University of Wolverhampton

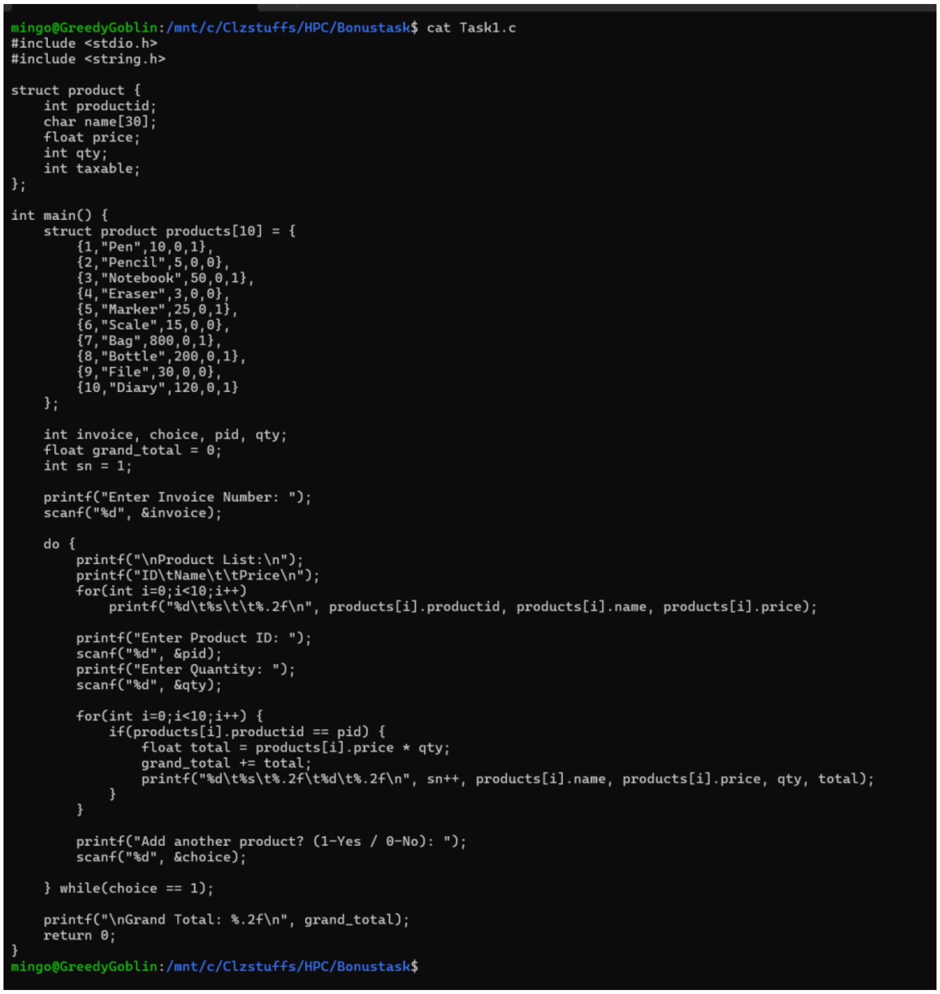
School of Mathematics and Computer Science

6CS005 High Performance Computing

Student ID:2407710

Name: Dhadkan K.C.
Tasks – Bonus Tasks

1.Create a product bag in C using an array with fields: productid, name, price, qty, taxable (0/1) Add at least 10 products. Create a setting_conf structure with tax_rate. When program runs: Ask for invoice number Display product list Ask user to choose productid and qty Ask user if they want to add another product If yes → show same product list and take input again If no → print bill with: SN Product Name Product Price Qty Total At bottom show grand total.

