EMPLOYEE MANAGEMENT

Input:

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


// Structure of the employee

struct emp {

    char name[50];

    float salary;

    int age;

    int id;
};

struct emp e;


// size of the structure

long int size = sizeof(e);


// In the start coordinates
// will be 0, 0
COORD cord = { 0, 0 };


// function to set the
// coordinates

void gotoxy(int x, int y)
{

    cord.X = x;
```

```
    cord.Y = y;
    SetConsoleCursorPosition(
        GetStdHandle(STD_OUTPUT_HANDLE),
        cord);
}

FILE *fp, *ft;
// Function to add the records
void addrecord()
{
    system("cls");
    fseek(fp, 0, SEEK_END);
    char another = 'y';

    while (another == 'y') {
        printf("\nEnter Name : ");
        scanf("%s", e.name);

        printf("\nEnter Age : ");
        scanf("%d", &e.age);

        printf("\nEnter Salary : ");
        scanf("%f", &e.salary);
```

```
        printf("\nEnter EMP-ID : ");
        scanf("%d", &e.id);

        fwrite(&e, size, 1, fp);

        printf("\nWant to add another"
               " record (Y/N) : ");
        fflush(stdin);

        scanf("%c", &another);
    }
}

// Function to delete the records
void deleterecord()
{
    system("cls");
    char empname[50];
    char another = 'y';

    while (another == 'y') {
        printf("\nEnter employee "
               "name to delete : ");
```

```
scanf("%s", empname);

ft = fopen("temp.txt", "wb");
rewind(fp);

while (fread(&e, size,
             1, fp)
       == 1) {
    if (strcmp(e.name,
               empname)
        != 0)
        fwrite(&e, size, 1, ft);
}

fclose(fp);
fclose(ft);
remove("data.txt");
rename("temp.txt", "data.txt");
fp = fopen("data.txt", "rb+");

printf("\nWant to delete another"
       " record (Y/N) :");
```

```
        fflush(stdin);

        another = getche();

    }
}

// Function to display the record

void displayrecord()
{

    system("cls");


    // sets pointer to start

    // of the file

    rewind(fp);


    printf("\n========================="

           "============================"

           "======");

    printf("\nNAME\t\tAGE\t\tSALARY\t\t"

           "\tID\n",

           e.name, e.age,

           e.salary, e.id);

    printf("============================"

           "============================"
```

```
            "====\n");


    while (fread(&e, size, 1, fp) == 1)
        printf("\n%s\t\t%d\t\t%.2f\t%10d",
                e.name, e.age, e.salary, e.id);


    printf("\n\n\n\t");
    system("pause");
}

// Function to modify the record
void modifyrecord()
{
    system("cls");
    char empname[50];
    char another = 'y';


    while (another == 'y') {
        printf("\nEnter employee name"
                " to modify : ");
        scanf("%s", empname);


        rewind(fp);
```

```
// While File is open
while (fread(&e, size, 1, fp) == 1) {
    // Compare the employee name
    // with ename
    if (strcmp(e.name, empname) == 0) {
        printf("\nEnter new name:");
        scanf("%s", e.name);
        printf("\nEnter new age :");
        scanf("%d", &e.age);
        printf("\nEnter new salary :");
        scanf("%f", &e.salary);
        printf("\nEnter new EMP-ID :");
        scanf("%d", &e.id);

        fseek(fp, -size, SEEK_CUR);
        fwrite(&e, size, 1, fp);
        break;
    }
}

// Ask for modifying another record
printf("\nWant to modify another"
```

```
            " record (Y/N) :");

        fflush(stdin);

        scanf("%c", &another);

    }
}


// Driver code

int main()
{

    int choice;



    // opening the file

    fp = fopen("data.txt", "rb+");



    // showing error if file is

    // unable to open.

    if (fp == NULL) {

        fp = fopen("data.txt", "wb+");

        if (fp == NULL) {

            printf("\nCannot open file...");

            exit(1);

