#include <stdio.h> ///for input output functions like printf, scanf

#include <stdlib.h>

#include <conio.h>

#include <windows.h> ///for windows related functions (not important)

#include <string.h> ///string operations

/\*\* List of Global Variable \*/

COORD coord = {**0**,**0**}; /// top-left corner of window

/\*\*

function : gotoxy

@param input: x and y coordinates

@param output: moves the cursor in specified position of console

\*/

void gotoxy(int x,int y)

{

coord.X = x;

coord.Y = y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE),coord);

}

/\*\* Main function started \*/

int main()

{

FILE \*fp, \*ft; /// file pointers

char another, choice;

/\*\* structure that represent a employee \*/

struct emp

{

char name[**40**]; ///name of employee

int age; /// age of employee

float bs; /// basic salary of employee

};

struct emp e; /// structure variable creation

char empname[**40**]; /// string to store name of the employee

long int recsize; /// size of each record of employee

/\*\* open the file in binary read and write mode

\* if the file EMP.DAT already exists then it open that file in read write mode

\* if the file doesn't exit it simply create a new copy

\*/

fp = fopen("EMP.DAT","rb+");

if(fp == NULL)

{

fp = fopen("EMP.DAT","wb+");

if(fp == NULL)

{

printf("Connot open file");

exit(**1**);

}

}

/// sizeo of each record i.e. size of structure variable e

recsize = sizeof(e);

/// infinite loop continues untile the break statement encounter

while(**1**)

{

system("cls"); ///clear the console window

gotoxy(**30**,**10**); /// move the cursor to postion 30, 10 from top-left corner

printf("1. Add Record"); /// option for add record

gotoxy(**30**,**12**);

printf("2. List Records"); /// option for showing existing record

gotoxy(**30**,**14**);

printf("3. Modify Records"); /// option for editing record

gotoxy(**30**,**16**);

printf("4. Delete Records"); /// option for deleting record

gotoxy(**30**,**18**);

printf("5. Exit"); /// exit from the program

gotoxy(**30**,**20**);

printf("Your Choice: "); /// enter the choice 1, 2, 3, 4, 5

fflush(stdin); /// flush the input buffer

choice = getche(); /// get the input from keyboard

switch(choice)

{

case '1': /// if user press 1

system("cls");

fseek(fp,**0**,SEEK\_END); /// search the file and move cursor to end of the file

/// here 0 indicates moving 0 distance from the end of the file

another = 'y';

while(another == 'y') /// if user want to add another record

{

printf("\nEnter name: ");

scanf("%s",e.name);

printf("\nEnter age: ");

scanf("%d", &e.age);

printf("\nEnter basic salary: ");

scanf("%f", &e.bs);

fwrite(&e,recsize,**1**,fp); /// write the record in the file

printf("\nAdd another record(y/n) ");

fflush(stdin);

another = getche();

}

break;

case '2':

system("cls");

rewind(fp); ///this moves file cursor to start of the file

while(fread(&e,recsize,**1**,fp)==**1**) /// read the file and fetch the record one record per fetch

{

printf("\n%s %d %.2f",e.name,e.age,e.bs); /// print the name, age and basic salary

}

getch();

break;

case '3': /// if user press 3 then do editing existing record

system("cls");

another = 'y';

while(another == 'y')

{

printf("Enter the employee name to modify: ");

scanf("%s", empname);

rewind(fp);

while(fread(&e,recsize,**1**,fp)==**1**) /// fetch all record from file

{

if(strcmp(e.name,empname) == **0**) ///if entered name matches with that in file

{

printf("\nEnter new name,age and bs: ");

scanf("%s%d%f",e.name,&e.age,&e.bs);

fseek(fp,-recsize,SEEK\_CUR); /// move the cursor 1 step back from current position

fwrite(&e,recsize,**1**,fp); /// override the record

break;

}

}

printf("\nModify another record(y/n)");

fflush(stdin);

another = getche();

}

break;

case '4':

system("cls");

another = 'y';

while(another == 'y')

{

printf("\nEnter name of employee to delete: ");

scanf("%s",empname);

ft = fopen("Temp.dat","wb"); /// create a intermediate file for temporary storage

rewind(fp); /// move record to starting of file

while(fread(&e,recsize,**1**,fp) == **1**) /// read all records from file

{

if(strcmp(e.name,empname) != **0**) /// if the entered record match

{

fwrite(&e,recsize,**1**,ft); /// move all records except the one that is to be deleted to temp file

}

}

fclose(fp);

fclose(ft);

remove("EMP.DAT"); /// remove the orginal file

rename("Temp.dat","EMP.DAT"); /// rename the temp file to original file name

fp = fopen("EMP.DAT", "rb+");

printf("Delete another record(y/n)");

fflush(stdin);

another = getche();

}

break;

case '5':

fclose(fp); /// close the file

exit(**0**); /// exit from the program

}

}

return **0**;

}