



CSE 101 COMPUTER PROGRAMMING

FINAL REPORT

ON

ICE CREAM PARLOR MANAGEMENT SYSTEM

Section: K22BH

Under the guidance of

JASPREET KAUR MAM : 27403

Group Members Details:

S No.	Roll No.	Name	Reg No.
1.	37	Parvati	12215266
2.	38	Nishant Kumar	12214304
3.	39	Abhishek Kumar	12215014
4.	40	Vishal Singh	12215059

Discipline of CSE/IT

Lovely School of Computer Science & Engineering

Lovely Professional University, Phagwara

CONTENTS

No.	Title	
1	Project Description	
2	Programming Code	
3	Module Explanation 1 st module 2 nd module 3 rd module 4 th module 5 th module	
4	Snapshot of Code	
5	Output of the Code	
6	DFD (level 0)	
7	Conclusion	

PROJECT DESCRIPTION

Ice-Cream Parlor Management system

An ice cream parlor management system using C code with modules for adding, deleting, searching, updating, and displaying information would be a software application designed to manage the operations of an ice cream parlor.

Each module would perform a specific function:

1. DISPLAY ICE CREAM LIST
2. ADD NEW ICE CREAM
3. UPDATE THE RECORD OF THE ICE CREAM
4. SEARCH ANY ICE CREAM
5. DELETE ANY ICE CREAM RECORD

- The add module would allow the user to input information for new ice cream flavours, toppings, or ingredients into the system.
- The delete module would allow the user to remove any unwanted or expired items from the inventory.
- The search module would allow the user to search for specific items in the inventory, such as a particular flavour of ice cream.
- The update module would allow the user to modify the information for an existing item in the inventory, such as changing the price or updating the stock level.
- The display module would allow the user to view the current inventory, including the available flavour's, toppings, and ingredients, along with their respective prices and stock levels.

Together, these modules would provide a comprehensive solution for managing an ice cream parlor's inventory and operations efficiently. Additionally, the C code could be used to integrate other features, such as employee management, reporting, and analytics, to further optimize the parlor's performance.

C PROGRAMMING CODE

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct ice_cream {
```

```
    int id;
```

```
    char name[50];
```

```
    int price;
```

```
    int stock;
```

```
};
```

```
void display_menu();
```

```
void add_ice_cream();
```

```
void update_ice_cream();
```

```
void search_ice_cream();
```

```
void delete_ice_cream();
```

```
int main() {
```

```
    int choice;
```

```
    do {
```

```
        display_menu();
```

```
        scanf("%d", &choice);
```

```
        switch(choice) {
```

```
            case 1:
```

```
                display_ice_creamlist();
```

```
            break;
```

```
        case 2:
            add_ice_cream();

            break;
        case 3:
            update_ice_cream();

            break;
        case 4:
            search_ice_cream();

            break;
        case 5:
            delete_ice_cream();

            break;
        case 6:
            printf("Exiting program...\n");
            exit(0);
            break;

        default:
            printf("Invalid choice. Please try again.\n");
    }
} while(choice != 6);

return 0;
}

void display_menu() {
    printf("\n\n Ice Cream Management System\n");
```

```

printf("1. List of ice cream\n");
printf("2. Add ice cream\n");
printf("3. Update ice cream\n");
printf("4. Search ice cream\n");
printf("5. Delete ice cream\n");
printf("6. Exit\n");
printf("Enter your choice: ");
}

void display_ice_creamlist(){

    FILE *fp;
    struct ice_cream ic;
    fp=fopen("ice_creams.dat","r");

    printf("\t\t\t\t\t===== ICE CREAM RECORD =====\n\n\n");

    if(fp==NULL){

        fprintf(stderr,"can't open file\n");
        exit(0);
    }else{
        printf("\t\t\t\t\tRECORDS :\n");
        printf("\t\t\t\t\t_____ \n\n");
    }

    while(fread(&ic, sizeof(ic), 1, fp)){
        printf("\n\t\t\t\t\t ID : %d",ic.id);
        printf("\n\t\t\t\t\t Name : %s",ic.name);
        printf("\n\t\t\t\t\t Price : %d",ic.price);
        printf("\n\t\t\t\t\t Stock : %d",ic.stock);
    }
}

```

```
        printf("\n\t\t\t\t\t_____ \n");

    }

    fclose(fp);

    getch();

}


void add_ice_cream() {
    struct ice_cream ic;
    FILE *fp;
    fp = fopen("ice_creams.dat", "ab");

    printf("\nEnter ice cream details:\n");
    printf("ID: ");
    scanf("%d", &ic.id);
    printf("Name: ");
    scanf("%s", &ic.name);
    printf("Price: ");
    scanf("%d", &ic.price);
    printf("Stock: ");
    scanf("%d", &ic.stock);

    fwrite(&ic, sizeof(ic), 1, fp);
    fclose(fp);

    printf("Ice cream added successfully.\n");
}
```

```
void update_ice_cream() {  
    int id;  
    struct ice_cream ic;  
    FILE *fp;  
    fp = fopen("ice_creams.dat", "rb+");  
  
    printf("\nEnter ID of ice cream to update: ");  
    scanf("%d", &id);  
  
    while(fread(&ic, sizeof(ic), 1, fp)) {  
        if(ic.id == id) {  
            printf("\nEnter new details for ice cream:\n");  
            printf("ID: ");  
            scanf("%d", &ic.id);  
            printf("Name: ");  
            scanf("%s", &ic.name);  
            printf("Price: ");  
            scanf("%d", &ic.price);  
            printf("Stock: ");  
            scanf("%d", &ic.stock);  
  
            fseek(fp, -(long)sizeof(ic), SEEK_CUR);  
            fwrite(&ic, sizeof(ic), 1, fp);  
            fclose(fp);  
  
            printf("Ice cream updated successfully.\n");  
            return;  
        }  
    }  
}
```

fclose(fp);


```

    printf("Ice cream not found.\n");
}

void search_ice_cream() {
    int id;
    struct ice_cream ic;
    FILE *fp;
    fp = fopen("ice_creams.dat", "r");

    printf("\nEnter ID of ice cream to search: ");
    scanf("%d", &id);

    while(fread(&ic, sizeof(ic), 1, fp)) {
        if(ic.id == id) {
            printf("\nID: %d\n", ic.id);
            printf("Name: %s\n", ic.name);
            printf("Price: %d\n", ic.price);
        }
    }
    fclose(fp);
}

void delete_ice_cream() {
    int id, count=0;
    struct ice_cream ic;
    FILE *fp, *temp_fp;
    fp = fopen("ice_creams.dat", "r");
    temp_fp = fopen("temp.dat", "w");

    if (fp == NULL) {
        printf("Error opening file.\n");
    }

```

```

        return;
    }

    printf("Enter ID of ice cream to delete: ");
    scanf("%d", &id);

    while (fread(&ic, sizeof(ic), 1, fp)) {
        if (ic.id == id) {
            count=1; // skip the record we want to delete
        }
        else{
            fwrite(&ic, sizeof(ic), 1, temp_fp);
        }
    }

    fclose(fp);
    fclose(temp_fp);
    if(!count){
        printf("\nrecord not found");
    }
    if(count){

