***Institute of Computer And Technology***

***B.Tech – CSE(BDA)***

***Name:- Dwij Vatsal Desai***

***Sem:- 2***

***Sub: - ESFP-II***

***Enrollment No.:- 23162121027***

***Prac:- 3***

***Date:- 3/2/2024***

**Q.1.**

**Q.2. 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].

**Algorithm:-**

1. Start

2. Create a structure for entering data about students.

3. Program a code with the use of DMA.

4. Collect the Data from the user.

5. Show the data using printf.

6. Select a person's name.

7. Show the data of the person's name.

8. End

**Code:-**

*/\**

*Name:- Dwij desai*

*Enrollment No.:- 23162121027*

*Prac:- 3*

*\*/*

*#include* <stdio.h>

*#include* <stdlib.h>

*#include* <malloc.h>

*#include* <string.h>

struct Car

{

    int CID;

    char Cname[30];

    char Cprice[20];

    char Ccolor[20];

    struct Car \*next;

};

struct Car \*newnode, \*head = NULL, \*end = NULL;

void Last()

{

    newnode = (struct Car \*)malloc(sizeof(struct Car));

    printf("Enter the value as: ID of car, Name of car, Price, color of car:-\n");

    scanf("%d %s %s %s", &newnode->CID, newnode->Cname, newnode->Cprice, newnode->Ccolor);

*if* (head == NULL)

    {

        newnode->next = NULL;

        head = newnode;

        end = newnode;

    }

*else*

    {

        end->next = newnode;

        end = newnode;

        end->next = NULL;

    }

}

void display()

{

    struct Car \*ttemp;

*if* (head == NULL)

    {

        printf("List is empty\n");

    }

*else*

    {

        printf("\nDisplay value:\n");

*for* (ttemp = head; ttemp != NULL; ttemp = ttemp->next)

        {

            printf("%d %s %s %s\n", ttemp->CID, ttemp->Cname, ttemp->Cprice, ttemp->Ccolor);

        }

        printf("\n");

    }

}

void findProduct()

{

    int choice, id;

    char name[30];

    printf("Enter choice (1 for ID, 2 for Name): ");

    scanf("%d", &choice);

*if* (choice == 1)

    {

        printf("Enter Car ID to find: ");

        scanf("%d", &id);

    }

*else* *if* (choice == 2)

    {

        printf("Enter Car Name to find: ");

        scanf("%s", name);

    }

    struct Car \*temp = head;

    int found = 0;

*while* (temp != NULL)

    {

*if* ((choice == 1 && temp->CID == id) || (choice == 2 && strcmp(temp->Cname, name) == 0))

        {

            printf("Car found: %d %s %s %s\n", temp->CID, temp->Cname, temp->Cprice, temp->Ccolor);

            found = 1;

        }

        temp = temp->next;

    }

*if* (!found)

    {

        printf("Car not found.\n");

    }

}

void deleteProduct()

{

    int choice, id;

    char name[30];

    printf("Enter choice (1 for ID, 2 for Name): ");

    scanf("%d", &choice);

*if* (choice == 1)

    {

        printf("Enter Car ID to delete: ");

        scanf("%d", &id);

    }

*else* *if* (choice == 2)

    {

        printf("Enter Car Name to delete: ");

        scanf("%s", name);

    }

    struct Car \*temp = head;

    struct Car \*prev = NULL;

*while* (temp != NULL)

    {

*if* ((choice == 1 && temp->CID == id) || (choice == 2 && strcmp(temp->Cname, name) == 0))

        {

*if* (prev == NULL)

            {

                head = temp->next;

            }

*else*

            {

                prev->next = temp->next;

            }

            free(temp);

            printf("Car deleted successfully.\n");

*return*;

        }

        prev = temp;

        temp = temp->next;

    }

    printf("Car not found.\n");

}

int main()

{

    int a = 0;

*for* (;;)

    {

        printf("Press <1> to add value at end \n");

        printf("Press <2> to display value \n");

        printf("Press <3> to find car \n");

        printf("Press <4> to delete car \n");

        printf("Press <5> to end code \n");

        printf("\nEnter number for menu:  ");

        scanf("%d", &a);

*switch* (a)

        {

*case* 1:

            int num\_Car;

            printf("\nHow many units do you want: ");

            scanf("%d", &num\_Car);

*for* (int i = 0; i < num\_Car; i++)

            {

                Last();

            }

*break*;

*case* 2:

            display();

*break*;

*case* 3:

            findProduct();

*break*;

*case* 4:

            deleteProduct();

*break*;

*case* 5:

*return* 0;

*default*:

            printf("Enter right number\n");

*break*;

