EXP 16: Develop a C program for implementing random access file for processing the employee details

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

#define FILENAME "employee.dat"

typedef struct {
   int id;
   char name[30];
   float salary;
   int isDeleted;  // 0 = active, 1 = deleted
} Employee;

// Function to add employee record
void addEmployee() {
   Employee e;
   FILE *fp = fopen(FILENAME, "ab");
   if (fp == NULL) {
      perror("Error opening file");
      return;
   }

   printf("Enter Employee ID: ");
   scanf("%d", &e.id);
   printf("Enter Name: ");
   scanf("%s", e.name);
```

```
    printf("Enter Salary: ");
    scanf("%f", &e.salary);
    e.isDeleted = 0;

    fwrite(&e, sizeof(Employee), 1, fp);
    fclose(fp);
    printf("Employee added successfully.\n");
}

// Function to display all employee records
void displayEmployees() {
    Employee e;
    FILE *fp = fopen(FILENAME, "rb");
    if (fp == NULL) {
        perror("Error opening file");
        return;
    }

    printf("\nEmployee Records:\n");
    while (fread(&e, sizeof(Employee), 1, fp)) {
        if (!e.isDeleted) {
            printf("ID: %d\tName: %s\tSalary: %.2f\n", e.id, e.name, e.salary);
        }
    }

    fclose(fp);
}
```

```
// Function to search employee by ID
void searchEmployee() {
    int id, found = 0;
    Employee e;
    FILE *fp = fopen(FILENAME, "rb");
    if (fp == NULL) {
        perror("Error opening file");
        return;
    }

    printf("Enter Employee ID to search: ");
    scanf("%d", &id);

    while (fread(&e, sizeof(Employee), 1, fp)) {
        if (e.id == id && !e.isDeleted) {
            printf("Record Found:\nID: %d\tName: %s\tSalary: %.2f\n", e.id, e.name, e.salary);
            found = 1;
            break;
        }
    }

    if (!found) {
        printf("Employee not found.\n");
    }

    fclose(fp);
}
```

```
// Function to update an employee's record
void updateEmployee() {
    int id, found = 0;
    Employee e;
    FILE *fp = fopen(FILENAME, "rb+");
    if (fp == NULL) {
        perror("Error opening file");
        return;
    }

    printf("Enter Employee ID to update: ");
    scanf("%d", &id);

    while (fread(&e, sizeof(Employee), 1, fp)) {
        if (e.id == id && !e.isDeleted) {
            printf("Current Name: %s, Salary: %.2f\n", e.name, e.salary);
            printf("Enter new Name: ");
            scanf("%s", e.name);
            printf("Enter new Salary: ");
            scanf("%f", &e.salary);

            fseek(fp, -sizeof(Employee), SEEK_CUR);
            fwrite(&e, sizeof(Employee), 1, fp);
            found = 1;
            printf("Employee updated successfully.\n");
            break;
        }
    }
```

```
    if (!found) {
        printf("Employee not found.\n");
    }

    fclose(fp);
}

// Function to delete employee record (mark as deleted)
void deleteEmployee() {
    int id, found = 0;
    Employee e;
    FILE *fp = fopen(FILENAME, "rb+");
    if (fp == NULL) {
        perror("Error opening file");
        return;
    }

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

    while (fread(&e, sizeof(Employee), 1, fp)) {
        if (e.id == id && !e.isDeleted) {
            e.isDeleted = 1;
            fseek(fp, -sizeof(Employee), SEEK_CUR);
            fwrite(&e, sizeof(Employee), 1, fp);
            found = 1;
            printf("Employee deleted successfully (logically).\n");
```

```
            break;
        }
    }

    if (!found) {
        printf("Employee not found.\n");
    }

    fclose(fp);
}

// Main menu
int main() {
    int choice;

    do {
        printf("\n==== Employee Management Menu ====\n");
        printf("1. Add Employee\n");
        printf("2. Display All Employees\n");
        printf("3. Search Employee\n");
        printf("4. Update Employee\n");
        printf("5. Delete Employee\n");
        printf("0. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: addEmployee(); break;
```

```
            case 2: displayEmployees(); break;
            case 3: searchEmployee(); break;
            case 4: updateEmployee(); break;
            case 5: deleteEmployee(); break;
            case 0: printf("Exiting program.\n"); break;
            default: printf("Invalid choice. Please try again.\n");
        }

    } while (choice != 0);

    return 0;
}
```

Sample Output

```
==== Employee Management Menu ====
1. Add Employee
2. Display All Employees
3. Search Employee
4. Update Employee
5. Delete Employee
0. Exit
Enter your choice: 3
Error opening file: No such file or directory
```

```
==== Employee Management Menu ====
1. Add Employee
2. Display All Employees
3. Search Employee
4. Update Employee
5. Delete Employee
0. Exit
Enter your choice:0
```

```
==== Employee Management Menu ====
1. Add Employee
2. Display All Employees
3. Search Employee
4. Update Employee
5. Delete Employee
0. Exit
Enter your choice: 2
Error opening file: No such file or directory
```