        }

    }
```

```
system("Color 3F");
printf("\n\n\n\n\t\t\t\t=============="
       "============================="
       "===========");
printf("\n\t\t\t\t~~~~~~~~~~~~~~~~~~~"
       "~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
       "~~~~~");
printf("\n\t\t\t\t==================="
       "============================="
       "=====");
printf("\n\t\t\t\t[|:::>:::>:::>::>  "
       "EMPLOYEE RECORD  <::<:::<:::"
       "<:::|]\t");
printf("\n\t\t\t\t==================="
       "============================="
       "=====");
printf("\n\t\t\t\t~~~~~~~~~~~~~~~~~~~"
       "~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
       "~~~");
printf("\n\t\t\t\t====================="
       "=============================\n");
```

```
printf("\n\n\n\t\t\t\t\t\t\t\t\t\t\t"
       "Developer : @Sushant_Gaurav"
       "\n\n\t\t\t\t");

system("pause");

while (1) {
    // Clearing console and asking the
    // user for input
    system("cls");
    gotoxy(30, 10);
    printf("\n1. ADD RECORD\n");
    gotoxy(30, 12);
    printf("\n2. DELETE RECORD\n");
    gotoxy(30, 14);
    printf("\n3. DISPLAY RECORDS\n");
    gotoxy(30, 16);
    printf("\n4. MODIFY RECORD\n");
    gotoxy(30, 18);
    printf("\n5. EXIT\n");
    gotoxy(30, 20);
    printf("\nENTER YOUR CHOICE...\n");
```

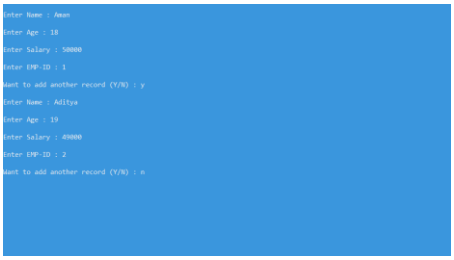
```
fflush(stdin);
scanf("%d", &choice);

// Switch Case
switch (choice) {
case 1:

    // Add the records
    addrecord();
    break;

case 2:

    // Delete the records
    deleterecord();
    break;

case 3:

    // Display the records
    displayrecord();
    break;
```

```
        case 4:


            // Modify the records
            modifyrecord();
            break;


        case 5:
            fclose(fp);
            exit(0);
            break;


        default:
            printf("\nINVALID CHOICE...\n");
        }
    }


    return 0;
}
```

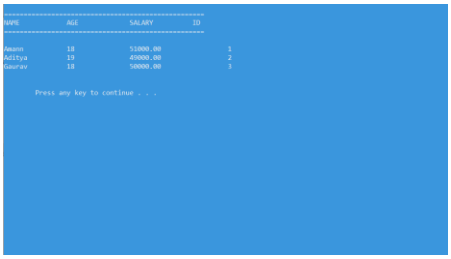
Output:

Displaying options:

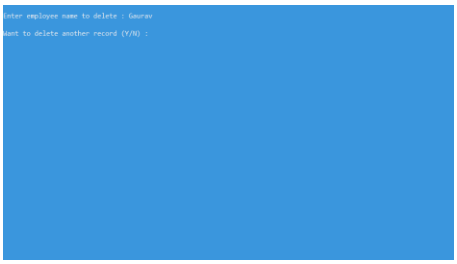
Adding employee record:



Displaying record:

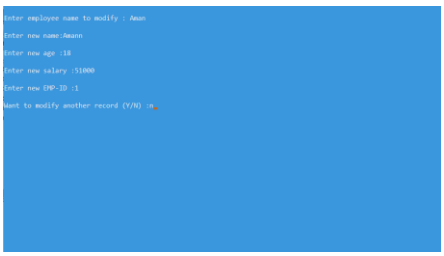
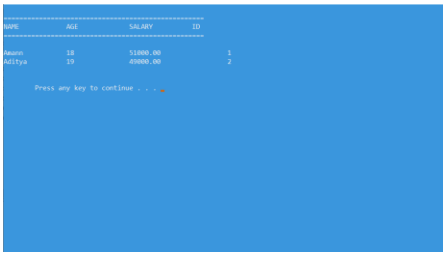


Deleting record:



After deleting a record:

Modifying record:



Record after modification: