**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

LAB 0

**CSH202B-T**

Q1: Create and display 5 records for students using file operations in C.

#include <stdio.h>

#include <stdlib.h>

struct student {

char name[50];

int roll;

float marks;

};

int main() {

int i, n = 5;

struct student s[n];

printf("Enter details of %d students:\n", n);

for (i = 0; i < n; i++) {

printf("Enter name of student %d: ", i+1);

scanf("%s", s[i].name);

printf("Enter roll number of student %d: ", i+1);

scanf("%d", &s[i].roll);

printf("Enter marks of student %d: ", i+1);

scanf("%f", &s[i].marks);

}

FILE \*fp = fopen("students.txt", "wb");

if (fp == NULL) {

printf("Error: Unable to create file\n");

exit(1);

}

fwrite(s, sizeof(struct student), n, fp);

fclose(fp);

fp = fopen("students.txt", "rb");

if (fp == NULL) {

printf("Error: Unable to open file\n");

exit(1);

}

printf("\nStudent Records:\n");

for (i = 0; i < n; i++) {

fread(&s[i], sizeof(struct student), 1, fp);

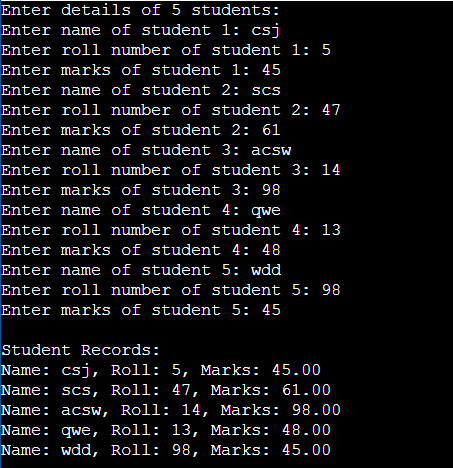
printf("Name: %s, Roll: %d, Marks: %.2f\n", s[i].name, s[i].roll, s[i].marks);

}

fclose(fp);

return 0;

}



Q2: Using file operations create, display, modify insert and delete records for employee system.  
Sol:-

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Employee {

int id;

char name[50];

float salary;

};

void create\_record() {

FILE \*fp;

struct Employee emp;

fp = fopen("employees.txt", "a");

printf("Enter employee ID: ");

scanf("%d", &emp.id);

printf("Enter employee name: ");

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

printf("Enter employee salary: ");

scanf("%f", &emp.salary);

fprintf(fp, "%d %s %.2f\n", emp.id, emp.name, emp.salary);

fclose(fp);

}

void display\_records() {

FILE \*fp;

struct Employee emp;

fp = fopen("employees.txt", "r");

printf("ID\tName\tSalary\n");

printf("----------------------------------------\n");

while(fscanf(fp, "%d %s %f", &emp.id, emp.name, &emp.salary) != EOF) {

printf("%d\t%s\t%.2f\n", emp.id, emp.name, emp.salary);

}

fclose(fp);

}

void modify\_record() {

FILE \*fp, \*fp\_temp;

struct Employee emp;

int id, found = 0;

fp = fopen("employees.txt", "r");

fp\_temp = fopen("temp.txt", "w");

printf("Enter employee ID to modify: ");

scanf("%d", &id);

while(fscanf(fp, "%d %s %f", &emp.id, emp.name, &emp.salary) != EOF) {

if(emp.id == id) {

printf("Enter new name: ");

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

printf("Enter new salary: ");

scanf("%f", &emp.salary);

found = 1;

}

fprintf(fp\_temp, "%d %s %.2f\n", emp.id, emp.name, emp.salary);

}

fclose(fp);

fclose(fp\_temp);

if(found == 0) {

printf("Record not found!\n");

remove("temp.txt");

} else {

remove("employees.txt");

rename("temp.txt", "employees.txt");

printf("Record modified successfully!\n");