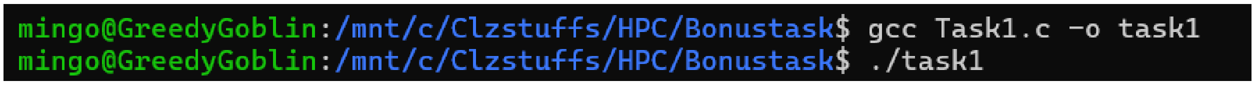


```
mingo@GreedyGoblin:/mnt/c/Clzstuffs/HPC/Bonustask$ cat Task1.c
#include <stdio.h>
#include <string.h>

struct product {
    int productid;
    char name[30];
    float price;
    int qty;
    int taxable;
};

int main() {
    struct product products[10] = {
        {1,"Pen",10,0,1},
        {2,"Pencil",5,0,0},
        {3,"Notebook",50,0,1},
        {4,"Eraser",3,0,0},
        {5,"Marker",25,0,1},
        {6,"Scale",15,0,0},
        {7,"Bag",800,0,1},
        {8,"Bottle",200,0,1},
        {9,"File",30,0,0},
        {10,"Diary",120,0,1}
    };

    int invoice, choice, pid, qty;
    float grand_total = 0;
    int sn = 1;

    printf("Enter Invoice Number: ");
    scanf("%d", &invoice);

    do {
        printf("\nProduct List:\n");
        printf("ID\tName\t\tPrice\n");
        for(int i=0;i<10;i++)
            printf("%d\t%s\t\t%.2f\n", products[i].productid, products[i].name, products[i].price);

        printf("Enter Product ID: ");
        scanf("%d", &pid);
        printf("Enter Quantity: ");
        scanf("%d", &qty);

        for(int i=0;i<10;i++) {
            if(products[i].productid == pid) {
                float total = products[i].price * qty;
                grand_total += total;
                printf("%d\t%s\t%.2f\t%d\t%.2f\n", sn++, products[i].name, products[i].price, qty, total);
            }
        }

        printf("Add another product? (1-Yes / 0-No): ");
        scanf("%d", &choice);

    } while(choice == 1);

    printf("\nGrand Total: %.2f\n", grand_total);
    return 0;
}
mingo@GreedyGoblin:/mnt/c/Clzstuffs/HPC/Bonustask$
```

The program adopts a basic product billing system that is executed in C programming language. It establishes a product structure where product information is kept like product ID, name, price, quantities and taxable status of the product. There is a pre-defined array of 10 products which serves as a product bag. The tax rate is stored in another structure setting_conf. Once the program has been loaded, the user is requested to input an invoice number and the list of products available is displayed. The user will select a product by typing a product ID and quantity in the product. The program then sums up the total price of that product, charges a tax on the same when it is indicated that the product is taxable and then add the price to the overall price. This is done in a loop until the user decides that he or she does not want to place additional products.

Compilation:

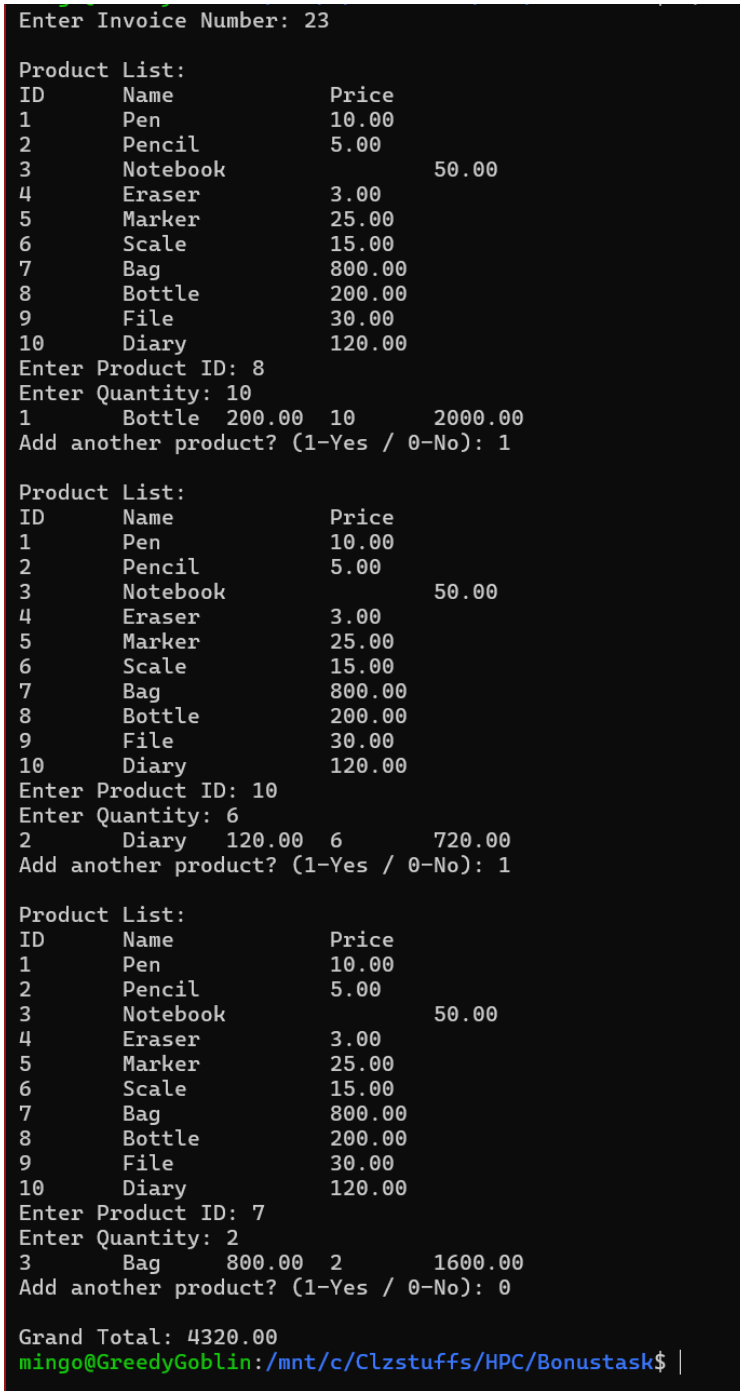
The command gcc Task1.c -o task1 is used in the terminal to compile the source file Task1.c. In this case, the C source file is Task1.c and the name of the executable file that will be generated is task1. In case syntax errors are absent in the code, the compiler manages to create the executable. This is followed by the execution of the program by the command ./task1.



```
mingo@GreedyGoblin:/mnt/c/Clzstuffs/HPC/Bonustask$ gcc Task1.c -o task1
mingo@GreedyGoblin:/mnt/c/Clzstuffs/HPC/Bonustask$ ./task1
```

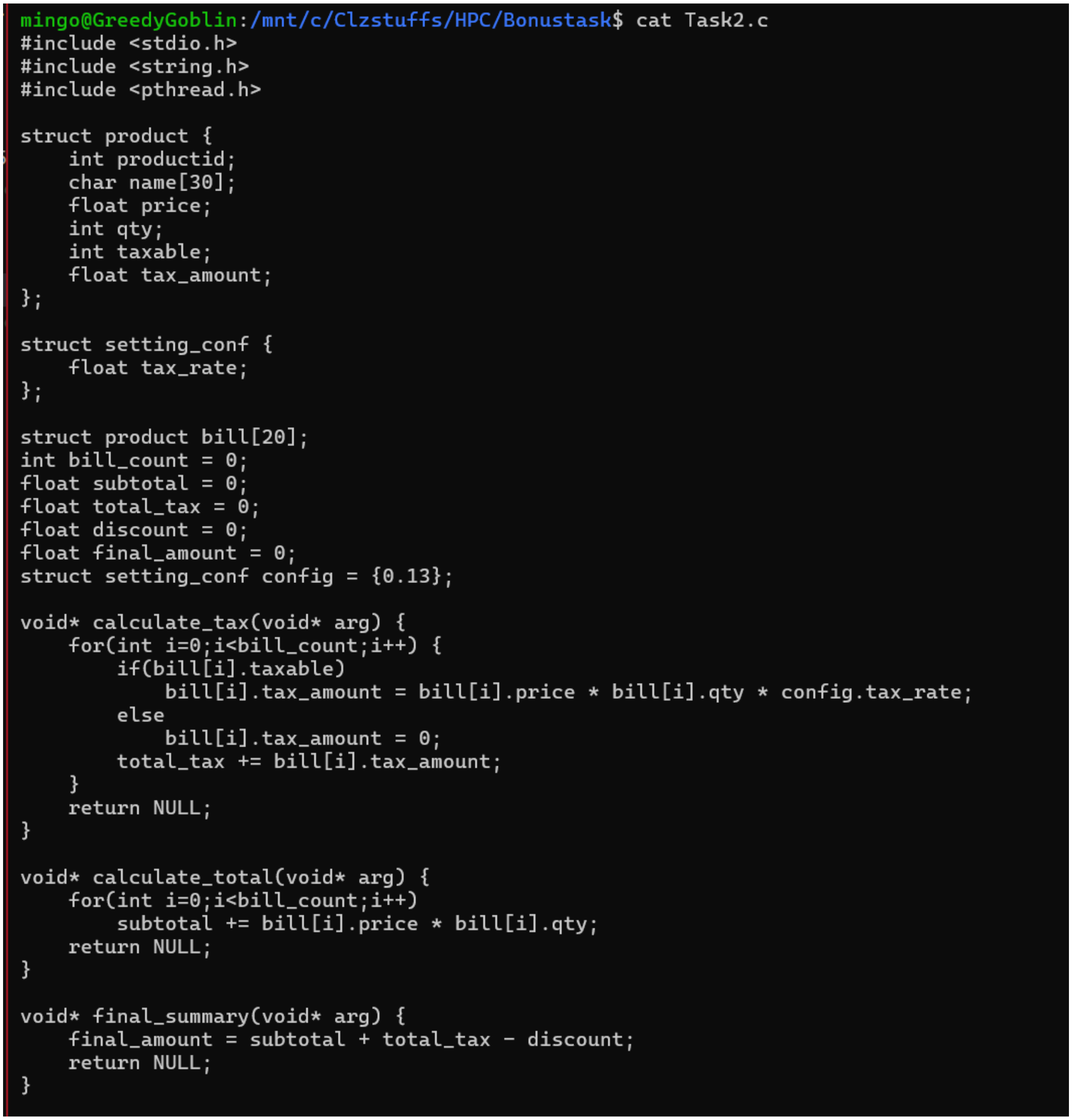
Output:

The first step that the program takes is to request the user to input an invoice number when the program is being implemented. Subsequently, it shows a list of the products with their IDs and prices. The program also displays as the user chooses items and quantities, one item per line in bill-like statement with the serial number, name of the item, price, quantity and total cost of the item. After adding the goods, a user may wish to add, the program then shows the total amount at the bottom, and that is, the total amount of the products chosen including the relevant tax

