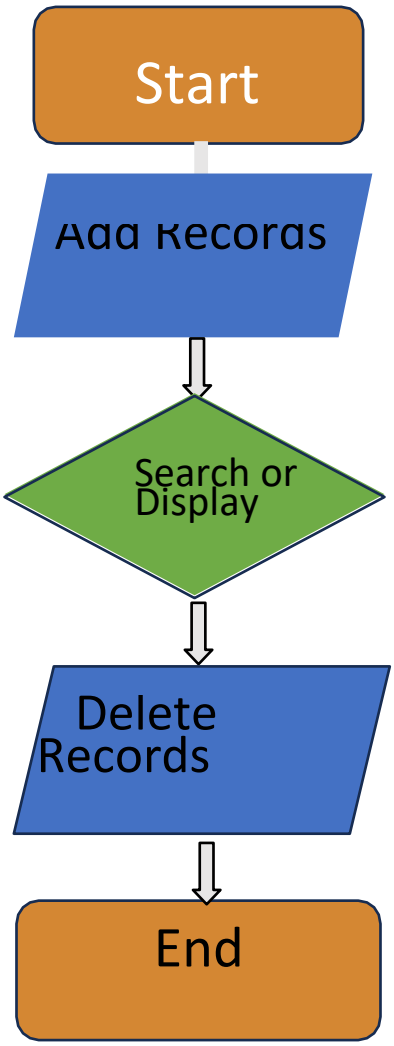
Name: Chauhan Manan Manishbhai
E_ID:23162171004
Class :- B
ESFP-II

Practical 3:

DMA: Definition: Purchase Billing Report.

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.

[Note: Perform this program using a single linked-list concept].


Start
Add Records
Search or Display
Delete Records
End

Code :

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct records {
    int id;
    char name[20];
    int qty;
    float price;
    float tpri;
    struct records* next;
};
struct records* start;
struct records* end;
int addRecord() {
    struct records* temp;
    temp = (struct records*)malloc(sizeof(struct records));
    printf("Enter PID, Pname, Pqty, Pprice: ");
    scanf("%d %s %d %f", &temp->id, temp->name, &temp->qty, &temp->price);
    temp->tpri = temp->qty * temp->price;
    if (start == NULL) {
        temp->next = NULL;
        start = temp;
        end = temp;
    } else {
        end->next = temp;
        end = temp;
        end->next = NULL;
    }
    return 0;
}
int displayRecords() {
    struct records* ptr;
    if (start == NULL) {
        printf("List is blank\n");
    } else {
        printf("========== Purchased Output Product Information ==========\n");
        printf("PID PName PQty Pprice PTprice\n");
```

```c
        for (ptr = start; ptr != NULL; ptr = ptr->next) {
            printf("%d %s %d %f %f\n", ptr->id, ptr->name, ptr->qty, ptr->price, ptr->tpri);
        }
    }
    return 0;
}
int searchRecord() {
    char choice;
    int searchId;
    char searchName[20];
    printf("Do you want to find individual product information (y/n)?: ");
    scanf(" %c", &choice);
    if (choice == 'y' || choice == 'Y') {
       int searchOption;
       printf("How do you want to find product information by ID or by Name: 1 for ID and 2 forName:")
        scanf("%d", &searchOption);
        if (searchOption == 1) {
            printf("Enter product Id: ");
            scanf("%d", &searchId);
        } else if (searchOption == 2) {
            printf("Enter product Name: ");
            scanf("%s", searchName);
        } else {
            printf("Invalid option\n");
            return;
        }
        struct records* ptr = start;
        printf("PID PName PQty Pprice PTprice\n");
        while (ptr != NULL) {
            if ((searchOption == 1 && ptr->id == searchId) || (searchOption == 2 && strcmp(ptr->name,
                searchName) == 0)) {
                printf("%d %s %d %f %f\n", ptr->id, ptr->name, ptr->qty, ptr->price, ptr->tpri);
                break;
            }
            ptr = ptr->next;
        }
        if (ptr == NULL) {
```

