
            scanf("%d", &newQuantity);
            store->items[i].quantity = newQuantity;
            printf("Stock updated successfully!\n");
            return;
        }
    }
    printf("Product not found!\n");
}
float calculateTotalValue(struct Store *store) {
    float total = 0;
    for(int i = 0; i < store->count; i++) {
        total += store->items[i].price * store->items[i].quantity;
    }
    return total;
}
```

Gabon_Johnrey_Problem7_Prog1
Sources
main.c

```
=====Store Management Program=====
1. Add Product
2. Show Products
3. Update Stock
4. Calculate Total Value
0. Exit
Choice: 1

Enter product name: Milk
Enter price: 150.99
Enter quantity: 50

1. Add Product
2. Show Products
3. Update Stock
4. Calculate Total Value
0. Exit
Choice: 1

Enter product name: Bread
Enter price: 50.00
Enter quantity: 30

1. Add Product
2. Show Products
3. Update Stock
4. Calculate Total Value
0. Exit
```

```
1. Add Product
2. Show Products
3. Update Stock
4. Calculate Total Value
0. Exit
Choice: 1

Enter product name: Eggs
Enter price: 15.99
Enter quantity: 40

1. Add Product
2. Show Products
3. Update Stock
4. Calculate Total Value
0. Exit
Choice: 1

Enter product name: Bananas
Enter price: 20.50
Enter quantity: 100
```

```
1. Add Product
2. Show Products
3. Update Stock
4. Calculate Total Value
0. Exit
Choice: 4

Total inventory value: $10984.15

1. Add Product
2. Show Products
3. Update Stock
4. Calculate Total Value
0. Exit
Choice: 0

Goodbye!
```

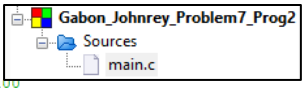
```
1. Add Product
2. Show Products
3. Update Stock
4. Calculate Total Value
0. Exit
Choice: 3

Enter product name: Milk
Enter new quantity: 45
Stock updated successfully!
```

```
1. Add Product
2. Show Products
3. Update Stock
4. Calculate Total Value
0. Exit
Choice: 2


Product List:
Name            Price   Quantity
Milk            P150.99  50
Bread           P50.00   30
Eggs            P15.99   40
Bananas         P20.50   100
```

PROBLEM 7 (PROGRAM 2)




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

#define MAX_ORDERS 100
typedef enum {
    PENDING,
    IN_PROCESS,
    READY,
    DELIVERED
} Status;
typedef enum {
    WASH = 1,
    DRY_CLEAN,
    IRON
} ServiceType;
struct Order {
    int id;
    char customerName[50];
    ServiceType service;
    float weight;
    float price;
    Status status;
    char dateReceived[20];
};
struct LaundryShop {
    struct Order orders[MAX_ORDERS];
    int orderCount;
    float rates[3];  // rates for wash, dry clean, iron
};
void initializeLaundryShop(struct LaundryShop *shop) {
    shop->orderCount = 0;
    shop->rates[0] = 5.0 * 25;  // wash rate per kg
    shop->rates[1] = 8.0 * 30;  // dry clean rate per kg
    shop->rates[2] = 3.0 * 20;  // iron rate per kg
}
void addOrder(struct LaundryShop *shop) {
    if(shop->orderCount >= MAX_ORDERS) {
        printf("Orders full!\n");
        return;
    }
    struct Order newOrder;
    newOrder.id = shop->orderCount + 1001;
    printf("Enter customer name: ");
    scanf(" %[^\n]s", newOrder.customerName);
    printf("Select service (1-Wash, 2-Dry Clean, 3-Iron): ");
    scanf("%d", (int*)&newOrder.service);
    printf("Enter weight in kg: ");
    scanf("%f", &newOrder.weight);
    printf("Enter date (DD/MM/YYYY): ");
    scanf(" %[^\n]s", newOrder.dateReceived);
    newOrder.price = shop->rates[newOrder.service - 1] * newOrder.weight;
    newOrder.status = PENDING;
    shop->orders[shop->orderCount] = newOrder;
    shop->orderCount++;
    printf("Order added! Total price: P%.2f\n", newOrder.price);
}
void displayOrders(struct LaundryShop *shop) {
    printf("\nCurrent Orders:\n");
    printf("ID\tCustomer\t\tService\t\tStatus\t\tPrice\n");
    for(int i = 0; i < shop->orderCount; i++) {
        char *service[] = {"Wash", "Dry Clean", "Iron"};
        char *status[] = {"Pending", "Processing", "Ready", "Delivered"};
        printf("%d\t%-20s\t%-10s\t%-10s\tP%.2f\n",
            shop->orders[i].id,
            shop->orders[i].customerName,
            service[shop->orders[i].service - 1],
            status[shop->orders[i].status],
            shop->orders[i].price);
    }
    // Display current rates
    printf("\nCurrent Rates (per kg):\n");
    printf("Wash: P%.2f\n", shop->rates[0]);
    printf("Dry Clean: P%.2f\n", shop->rates[1]);
    printf("Iron: P%.2f\n", shop->rates[2]);
}
void updateOrderStatus(struct LaundryShop *shop) {
    int id, newStatus;
    printf("Enter order ID: ");
    scanf("%d", &id);
    for(int i = 0; i < shop->orderCount; i++) {
        if(shop->orders[i].id == id) {
            printf("Enter new status (0-Pending, 1-Processing, 2-Ready, 3-Delivered): ");
            scanf("%d", &newStatus);
            shop->orders[i].status = newStatus;
            printf("Status updated successfully!\n");
            return;
        }
    }
    printf("Order not found!\n");
}
int main() {
    struct LaundryShop shop;
    initializeLaundryShop(&shop);
    int choice;
    do {
        printf("\n==========Laundry Shop Management Program==========\n");
        printf("1. Add New Order\n");
        printf("2. Display Orders\n");
        printf("3. Update Order Status\n");
        printf("0. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);
        printf("\n");

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

```
==========Laundry Shop Management Program==========
1. Add New Order
2. Display Orders
3. Update Order Status
0. Exit
Enter choice: 1

Enter customer name: Elliot Anderson
Select service (1-Wash, 2-Dry Clean, 3-Iron): 1
Enter weight in kg: 3.5
Enter date (DD/MM/YYYY): 11/12/2024
Order added! Total price: P437.50

==========Laundry Shop Management Program==========
1. Add New Order
2. Display Orders
3. Update Order Status
0. Exit
Enter choice: 1

Enter customer name: Mary Johnson
Select service (1-Wash, 2-Dry Clean, 3-Iron): 2
Enter weight in kg: 2
Enter date (DD/MM/YYYY): 10/12/2024
Order added! Total price: P480.00
```

```
==========Laundry Shop Management Program==========
1. Add New Order
2. Display Orders
3. Update Order Status
0. Exit
Enter choice: 2


Current Orders:
ID      Customer                Service         Status          Price
1001    Elliot Anderson         Wash            Pending         P437.50
1002    Mary Johnson            Dry Clean       Pending         P480.00

Current Rates (per kg):
Wash: P125.00
Dry Clean: P240.00
Iron: P60.00
```

```
==========Laundry Shop Management Program==========
1. Add New Order
2. Display Orders
3. Update Order Status
0. Exit
Enter choice: 3

Enter order ID: 1001
Enter new status (0-Pending, 1-Processing, 2-Ready, 3-Delivered): 1
Status updated successfully!
```

```
==========Laundry Shop Management Program==========
1. Add New Order
2. Display Orders
3. Update Order Status
0. Exit
Enter choice: 0

Goodbye!

Process returned 0 (0x0)   execution time : 76.669 s
Press any key to continue.
```