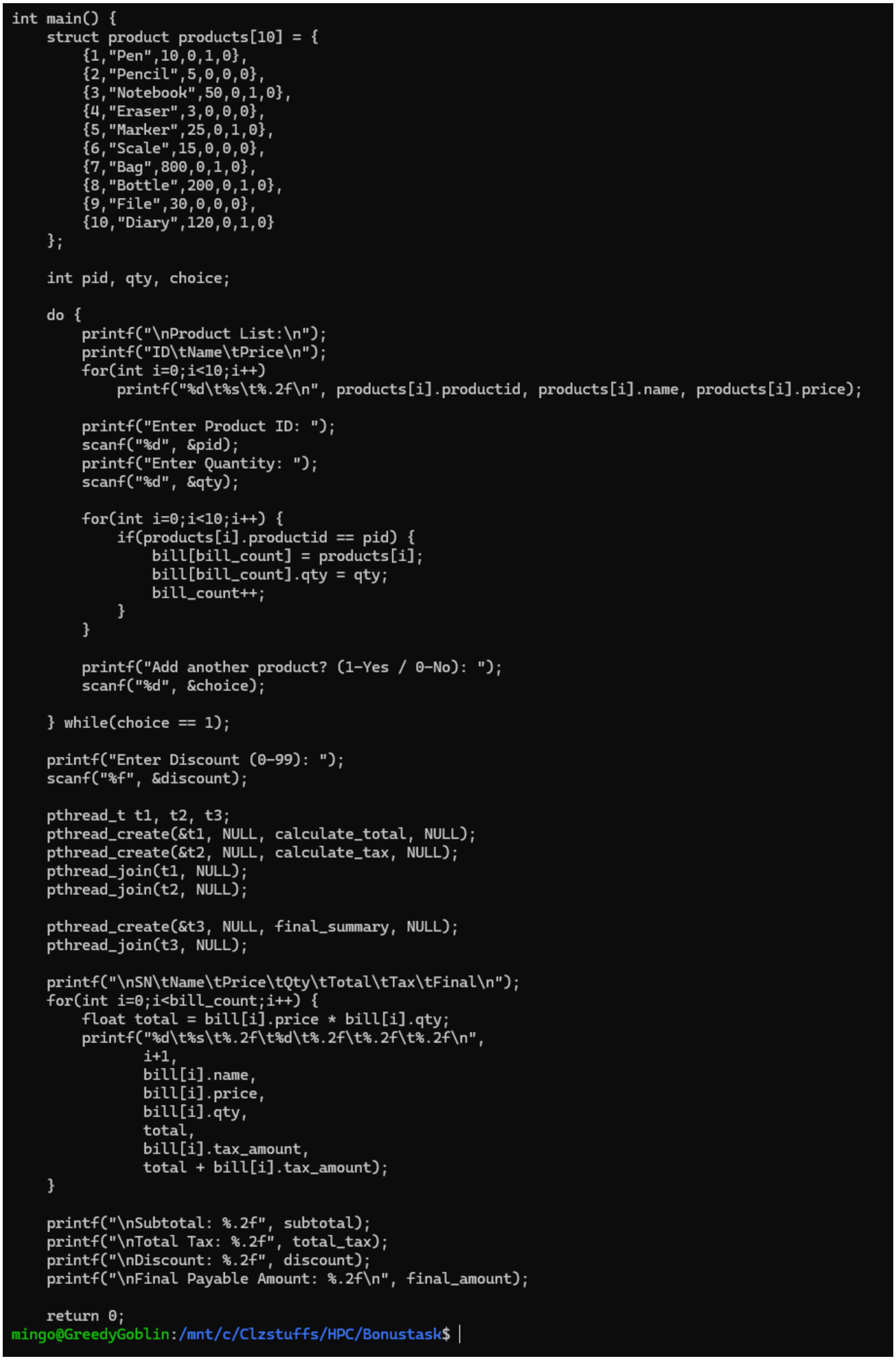


```
Enter Invoice Number: 23

Product List:
ID      Name            Price
1       Pen             10.00
2       Pencil          5.00
3       Notebook                50.00
4       Eraser          3.00
5       Marker          25.00
6       Scale           15.00
7       Bag             800.00
8       Bottle          200.00
9       File            30.00
10      Diary           120.00
Enter Product ID: 8
Enter Quantity: 10
1       Bottle  200.00  10      2000.00
Add another product? (1-Yes / 0-No): 1

Product List:
ID      Name            Price
1       Pen             10.00
2       Pencil          5.00
3       Notebook                50.00
4       Eraser          3.00
5       Marker          25.00
6       Scale           15.00
7       Bag             800.00
8       Bottle          200.00
9       File            30.00
10      Diary           120.00
Enter Product ID: 10
Enter Quantity: 6
2       Diary   120.00  6       720.00
Add another product? (1-Yes / 0-No): 1

Product List:
ID      Name            Price
1       Pen             10.00
2       Pencil          5.00
3       Notebook                50.00
4       Eraser          3.00
5       Marker          25.00
6       Scale           15.00
7       Bag             800.00
8       Bottle          200.00
9       File            30.00
10      Diary           120.00
Enter Product ID: 7
Enter Quantity: 2
3       Bag     800.00  2       1600.00
Add another product? (1-Yes / 0-No): 0

Grand Total: 4320.00
mingo@GreedyGoblin:/mnt/c/Clzstuffs/HPC/Bonustask$
```

