

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-

```
PS C:\Users\dwijd\OneDrive\Documents\collage practicals\ESFP-II> cd "c:\Users\dwijd\OneDr
new } ; if ($?) { .\new }
Press <1> to add value at end
Press <2> to display value
Press <3> to find car
Press <4> to delete car
Press <5> to end code
```

Enter number for menu: 1

How many units do you want: 2
Enter the value as: ID of car, Name of car, Price, color of car:-
123 nano 340000 green

Enter the value as: ID of car, Name of car, Price, color of car:-
653 vista 236000 black

Press <1> to add value at end
Press <2> to display value
Press <3> to find car
Press <4> to delete car
Press <5> to end code

Enter number for menu: 2

Display value:
123 nano 340000 green
653 vista 236000 black

Press <1> to add value at end
Press <2> to display value
Press <3> to find car
Press <4> to delete car

Press <5> to end code

Enter number for menu: 3
Enter choice (1 for ID, 2 for Name): 1
Enter Car ID to find: 123
Car found: 123 nano 340000 green
Press <1> to add value at end
Press <2> to display value
Press <3> to find car
Press <4> to delete car
Press <5> to end code

Enter number for menu: 4
Enter choice (1 for ID, 2 for Name): 2
Enter Car Name to delete: nano
Car deleted successfully.

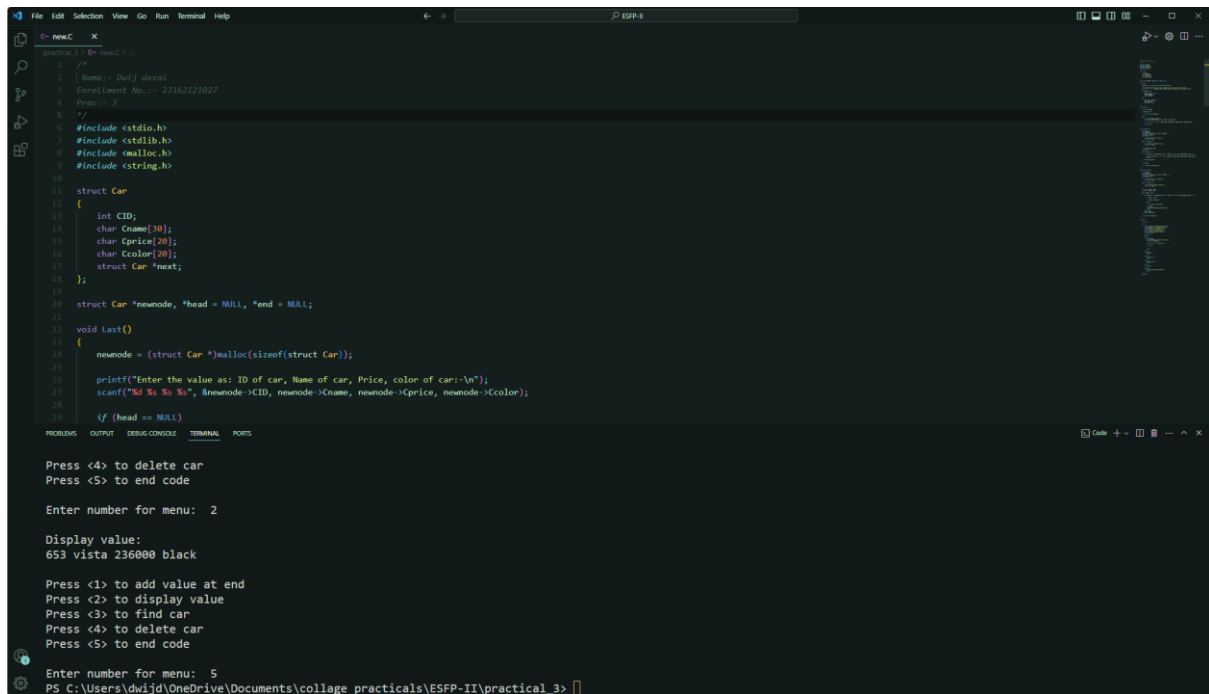
```
Press <1> to add value at end
Press <2> to display value
Press <3> to find car
Press <4> to delete car
Press <5> to end code

Enter number for menu: 2

Display value:
653 vista 236000 black

Press <1> to add value at end
Press <2> to display value
Press <3> to find car
Press <4> to delete car
Press <5> to end code

Enter number for menu: 5
PS C:\Users\dwijd\OneDrive\Documents\collage practicals\ESFP-II\practical_3>
```



The screenshot shows a Visual Studio Code editor with a C program for a linked list. The code defines a `Car` struct with fields for `CID`, `Name`, `Price`, `Color`, and a `next` pointer. It includes functions for adding, displaying, finding, and deleting cars. The terminal output shows the program running, with the user entering menu options 2 and 5, and the program displaying the value 653 vista 236000 black.

```
1 //
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <malloc.h>
5 #include <string.h>
6
7 struct Car
8 {
9     int CID;
10    char Name[30];
11    char Price[20];
12    char Color[20];
13    struct Car *next;
14 };
15
16 struct Car *newnode, *head = NULL, *end = NULL;
17
18 void Last()
19 {
20     newnode = (struct Car *)malloc(sizeof(struct Car));
21
22     printf("Enter the value as: ID of car, Name of car, Price, color of car:-\n");
23     scanf("%d %s %s %s", &newnode->CID, newnode->Name, newnode->Price, newnode->Color);
24
25     if (head == NULL)
26     {
27         head = newnode;
28         end = newnode;
29     }
30     else
31     {
32         end->next = newnode;
33         end = newnode;
34     }
35 }
36
37 int main()
38 {
39     int choice;
40     do
41     {
42         printf("Press <1> to add value at end\n");
43         printf("Press <2> to display value\n");
44         printf("Press <3> to find car\n");
45         printf("Press <4> to delete car\n");
46         printf("Press <5> to end code\n");
47
48         printf("Enter number for menu: ");
49         scanf("%d", &choice);
50
51         switch (choice)
52         {
53             case 1: Last(); break;
54             case 2: display(); break;
55             case 3: find(); break;
56             case 4: delete(); break;
57             case 5: exit(0); break;
58             default: printf("Invalid choice\n"); break;
59         }
60     } while (choice != 5);
61
62     return 0;
63 }
```

Press <4> to delete car
Press <5> to end code

Enter number for menu: 2

Display value:
653 vista 236000 black

Press <1> to add value at end
Press <2> to display value
Press <3> to find car
Press <4> to delete car
Press <5> to end code

Enter number for menu: 5
PS C:\Users\dwijd\OneDrive\Documents\collage practicals\ESFP-II\practical_3>

Photo of code:-


```

1  /*
2   Name:- Divij Desai
3   Enrollment No:- 23162121027
4   Reg:- 4
5   */
6   #include <stdio.h>
7   #include <stdlib.h>
8   #include <malloc.h>
9   #include <string.h>
10
11   struct Car
12   {
13       int CID;
14       char Cname[30];
15       char Cprice[20];
16       char Ccolor[20];
17       struct Car *next;
18   };
19
20   struct Car *newnode, *head = NULL, *end = NULL;
21
22   void Last()
23   {
24       newnode = (struct Car *)malloc(sizeof(struct Car));
25
26       printf("Enter the value as: ID of car, Name of car, Price, color of car:-\n");
27       scanf("%d %s %s %s", &newnode->CID, newnode->Cname, newnode->Cprice, newnode->Ccolor);
28
29       if (head == NULL)
30       {
31           newnode->next = NULL;
32           head = newnode;
33           end = newnode;
34       }
35       else
36       {
37           end->next = newnode;
38           end = newnode;
39           end->next = NULL;
40       }
41   }
42
43   void display()
44   {
45       struct Car *ttemp;
46
47       if (head == NULL)
48       {
49           printf("List is empty\n");
50       }
51       else
52       {
53           printf("\nDisplay value:\n");
54           for (ttemp = head; ttemp != NULL; ttemp = ttemp->next)
55           {
56               printf("%d %s %s %s\n", ttemp->CID, ttemp->Cname, ttemp->Cprice, ttemp->Ccolor);
57           }
58           printf("\n");
59       }
60   }
61
62   void findProduct()
63   {
64       int choice, id;
65       char name[30];
66       printf("Enter choice (1 for ID, 2 for Name): ");
67       scanf("%d", &choice);
68       if (choice == 1)
69       {
70           printf("Enter Car ID to find: ");
71           scanf("%d", &id);
72       }
73       else if (choice == 2)
74       {
75           printf("Enter Car Name to find: ");
76           scanf("%s", name);
77       }
78
79       struct Car *temp = head;
80       int found = 0;
81
82       while (temp != NULL)
83       {
84           if ((choice == 1 && temp->CID == id) || (choice == 2 && strcmp(temp->Cname, name) == 0))
85           {
86               printf("Car found: %d %s %s %s\n", temp->CID, temp->Cname, temp->Cprice, temp->Ccolor);
87               found = 1;
88           }
89           temp = temp->next;
90       }
91
92       if (!found)
93       {
94           printf("Car not found.\n");
95       }
96   }
97
98   void deleteProduct()
99   {
100       int choice, id;
101       char name[30];
102       printf("Enter choice (1 for ID, 2 for Name): ");
103       scanf("%d", &choice);
104       if (choice == 1)
105       {
106           printf("Enter Car ID to delete: ");
107           scanf("%d", &id);
108       }
109       else if (choice == 2)
110       {
111           printf("Enter Car Name to delete: ");
112           scanf("%s", name);
113       }
114
115       struct Car *temp = head;
116       struct Car *prev = NULL;
117
118       while (temp != NULL)
119       {
120           if ((choice == 1 && temp->CID == id) || (choice == 2 && strcmp(temp->Cname, name) == 0))
121           {
122               if (prev == NULL)
123               {
124                   head = temp->next;
125               }
126               else
127               {
128                   prev->next = temp->next;
129               }
130               free(temp);
131               printf("Car deleted successfully.\n");
132               return;
133           }
134           prev = temp;
135           temp = temp->next;
136       }
137
138       printf("Car not found.\n");
139   }
140
141   int main()
142   {
143       int a = 0;
144
145       for (;;)
146       {
147           printf("Press <1> to add value at end \n");
148           printf("Press <2> to display value \n");
149           printf("Press <3> to find car \n");
150           printf("Press <4> to delete car \n");
151           printf("Press <5> to end code \n");
152           printf("\nEnter number for menu: ");
153           scanf("%d", &a);
154
155           switch (a)
156           {
157               case 1:
158                   int num_Car;
159                   printf("\nHow many units do you want: ");
160                   scanf("%d", &num_Car);
161
162                   for (int i = 0; i < num_Car; i++)
163                   {
164                       Last();
165                   }
166
167                   break;
168
169               case 2:
170                   display();
171                   break;
172
173               case 3:
174                   findProduct();
175                   break;
176
177               case 4:
178                   deleteProduct();
179                   break;
180
181               case 5:
182                   return 0;
183
184               default:
185                   printf("Enter right number\n");
186                   break;
187           }
188       }
189       return 0;
190   }
191

```