        remove("ice_creams.dat");
        rename("temp.dat", "ice_creams.dat");
        printf("Ice cream record deleted successfully.\n");
    }
}

```

MODULE EXPLANATION

MODULE 1 : DISPLAY ICE CREAM LIST

The C code function called "display_ice_creamlist" that reads a binary file named "ice_creams.dat" and displays the contents of the file on the console.

The function first declares a file pointer variable "fp" and an ice_cream structure variable "ic". It then opens the file "ice_creams.dat" in read mode using the "fopen()" function.

The function checks whether the file was opened successfully or not. If the file could not be opened, the function prints an error message on the console and terminates the program using the "exit()" function. If the file was opened successfully, the function displays a header message on the console.

The function then reads the binary file using the "fread()" function and stores the contents in the "ic" variable. It then displays the contents of the "ic" variable on the console using the "printf()" function. This process is repeated until the end of the file is reached.

Finally, the function closes the file using the "fclose()" function and waits for a key press using the "getch()" function.

Note: The implementation of the "ice_cream" structure is not provided in the code snippet, so it is unclear what data the structure contains

MODULE 2 : ADD NEW ICE CREAM

The C CODE function called "add_ice_cream" that allows the user to add a new ice cream record to a binary file named "ice_creams.dat".

The function first declares an ice_cream structure variable "ic" and a file pointer variable "fp". It then opens the file "ice_creams.dat" in append binary mode using the "fopen()" function and assigns the file pointer to the "fp" variable.

The function then prompts the user to enter the details of the new ice cream record, such as ID, name, price, and stock, using the "printf()" and "scanf()" functions.

After the user enters the details, the function writes the contents of the "ic" variable to the file using the "fwrite()" function. The "fwrite()" function writes the binary representation of the "ic" variable to the file.

Finally, the function closes the file using the "fclose()" function.

Note: The implementation of the "ice_cream" structure is not provided in the code snippet, so it is unclear what data the structure contains.

MODULE 3 : UPDATE THE RECORD OF THE ICE CREAM

This is a function written in C that updates the details of an ice cream in a binary file called "ice_creams.dat". The function takes no arguments and returns no value. Here's a breakdown of how it works:

1. It declares a variable `id` to store the ID of the ice cream to update, and a struct `ice_cream` to store the ice cream's details.
2. It opens the file "ice_creams.dat" in binary read-write mode ("`rb+`") using `fopen()` and stores the file pointer in `fp`.
3. It prompts the user to enter the ID of the ice cream to update using `printf()` and `scanf()`.
4. It reads each ice cream record from the file using `fread()` and stores it in the `ic` struct until the end of the file is reached.
5. For each record, it checks if the ID of the current ice cream matches the ID entered by the user.
6. If a match is found, it prompts the user to enter the new details of the ice cream using `printf()` and `scanf()`.