2. Incremental Requirements (Task 2 Extension) Add new field tax_amount inside the product bill calculation. Calculate tax_amount only if taxable == 1 using setting_conf.tax_rate. Show tax_amount in the bill table alongside: SN Product Name Product Price Qty Total Tax Amount Final Total (price*qty + tax_amount) At the end of the bill, show: Subtotal (sum of item totals without tax) Total Tax Discount % (user input 0–99) Discount Amount Final Payable Amount (subtotal + total_tax – discount) Use multithreading for any calculation step (tax calculation, total calculation, final summary).



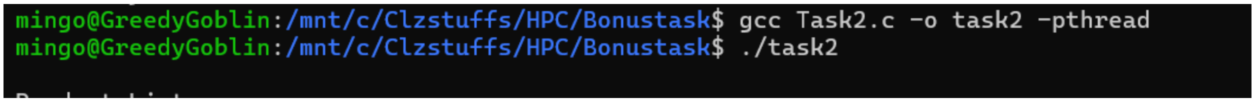
```
mingo@GreedyGoblin:/mnt/c/Clzstuffs/HPC/Bonustask$ cat Task2.c
#include <stdio.h>
#include <string.h>
#include <pthread.h>

struct product {
    int productid;
    char name[30];
    float price;
    int qty;
    int taxable;
    float tax_amount;
};

struct setting_conf {
    float tax_rate;
};

struct product bill[20];
int bill_count = 0;
float subtotal = 0;
float total_tax = 0;
float discount = 0;
float final_amount = 0;
struct setting_conf config = {0.13};

void* calculate_tax(void* arg) {
    for(int i=0;i<bill_count;i++) {
        if(bill[i].taxable)
            bill[i].tax_amount = bill[i].price * bill[i].qty * config.tax_rate;
        else
            bill[i].tax_amount = 0;
        total_tax += bill[i].tax_amount;
    }
    return NULL;
}

void* calculate_total(void* arg) {
    for(int i=0;i<bill_count;i++)
        subtotal += bill[i].price * bill[i].qty;
    return NULL;
}

void* final_summary(void* arg) {
    final_amount = subtotal + total_tax - discount;
    return NULL;
}
```



```
int main() {
    struct product products[10] = {
        {1,"Pen",10,0,1,0},
        {2,"Pencil",5,0,0,0},
        {3,"Notebook",50,0,1,0},
        {4,"Eraser",3,0,0,0},
        {5,"Marker",25,0,1,0},
        {6,"Scale",15,0,0,0},
        {7,"Bag",800,0,1,0},
        {8,"Bottle",200,0,1,0},
        {9,"File",30,0,0,0},
        {10,"Diary",120,0,1,0}
    };

    int pid, qty, choice;

    do {
        printf("\nProduct List:\n");
        printf("ID\tName\tPrice\n");
        for(int i=0;i<10;i++)
            printf("%d\t%s\t%.2f\n", products[i].productid, products[i].name, products[i].price);

        printf("Enter Product ID: ");
        scanf("%d", &pid);
        printf("Enter Quantity: ");
        scanf("%d", &qty);

        for(int i=0;i<10;i++) {
            if(products[i].productid == pid) {
                bill[bill_count] = products[i];
                bill[bill_count].qty = qty;
                bill_count++;
            }
        }

        printf("Add another product? (1-Yes / 0-No): ");
        scanf("%d", &choice);

    } while(choice == 1);

    printf("Enter Discount (0-99): ");
    scanf("%f", &discount);

    pthread_t t1, t2, t3;
    pthread_create(&t1, NULL, calculate_total, NULL);
    pthread_create(&t2, NULL, calculate_tax, NULL);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);

    pthread_create(&t3, NULL, final_summary, NULL);
    pthread_join(t3, NULL);

    printf("\nSN\tName\tPrice\tQty\tTotal\tTax\tFinal\n");
    for(int i=0;i<bill_count;i++) {
        float total = bill[i].price * bill[i].qty;
        printf("%d\t%s\t%.2f\t%d\t%.2f\t%.2f\t%.2f\n",
               i+1,
               bill[i].name,
               bill[i].price,
               bill[i].qty,
               total,
               bill[i].tax_amount,
               total + bill[i].tax_amount);
    }

    printf("\nSubtotal: %.2f", subtotal);
    printf("\nTotal Tax: %.2f", total_tax);
    printf("\nDiscount: %.2f", discount);
    printf("\nFinal Payable Amount: %.2f\n", final_amount);

    return 0;
mingo@GreedyGoblin:/mnt/c/Clzstuffs/HPC/Bonustask$
```

It is an advanced product billing system in C which employs multithreading using pthreads. It creates a product structure which is used to store product information such as tax information and a setting-conf structure used to store the tax rate. The user is able to choose products and quantities in a predetermined list and the chosen items are placed in another array bill. Once the product is selected the program requests a discount. This is followed by three different threads, one thread is used to determine the subtotal, another to determine the tax to the taxable products, and a third to determine the final payable amount. With threads, one can make various computations (subtotal, tax, and final total) that can be separated with shared information

Compilation:

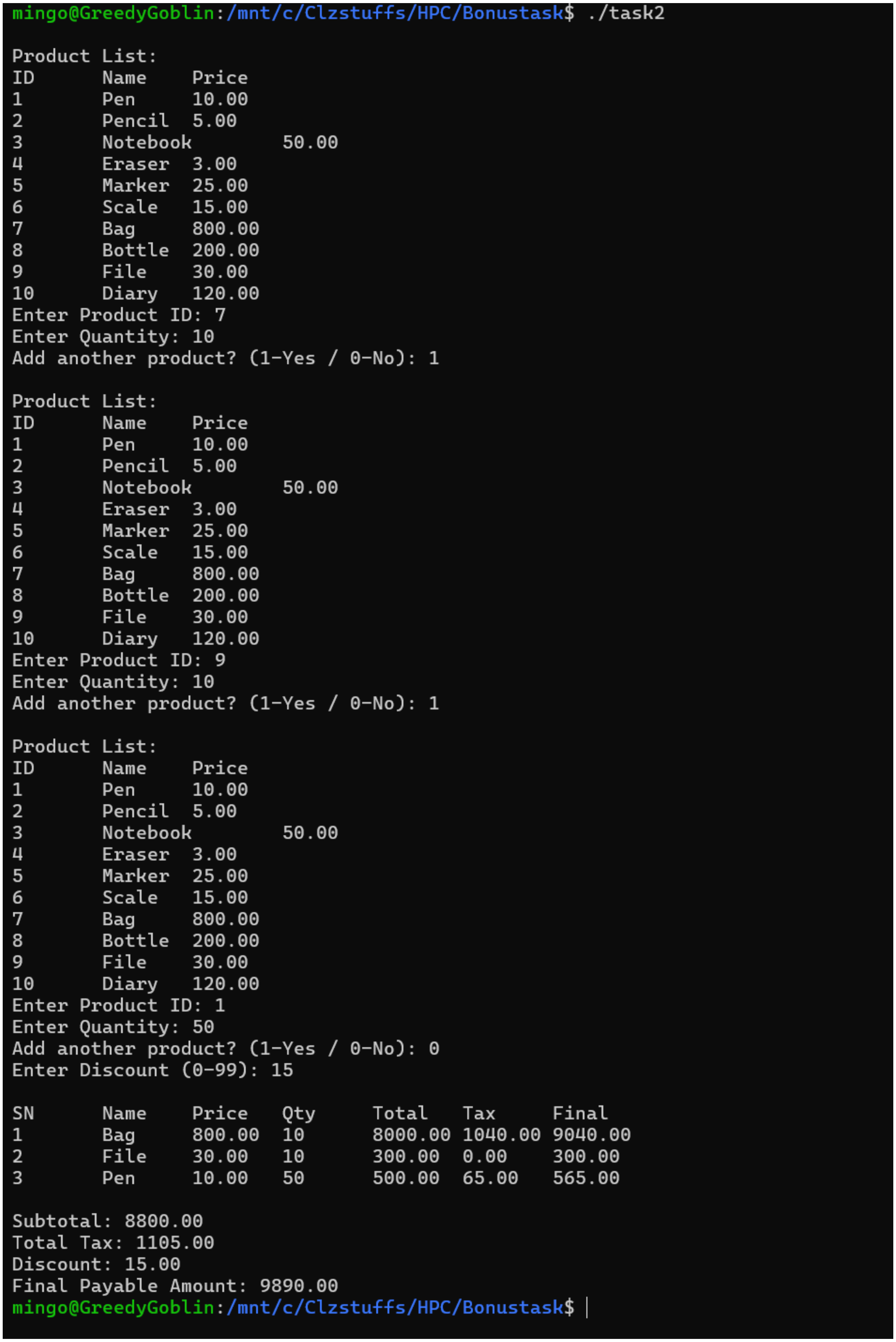


```
mingo@GreedyGoblin:/mnt/c/Clzstuffs/HPC/Bonustask$ gcc Task2.c -o task2 -pthread
mingo@GreedyGoblin:/mnt/c/Clzstuffs/HPC/Bonustask$ ./task2
```

Since the application is based on the POSIX threads library (pthread.h), it has to be compiled with the -pthread flag. The compilation process is executed by use of command gcc Task2.c -o task2 -pthread in which the source file (say Task2.c) is compiled to create an executable task1. The purpose of this command is to make the compiler connect the pthread library. In case the compilation is successful and no errors were found, a running program called task1 is generated. Then the program is run with the help of ./task2 in the terminal.

Output:

The output underlines that the program initially tells the list of the products and requests the user to input a product ID and a product quantity, and the process repeats itself until the user decides to add no more products. Once the products have been selected, the program requests the user to input the value of the discount. It then sums the price and quantity of each item chosen to compute the total price of the item and imposes taxes only on items that are marked taxable using the set tax rate and shows each item with both its total and tax. Lastly, the program sums up the subtotal, the total tax, discount that is inputted in the program and shows the final payable amount.



```
mingo@GreedyGoblin:/mnt/c/Clzstuffs/HPC/Bonustask$ ./task2

Product List:
ID      Name    Price
1       Pen     10.00
2       Pencil  5.00
3       Notebook        50.00
4       Eraser  3.00
5       Marker  25.00
6       Scale   15.00
7       Bag     800.00
8       Bottle  200.00
9       File    30.00
10      Diary   120.00
Enter Product ID: 7
Enter Quantity: 10
Add another product? (1-Yes / 0-No): 1

Product List:
ID      Name    Price
1       Pen     10.00
2       Pencil  5.00
3       Notebook        50.00
4       Eraser  3.00
5       Marker  25.00
6       Scale   15.00
7       Bag     800.00
8       Bottle  200.00
9       File    30.00
10      Diary   120.00
Enter Product ID: 9
Enter Quantity: 10
Add another product? (1-Yes / 0-No): 1

Product List:
ID      Name    Price
1       Pen     10.00
2       Pencil  5.00
3       Notebook        50.00
4       Eraser  3.00
5       Marker  25.00
6       Scale   15.00
7       Bag     800.00
8       Bottle  200.00
9       File    30.00
10      Diary   120.00
Enter Product ID: 1
Enter Quantity: 50
Add another product? (1-Yes / 0-No): 0
Enter Discount (0-99): 15

SN      Name    Price   Qty     Total   Tax     Final
1       Bag     800.00  10      8000.00 1040.00 9040.00
2       File    30.00   10      300.00  0.00    300.00
3       Pen     10.00   50      500.00  65.00   565.00

Subtotal: 8800.00
Total Tax: 1105.00
Discount: 15.00
Final Payable Amount: 9890.00
mingo@GreedyGoblin:/mnt/c/Clzstuffs/HPC/Bonustask$ 
```