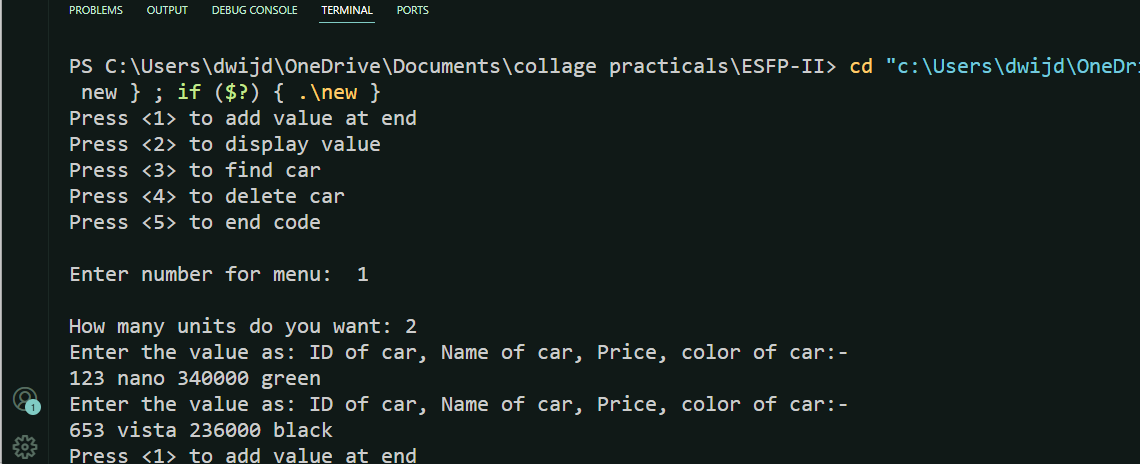
        }

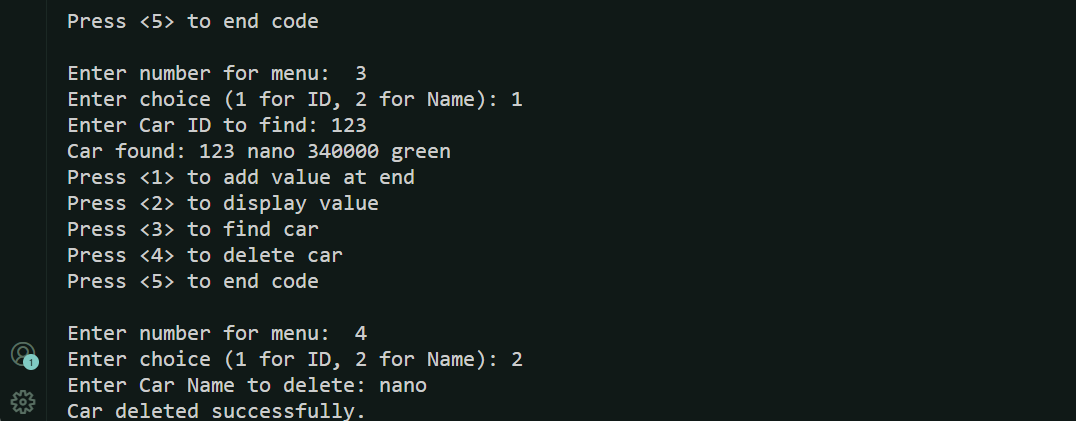
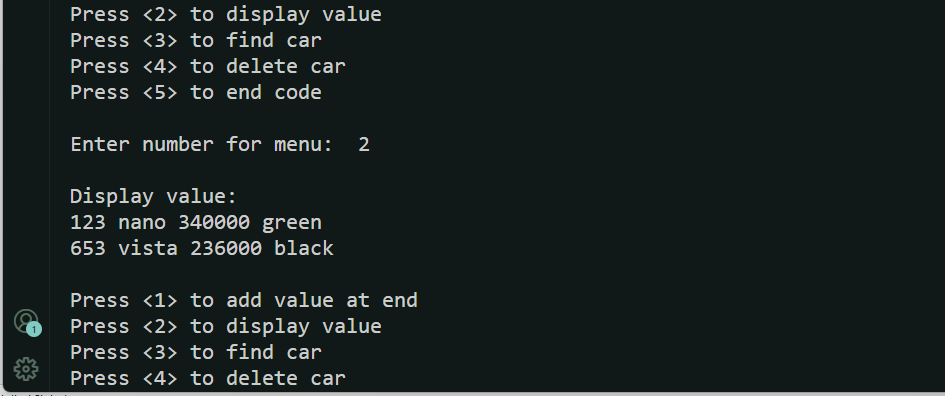
    }

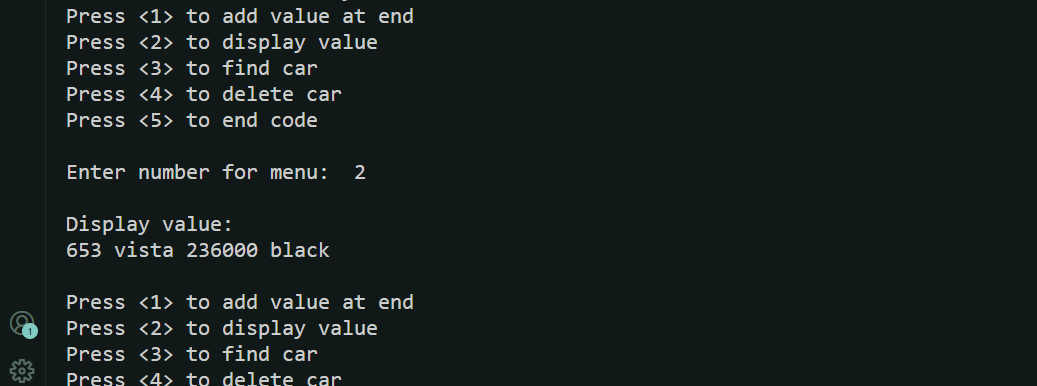
*return* 0;