7. It uses `fseek()` to move the file pointer back to the beginning of the current record, and then uses `fwrite()` to overwrite the old record with the new one.
8. It closes the file using `fclose()`, prints a success message using `printf()`, and returns from the function.
9. If no match is found, it closes the file using `fclose()` and prints a failure message using `printf()`.

MODULE 4 : SEARCH ICE CREAM

This is a function written in C that searches for an ice cream in a binary file called "ice_creams.dat" and prints its details. The function takes no arguments and returns no value. Here's a breakdown of how it works:

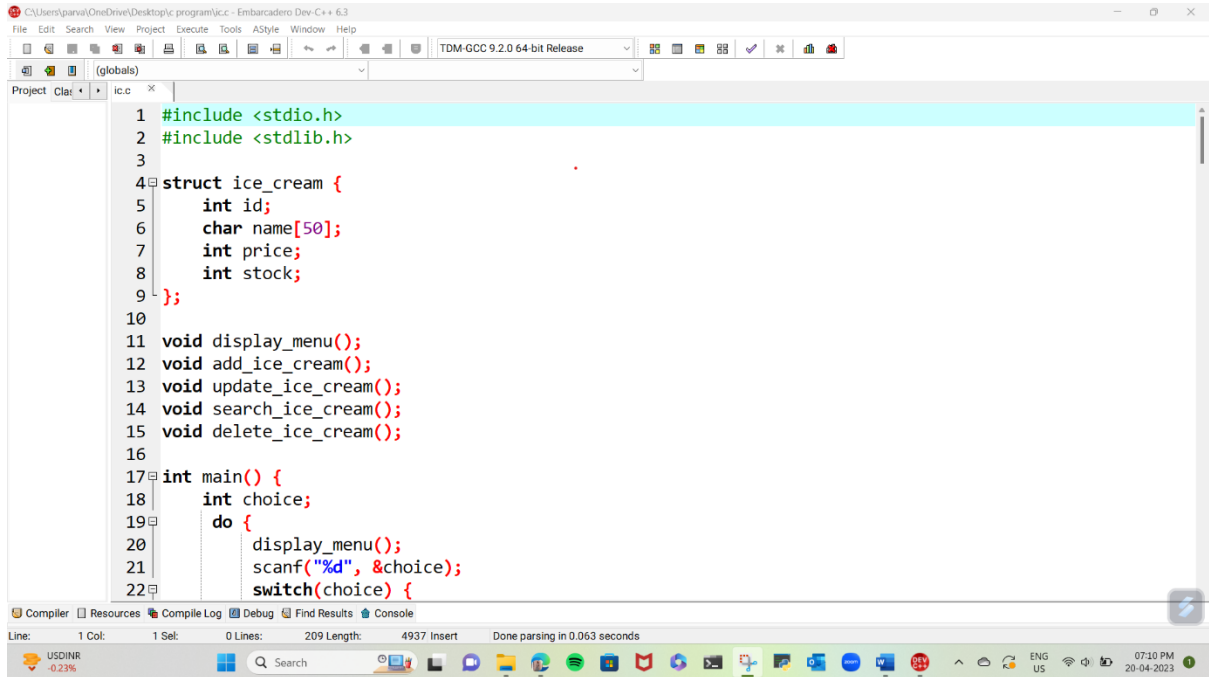
1. It declares a variable `id` to store the ID of the ice cream to search for, and a struct `ice_cream` to store the ice cream's details.
2. It opens the file "ice_creams.dat" in binary read mode ("`r`") using `fopen()` and stores the file pointer in `fp`.
3. It prompts the user to enter the ID of the ice cream to search for using `printf()` and `scanf()`.
4. It reads each ice cream record from the file using `fread()` and stores it in the `ic` struct until the end of the file is reached.
5. For each record, it checks if the ID of the current ice cream matches the ID entered by the user.
6. If a match is found, it prints the details of the ice cream using `printf()`.
7. It continues to read through the rest of the file until the end is reached.
8. It closes the file using `fclose()`.

MODULE 5 : DELETE ICE CREAM RECORD

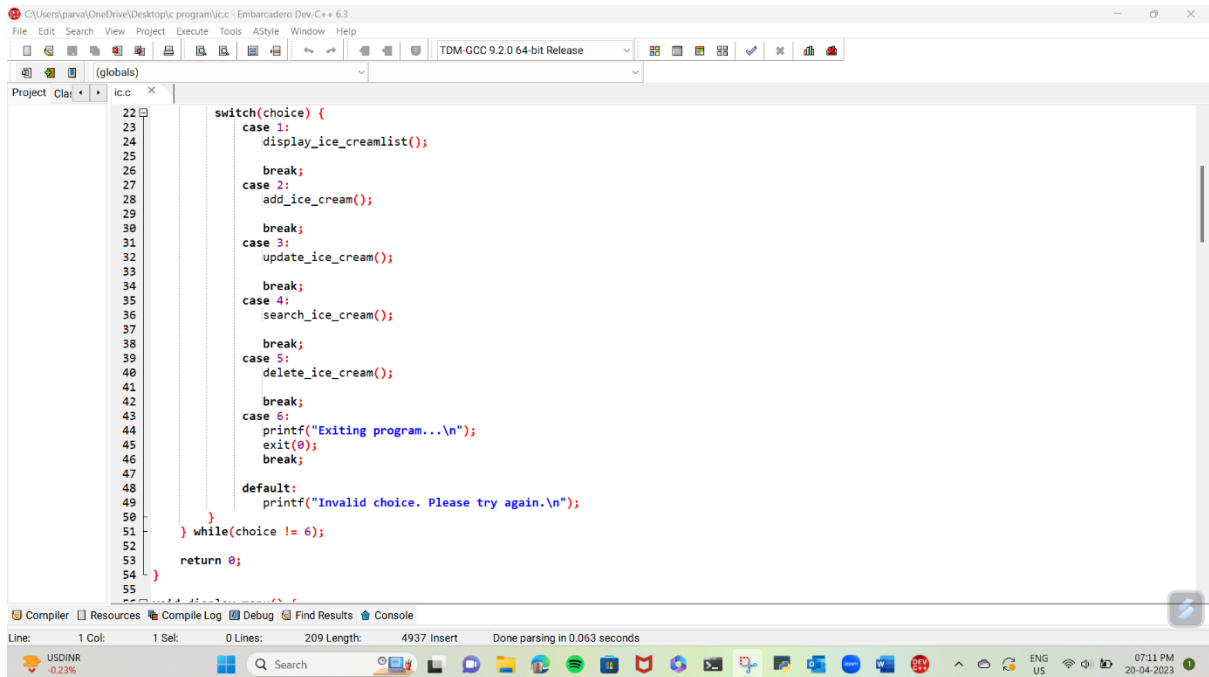
This is a C program that deletes a record of an ice cream from a file named "ice_creams.dat". It works by creating a temporary file called "temp.dat", reading each record from the original file and writing all the records except the one to be deleted to the temporary file. Then, it removes the original file and renames the temporary file to the original filename.

Note: This code assumes that the `struct ice_cream` has a member named `id` that contains the unique identifier for each ice cream record. It also assumes that the file "ice_creams.dat" exists and contains ice cream records in binary format.

SNAPSHOTS OF CODE



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct ice_cream {
5     int id;
6     char name[50];
7     int price;
8     int stock;
9 };
10
11 void display_menu();
12 void add_ice_cream();
13 void update_ice_cream();
14 void search_ice_cream();
15 void delete_ice_cream();
16
17 int main() {
18     int choice;
19     do {
20         display_menu();
21         scanf("%d", &choice);
22         switch(choice) {
```



```
22         switch(choice) {
23             case 1:
24                 display_ice_creamlist();
25                 break;
26             case 2:
27                 add_ice_cream();
28                 break;
29             case 3:
30                 update_ice_cream();
31                 break;
32             case 4:
33                 search_ice_cream();
34                 break;
35             case 5:
36                 delete_ice_cream();
37                 break;
38             case 6:
39                 printf("Exiting program...\n");
40                 exit(0);
41                 break;
42             default:
43                 printf("Invalid choice. Please try again.\n");
44         }
45     } while(choice != 6);
46     return 0;
47 }
```