```
75              printf("Product not found\n");
76          }
77      }
78  }
79  int deleteRecord() {
80      char choice;
81      int deleteId;
82      char deleteName[20];
83      printf("Do you want to delete product record (y/n): ");
84      scanf(" %c", &choice);
85      if (choice == 'y' || choice == 'Y') {
86          int deleteOption;
87          printf("How do you want to delete record by ID or By name? 1 for by ID and 2 for by Name: ");
88          scanf("%d", &deleteOption);
89          if (deleteOption == 1) {
90              printf("Enter Product ID: ");
91              scanf("%d", &deleteId);
92          } else if (deleteOption == 2) {
93              printf("Enter Product Name: ");
94              scanf("%s", deleteName);
95          } else {
96              printf("Invalid option\n");
97              return;
98          }
99          struct records* current = start;
100         struct records* prev = NULL;
101         while (current != NULL) {
102             if ((deleteOption == 1 && current->id == deleteId) || (deleteOption == 2 &&
103             strcmp(current->name, deleteName) == 0)) {
104                 if (prev == NULL) {
105                     start = current->next;
106                 } else {
107                     prev->next = current->next;
108                 }
109                 free(current);
110                 printf("============= Updated Record ============\n");
111                 displayRecords();
```

```c
            break;
        }
        prev = current;
        current = current->next;
    }
    if (current == NULL) {
        printf("Product not found\n");
    }
  }
}
int main() {
    int choice;
    do {
        printf("\n1. Add Record\n");
        printf("2. Display Records\n");
        printf("3. Search Record\n");
        printf("4. Delete Record\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                addRecord();
                break;
            case 2:
                displayRecords();
                break;
            case 3:
                searchRecord();
                break;
            case 4:
                deleteRecord();
                break;
            case 5:
                printf("Exiting the program\n");
                break;
            default:
                printf("Invalid choice, please enter again.\n");
        }
    } while (choice != 5);
    return 0;
}
```

Output:

```
 1. Add Record
 2. Display Records
 3. Search Record
 4. Delete Record
 5. Exit
 Enter your choice: 1
 Enter PID, Pname, Pqty, Pprice: 1 Bat 2 15000

 1. Add Record
 2. Display Records
 3. Search Record
 4. Delete Record
 5. Exit
 Enter your choice: 1
 Enter PID, Pname, Pqty, Pprice: 2 Bowl 5 75

 1. Add Record
 2. Display Records
 3. Search Record
 4. Delete Record
 5. Exit
 Enter your choice: 1
 Enter PID, Pname, Pqty, Pprice: 3 Glove 3 1500

 1. Add Record
 2. Display Records
 3. Search Record
 4. Delete Record
 5. Exit
 Enter your choice: 2
 ========== Purchased Output Product Information ============
 PID PName PQty Pprice PTprice
 1 Bat 2 15000.000000 30000.000000
 2 Bowl 5 75.000000 375.000000
 3 Glove 3 1500.000000 4500.000000

 1. Add Record
 2. Display Records
 3. Search Record
 4. Delete Record
 5. Exit
 Enter your choice: 3
 Do you want to find individual product information (y/n)?: y
 How do you want to find product information by ID or by Name: 1 for ID and 2 for Name: 2
 Enter product Name: Glove
PID PName PQty Pprice PTprice
3 Glove 3 1500.000000 4500.000000

1. Add Record
2. Display Records
3. Search Record
4. Delete Record
5. Exit
Enter your choice: 4
Do you want to delete product record (y/n): y
How do you want to delete record by ID or By name? 1 for by ID and 2 for by Name: 1
Enter Product ID: 3
============ Updated Record ============
========== Purchased Output Product Information ============
PID PName PQty Pprice PTprice
1 Bat 2 15000.000000 30000.000000
2 Bowl 5 75.000000 375.000000

1. Add Record
2. Display Records
3. Search Record
4. Delete Record
5. Exit
Enter your choice: 5
Exiting the program
```