}

}

void insert\_record() {

FILE \*fp, \*fp\_temp;

struct Employee emp;

int id, found = 0;

fp = fopen("employees.txt", "r");

fp\_temp = fopen("temp.txt", "w");

printf("Enter employee ID to insert: ");

scanf("%d", &id);

while(fscanf(fp, "%d %s %f", &emp.id, emp.name, &emp.salary) != EOF) {

if(emp.id == id) {

printf("Record already exists!\n");

found = 1;

break;

} else if(emp.id > id) {

fprintf(fp\_temp, "%d %s %.2f\n", id, "", 0.0);

found = 1;

break;

}

fprintf(fp\_temp, "%d %s %.2f\n", emp.id, emp.name, emp.salary);

}

if(found == 0) {

fprintf(fp\_temp, "%d %s %.2f\n", id, "", 0.0);

}

fclose(fp);

fclose(fp\_temp);

remove("employees.txt");

rename("temp.txt", "employees.txt");

printf("Record modified successfully!\n");

}

void delete\_record() {

FILE \*fp, \*fp\_temp;

struct Employee emp;

int id, found = 0;

fp = fopen("employees.txt", "r");

fp\_temp = fopen("temp.txt", "w");

printf("Enter employee ID to delete: ");

scanf("%d", &id);

while(fscanf(fp, "%d %s %f", &emp.id, emp.name, &emp.salary) != EOF) {

if(emp.id == id) {

found = 1;

} else {

fprintf(fp\_temp, "%d %s %.2f\n", emp.id, emp.name, emp.salary);

}

}

fclose(fp);

fclose(fp\_temp);

if(found == 0) {

printf("Record not found!\n");

remove("temp.txt");

} else {

remove("employees.txt");

rename("temp.txt", "employees.txt");

printf("Record deleted successfully!\n");

}

}

int main() {

int choice;

while(1) {

printf("\nEmployee System\n");

printf("---------------\n");

printf("1. Create Record\n");

printf("2. Display Records\n");

printf("3. Modify Record\n");

printf("4. Insert Record\n");

printf("5. Delete Record\n");

printf("6. Exit\n");

printf("\nEnter your choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

create\_record();

break;

case 2:

display\_records();

break;

case 3:

modify\_record();

break;

case 4:

insert\_record();

break;

case 5:

delete\_record();

break;

case 6:

exit(0);

break;

default:

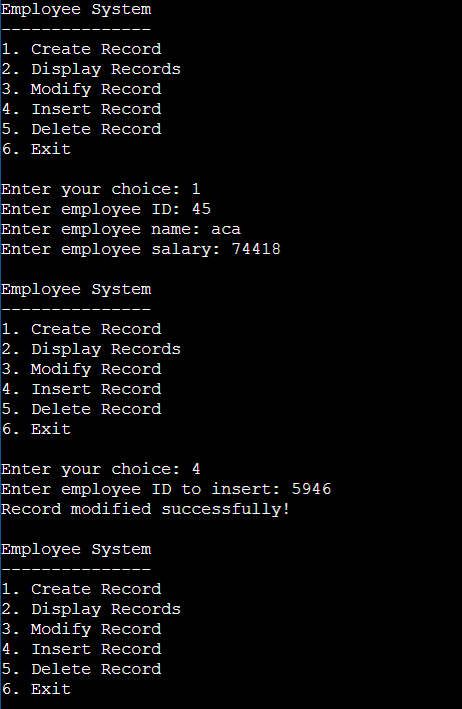
printf("Invalid choice!\n");

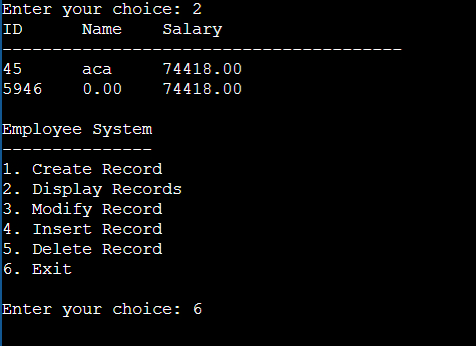
}

}

return 0;

}





OUTPUT:-