This screenshot shows a C++ IDE window with the file 'ic.c' open. The code defines a function 'display_menu()' which prints a menu for an 'Ice Cream Management System'. The menu options are: 1. List of ice cream, 2. Add ice cream, 3. Update ice cream, 4. Search ice cream, 5. Delete ice cream, and 6. Exit. It also prompts the user to 'Enter your choice: '.

```
55
56 void display_menu() {
57     printf("Ice Cream Management System\n");
58     printf("\n");
59     printf("1. List of ice cream\n");
60     printf("2. Add ice cream\n");
61     printf("3. Update ice cream\n");
62     printf("4. Search ice cream\n");
63     printf("5. Delete ice cream\n");
64     printf("6. Exit\n");
65     printf("Enter your choice: ");
66 }
67
68 void display_ice_creamlist(){
```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help), a toolbar, and a status bar at the bottom showing 'Line: 54 Col: 2 Sel: 0 Lines: 209 Length: 4937 Insert Done parsing in 0.063 seconds'. The Windows taskbar at the very bottom shows the date and time as 07:12 PM on 20-04-2023.

This screenshot shows the same C++ IDE window, now displaying the 'display_ice_creamlist()' function. This function opens a file named 'ice_creams.dat' in read mode. It checks if the file was opened successfully. If not, it prints an error message and exits. If successful, it prints a header 'ICE CREAM RECORD' and then enters a loop to read and display each record. Each record contains the ID, Name, Price, and Stock of an ice cream.

```
67
68 void display_ice_creamlist(){
69
70     FILE *fp;
71     struct ice_cream ic;
72     fp=fopen("ice_creams.dat","r");
73
74     printf("\t\t\t\t\t===== ICE CREAM RECORD =====\n\n");
75
76     if(fp==NULL){
77
78         fprintf(stderr,"can't open file\n");
79         exit(0);
80     }else{
81         printf("\t\t\t\t\tRECORDS :\n");
82         printf("\t\t\t\t\t_____\n\n");
83     }
84
85     while(fread(&ic, sizeof(ic), 1, fp)){
86         printf("\n\t\t\t\t\tID : %d",ic.id);
87         printf("\n\t\t\t\t\tName : %s",ic.name);
88         printf("\n\t\t\t\t\tPrice : %d",ic.price);
89         printf("\n\t\t\t\t\tStock : %d",ic.stock);
90
91         printf("\n\t\t\t\t\t_____\n");
92     }
93     fclose(fp);
94     getch();
95 }
```

The IDE interface is consistent with the first screenshot, showing the same menu bar, toolbar, and status bar. The Windows taskbar at the bottom shows the date and time as 07:13 PM on 20-04-2023.


```
C:\Users\parva\OneDrive\Desktop\program\icc - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Class icc
97 }
98
99
100 void add_ice_cream() {
101     struct ice_cream ic;
102     FILE *fp;
103     fp = fopen("ice_creams.dat", "ab");
104
105     printf("\nEnter ice cream details:\n");
106     printf("ID: ");
107     scanf("%d", &ic.id);
108     printf("Name: ");
109     scanf("%s", &ic.name);
110     printf("Price: ");
111     scanf("%d", &ic.price);
112     printf("Stock: ");
113     scanf("%d", &ic.stock);
114
115     fwrite(&ic, sizeof(ic), 1, fp);
116     fclose(fp);
117
118     printf("Ice cream added successfully.\n");
119 }
120

Compiler Resources Compile Log Debug Find Results Console
Line: 54 Col: 2 Sel: 0 Lines: 209 Length: 4937 Insert Done parsing in 0.063 seconds
USDINR -0.23% 07:13 PM 20-04-2023
```

```
C:\Users\parva\OneDrive\Desktop\program\icc - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Class icc
121 void update_ice_cream() {
122     int id;
123     struct ice_cream ic;
124     FILE *fp;
125     fp = fopen("ice_creams.dat", "rb+");
126
127     printf("\nEnter ID of ice cream to update: ");
128     scanf("%d", &id);
129
130     while(fread(&ic, sizeof(ic), 1, fp)) {
131         if(ic.id == id) {
132             printf("\nEnter new details for ice cream:\n");
133             printf("ID: ");
134             scanf("%d", &ic.id);
135             printf("Name: ");
136             scanf("%s", &ic.name);
137             printf("Price: ");
138             scanf("%d", &ic.price);
139             printf("Stock: ");
140             scanf("%d", &ic.stock);
141
142             fseek(fp, -(long)sizeof(ic), SEEK_CUR);
143             fwrite(&ic, sizeof(ic), 1, fp);
144             fclose(fp);
145
146             printf("Ice cream updated successfully.\n");
147             return;
148         }
149     }
150 }
```

```
C:\Users\parval\OneDrive\Desktop\c program\icc - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Clas
icc
154
155 void search_ice_cream() {
156     int id;
157     struct ice_cream ic;
158     FILE *fp;
159     fp = fopen("ice_creams.dat", "r");
160
161     printf("\nEnter ID of ice cream to search: ");
162     scanf("%d", &id);
163
164     while(fread(&ic, sizeof(ic), 1, fp)) {
165         if(ic.id == id) {
166             printf("\nID: %d\n", ic.id);
167             printf("Name: %s\n", ic.name);
168             printf("Price: %d\n", ic.price);
169         }
170     }
171     fclose(fp);
172 }
173
174 void delete_ice_cream() {
```

```
C:\Users\parval\OneDrive\Desktop\c program\icc - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Clas
icc
173
174 void delete_ice_cream() {
175     int id, count=0;
176     struct ice_cream ic;
177     FILE *fp, *temp_fp;
178     fp = fopen("ice_creams.dat", "r");
179     temp_fp = fopen("temp.dat", "w");
180
181     if (fp == NULL) {
182         printf("Error opening file.\n");
183         return;
184     }
185
186     printf("Enter ID of ice cream to delete: ");
187     scanf("%d", &id);
188
189     while (fread(&ic, sizeof(ic), 1, fp)) {
190         if (ic.id == id) {
191             count++; // skip the record we want to delete
192         }
193         else {
194             fwrite(&ic, sizeof(ic), 1, temp_fp);
195         }
196     }
197
198     fclose(fp);
199     fclose(temp_fp);
200     if(!count){
201         printf("\nrecord not found");
202     }
203     if(count){
204
205         remove("ice_creams.dat");
206         rename("temp.dat", "ice_creams.dat");
207         printf("Ice cream record deleted successfully.\n");
208     }
}
```

