Institute of Computer Technology

B. Tech. Computer Science and Engineering

Sub: ESFP – II

Course Code: 2CSE203

Practical – 3

Name: Jaymin Gondaliya

Enrollment No: 23162171007

Sem - 2

Branch: CS

Class: B

Batch: 25

Objective:

To learn DMA (Dynamic memory allocation) and Single Linked-list

Problem Definition-1:

In a model town, there is one stationary shop where you can purchase all cosmetic product items. So, the shop owner wants to make a project for his shop for managing product sales and purchasing record status in a proper format. For that, you have to make a program. where, if a customer wants to purchase a product from a shop, for that, you have to take input as product_id, product_name, product_qty, product_price from customer. Accordingly, you have to print the purchase bill on screen as product_id, product_name, product_qty, product_price and product total_price format. And as per customer choice you can also search the product list item from store by product_id or product_name, if you want to delete records from purchase list you can also perform. So, as per the above given scenario make a proper dynamic memory allocation program with the help of structure, where you have to perform all above given said requirements.

Code:

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

struct Product
```

```c
{
    int id;
    char name[50];
    int qty;
    float price;
    float total;
    struct Product *next;
};

struct Product *head = NULL;

void addProduct()
{
    struct Product *new = (struct Product *)malloc(sizeof(struct Product));
    if (new == NULL)
    {
        printf("Memory allocation fail\n");
        exit(1);
    }
    printf("Enter Product ID: ");
    scanf("%d", &new->id);
    printf("Enter Product Name: ");
    scanf("%s", new->name);
    printf("Enter Product Quantity: ");
    scanf("%d", &new->qty);
    printf("Enter Product Price: ");
    scanf("%f", &new->price);
    new->total = new->qty * new->price;
    new->next = head;
    head = new;
}

void displayProducts()
{
    struct Product *current = head;
    printf("ID\tName\tQty\tprice\ttotal\n");
    while (current != NULL)
    {
        printf("%d\t%s\t%d\t%.2f\t%.2f\n", current->id, current->name,
               current->qty, current->price, current->total);
        current = current->next;
    }
}

void searchProduct()
{
    char searchStr[50];
    printf("Enter Product ID or Name to search: ");
```

```
    scanf("%s", searchStr);
    struct Product *current = head;
    while (current != NULL)
    {
        if (current->id == atoi(searchStr) || strcmp(current->name, searchStr)
== 0)
        {
            printf("ID\tName\tQty\tprice\ttotal\n");
            printf("%d\t%s\t%d\t%.2f\t%.2f\n", current->id, current->name,
                   current->qty, current->price, current->total);
            return;
        }
        current = current->next;
    }
    printf("Product with ID/Name %s not found.\n", searchStr);
}

void deleteProduct()
{
    char deleteStr[50];
    printf("Enter Product ID or Name to delete: ");
    scanf("%s", deleteStr);
    struct Product *current = head, *prev = NULL;
    while (current != NULL)
    {
        if (current->id == atoi(deleteStr) || strcmp(current->name, deleteStr)
== 0)
        {
            if (prev == NULL)
            {
                head = current->next;
            }
            else
            {
                prev->next = current->next;
            }
            free(current);
            printf("Product with ID/Name %s deleted successfully.\n",
deleteStr);
            return;
        }
        prev = current;
        current = current->next;
    }
    printf("Product with ID/Name %s not found.\n", deleteStr);
}

int main()
```

```c
{
    int num, choice;
    printf("How many products you want to purchase: ");
    scanf("%d", &num);

    for (int i = 0; i < num; i++)
    {
        printf("\nProduct %d:\n", i + 1);
        addProduct();
    }

    while (1)
    {
        printf("\n1. Display Products\n");
        printf("2. Search Product\n");
        printf("3. Delete Product\n");
        printf("4. Exit\n");
        printf("Enter Your Choice: ");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            displayProducts();
            break;
        case 2:
            searchProduct();
            break;
        case 3:
            deleteProduct();
            break;
        case 4:
            exit(0);
        default:
            printf("Invalid Choice\n");
        }
    }
    return 0;
}
```

Output –

```
Enter Product Name: mouse
Enter Product Quantity: 1
Enter Product Price: 500

1. Display Products
2. Search Product
3. Delete Product
4. Exit
Enter Your Choice: 1
ID      Name    Qty     price   total
4949    mouse   1       500.00  500.00
5050    watch   1       400.00  400.00
4006    pen     2       15.00   30.00

1. Display Products
2. Search Product
3. Delete Product
4. Exit
Enter Your Choice: 2
Enter Product ID or Name to search: mouse
ID      Name    Qty     price   total
4949    mouse   1       500.00  500.00

1. Display Products
2. Search Product
3. Delete Product
4. Exit
Enter Your Choice: 3
Enter Product ID or Name to delete: 4006
Product with ID/Name 4006 deleted successfully.

1. Display Products
2. Search Product
3. Delete Product
4. Exit
Enter Your Choice: 1
ID      Name    Qty     price   total
4949    mouse   1       500.00  500.00
5050    watch   1       400.00  400.00

1. Display Products
2. Search Product
3. Delete Product
4. Exit
Enter Your Choice:
```