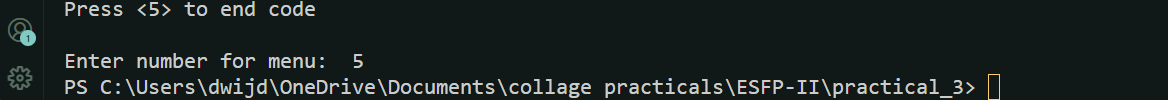
}

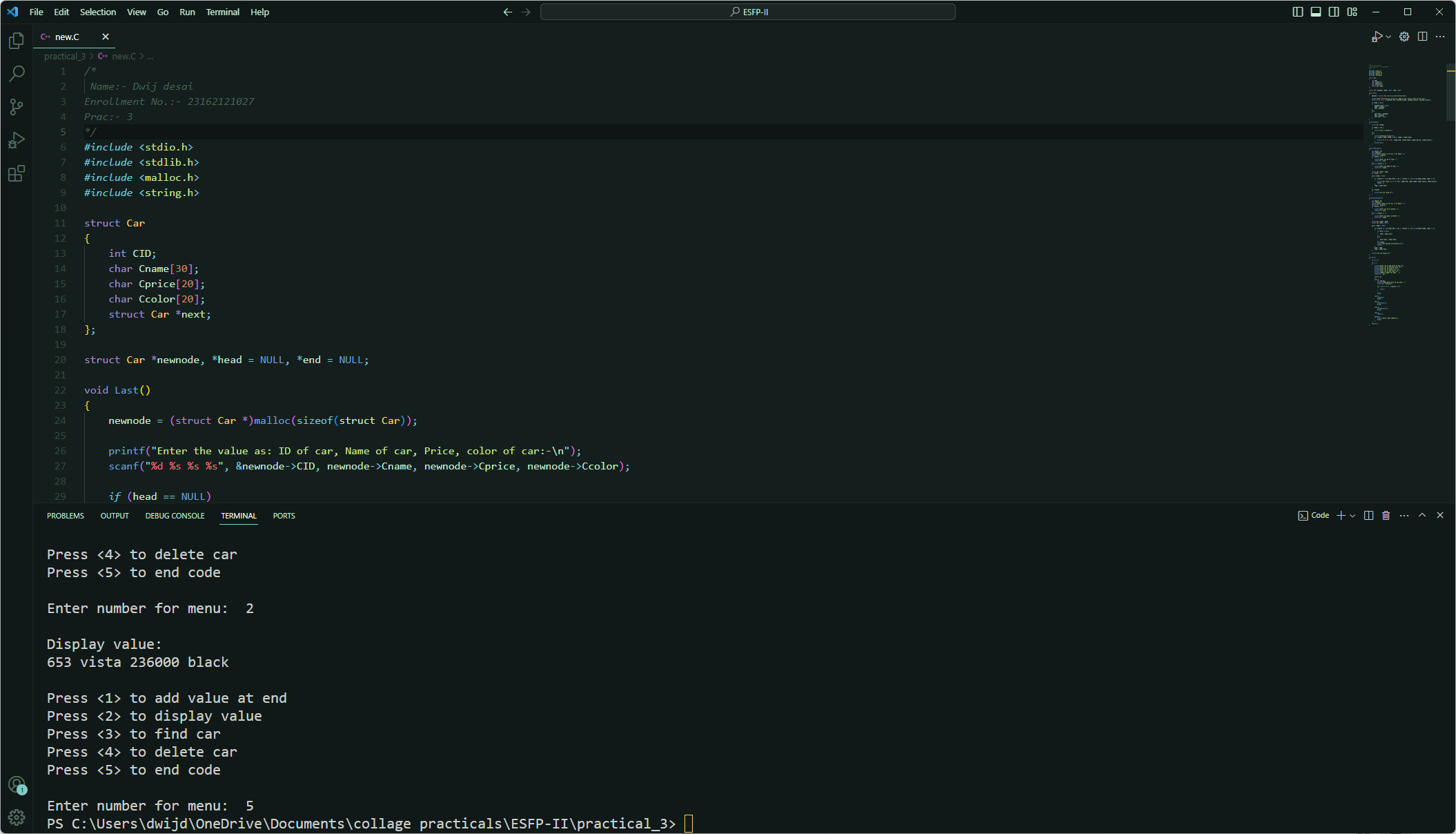
***Output-***







******

******

***Photo of code:-***

******