OUTPUT OF CODE

```
C:\Users\parva\OneDrive\Des  x  +  v
Ice Cream Management System

1. List of ice cream
2. Add ice cream
3. Update ice cream
4. Search ice cream
5. Delete ice cream
6. Exit
Enter your choice: 1

===== ICE CREAM RECORD =====

RECORDS :
-----

ID : 1
Name : strawberry
Price : 25
Stock : 30
-----

ID : 2
Name : chocolate
Price : 30
Stock : 40
-----

ID : 3
Name : vanilla
Price : 35
Stock : 40
-----

ID : 4
Name : Butterscotch
Price : 50
Stock : 40
-----

USDINR -0.23%  Search  ENG US  07:28 PM 20-04-2023
```

```
C:\Users\parva\OneDrive\Des  x  +  v

Price : 30
Stock : 40
-----

ID : 3
Name : vanilla
Price : 35
Stock : 40
-----

ID : 4
Name : Butterscotch
Price : 50
Stock : 40
-----

Ice Cream Management System

1. List of ice cream
2. Add ice cream
3. Update ice cream
4. Search ice cream
5. Delete ice cream
6. Exit
Enter your choice: 2

Enter ice cream details:
ID: 5
Name: mint
Price: 40
Stock: 25
Ice cream added successfully.
Ice Cream Management System

1. List of ice cream
2. Add ice cream
3. Update ice cream
4. Search ice cream
5. Delete ice cream
6. Exit
Enter your choice:

USDINR -0.23%  Search  ENG US  07:30 PM 20-04-2023
```

```
C:\Users\parva\OneDrive\Des x + v
1. List of ice cream
2. Add ice cream
3. Update ice cream
4. Search ice cream
5. Delete ice cream
6. Exit
Enter your choice: 3

Enter ID of ice cream to update: 5

Enter new details for ice cream:
ID: 5
Name: coffee
Price: 40
Stock: 45
Ice cream updated successfully.
Ice Cream Management System

1. List of ice cream
2. Add ice cream
3. Update ice cream
4. Search ice cream
5. Delete ice cream
6. Exit
Enter your choice: 4

Enter ID of ice cream to search: 2

ID: 2
Name: chocolate
Price: 30
Ice Cream Management System

1. List of ice cream
2. Add ice cream
3. Update ice cream
4. Search ice cream
5. Delete ice cream
6. Exit
Enter your choice: |
```

```
C:\Users\parva\OneDrive\Des x + v
Enter your choice: 4

Enter ID of ice cream to search: 2

ID: 2
Name: chocolate
Price: 30
Ice Cream Management System

1. List of ice cream
2. Add ice cream
3. Update ice cream
4. Search ice cream
5. Delete ice cream
6. Exit
Enter your choice: 5
Enter ID of ice cream to delete: 5
Ice cream record deleted successfully.
Ice Cream Management System

1. List of ice cream
2. Add ice cream
3. Update ice cream
4. Search ice cream
5. Delete ice cream
6. Exit
Enter your choice: 1

===== ICE CREAM RECORD =====

RECORDS :
-----

ID : 1
Name : strawberry
Price : 25
Stock : 30
-----
```

```
C:\Users\parva\OneDrive\Des  x  +  v

-----
ID : 1
Name : strawberry
Price : 25
Stock : 30
-----

ID : 2
Name : chocolate
Price : 30
Stock : 40
-----

ID : 3
Name : vanilla
Price : 35
Stock : 40
-----

ID : 4
Name : Butterscotch
Price : 50
Stock : 40
-----

Ice Cream Management System
1. List of ice cream
2. Add ice cream
3. Update ice cream
4. Search ice cream
5. Delete ice cream
6. Exit
Enter your choice: 6
Exiting program...

-----
Process exited after 12.97 seconds with return value 0
Press any key to continue . . .

USDINR -0.23%
Search
ENG US 07:36 PM 20-04-2023
```

```
C:\Users\parva\OneDrive\Des  x  +  v

1. List of ice cream
2. Add ice cream
3. Update ice cream
4. Search ice cream
5. Delete ice cream
6. Exit
Enter your choice: 3

Enter ID of ice cream to update: 1

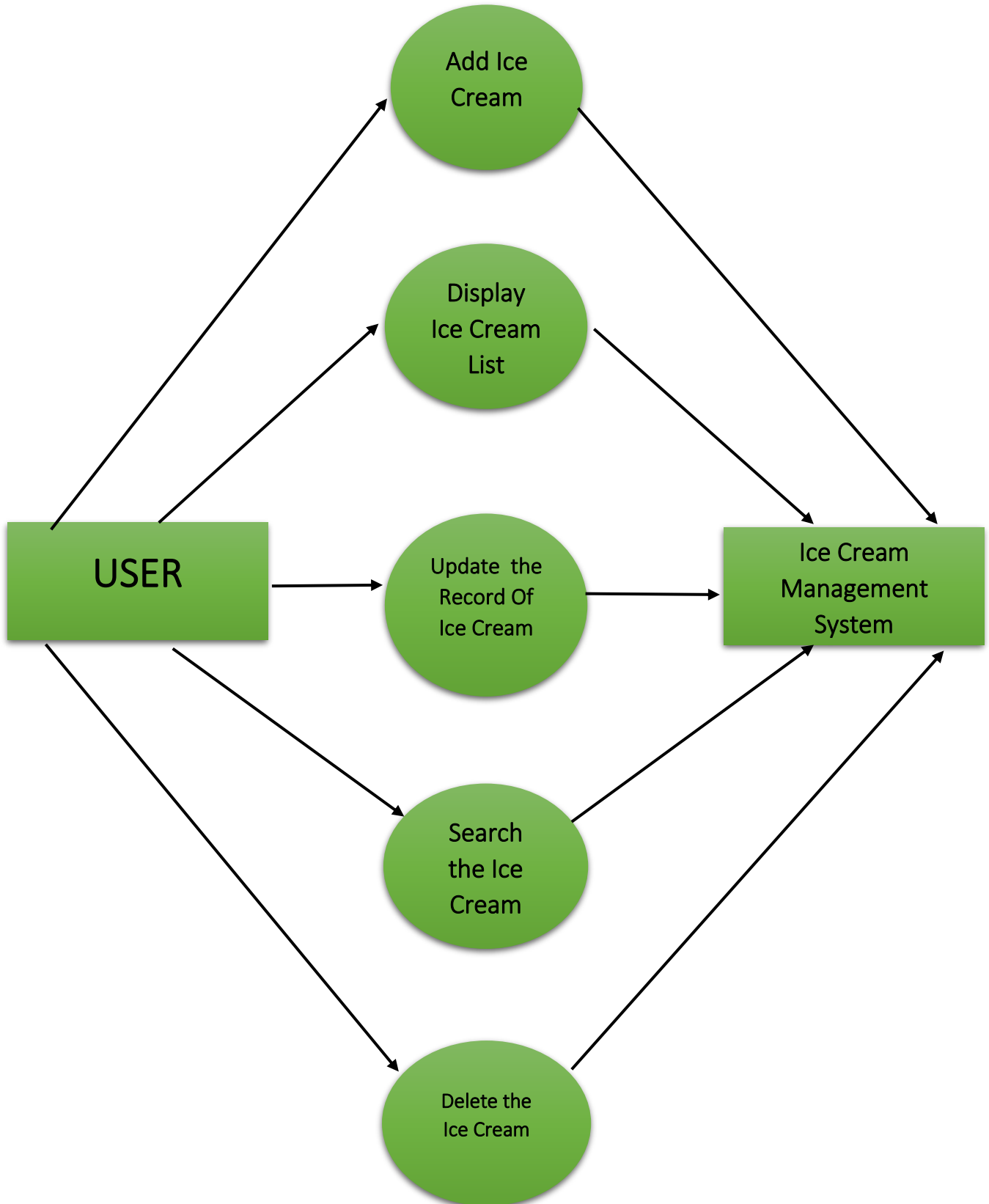
Enter new details for ice cream:
ID: 1
Name: mawakulfi
Price: 20
Stock: 50
Ice cream updated successfully.
Ice Cream Management System

1. List of ice cream
2. Add ice cream
3. Update ice cream
4. Search ice cream
5. Delete ice cream
6. Exit
Enter your choice: 6
Exiting program...

-----
Process exited after 45.13 seconds with return value 0
Press any key to continue . . .

USDINR -0.23%
Search
ENG US 07:38 PM 20-04-2023
```

DFD (0 LEVEL)



CONCLUSION

The program allows the user to add, delete, update, search for, and display ice cream items stored in a binary file. These functions provide basic functionality for managing inventory data, but there may be limitations in terms of the scalability, performance, and security of the system.

Additionally, it is unclear whether this program provides any user interface or other features that would make it usable for real-world applications. Further development and testing would be required to ensure the program is reliable, efficient, and user-friendly.

In summary, while this system may be useful for a small-scale ice cream parlor, it would likely require significant additional development to be used effectively in a larger-scale, real-world environment.