S.No: 1	Exp. Name: Project Module	Date: 2024-06-13

Aim:

Project Module

Source Code:

hello.c

ID: 230381171042111126

2023-2027-H

K.Ramakrishnan College of Technology

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

#define MAX_EMPLOYEES 100

// Employee structure
struct Employee {
    int id;
    char name[50];
    float salary;
};

// Function prototypes
void addEmployee(struct Employee employees[], int *count);
void displayEmployees(struct Employee employees[], int count);
void deleteEmployeeById(struct Employee employees[], int *count,
int id);
void deleteEmployeeByName(struct Employee employees[], int
*count, char *name);
void displayEmployeesBySalaryRange(struct Employee employees[],
int count, float minSalary, float maxSalary);

int main() {
    struct Employee employees[MAX_EMPLOYEES];
    int count = 0;
    int choice, id;
    char name[50];
    float minSalary, maxSalary;

    while (1) {
        printf("\nEmployee Management System\n");
        printf("1. Add Employee\n");
        printf("2. Display Employees\n");
        printf("3. Delete Employee by ID\n");
        printf("4. Delete Employee by Name\n");
        printf("5. Display Employees by Salary Range\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                addEmployee(employees, &count);
                break;
            case 2:
                displayEmployees(employees, count);
```

```
                break;
            case 3:
                printf("Enter Employee ID to delete: ");
                scanf("%d", &id);
                deleteEmployeeById(employees, &count, id);
                break;
            case 4:
                printf("Enter Employee Name to delete: ");
                scanf("%s", name);
                deleteEmployeeByName(employees, &count, name);
                break;
            case 5:
                printf("Enter minimum salary: ");
                scanf("%f", &minSalary);
                printf("Enter maximum salary: ");
                scanf("%f", &maxSalary);
                displayEmployeesBySalaryRange(employees, count,
minSalary, maxSalary);
                break;
            case 6:
                return 0;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    }

    return 0;
}

// Function to add an employee
void addEmployee(struct Employee employees[], int *count) {
    if (*count >= MAX_EMPLOYEES) {
        printf("Cannot add more employees. Maximum limit
reached.\n");
        return;
    }

    struct Employee newEmployee;
    newEmployee.id = *count + 1;
    printf("Enter name: ");
    scanf("%s", newEmployee.name);
    printf("Enter salary: ");
    scanf("%f", &newEmployee.salary);

    employees[*count] = newEmployee;
    (*count)++;
```

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

// Function to display all employees
void displayEmployees(struct Employee employees[], int count) {
    if (count == 0) {
        printf("No employees to display.\n");
        return;
    }

    printf("\nEmployee List:\n");
    for (int i = 0; i < count; i++) {
        printf("ID: %d, Name: %s, Salary: %.2f\n",
               employees[i].id, employees[i].name,
employees[i].salary);
    }
}

// Function to delete an employee by ID
void deleteEmployeeById(struct Employee employees[], int *count,
int id) {
    int found = 0;

    for (int i = 0; i < *count; i++) {
        if (employees[i].id == id) {
            found = 1;
            for (int j = i; j < *count - 1; j++) {
                employees[j] = employees[j + 1];
            }
            (*count)--;
            printf("Employee deleted successfully.\n");
            break;
        }
    }

    if (!found) {
        printf("Employee with ID %d not found.\n", id);
    }
}

// Function to delete an employee by name
void deleteEmployeeByName(struct Employee employees[], int
*count, char *name) {
    int found = 0;
```

```
    for (int i = 0; i < *count; i++) {
        if (strcmp(employees[i].name, name) == 0) {
            found = 1;
            for (int j = i; j < *count - 1; j++) {
                employees[j] = employees[j + 1];
            }
            (*count)--;
            printf("Employee deleted successfully.\n");
            break;
        }
    }

    if (!found) {
        printf("Employee with name %s not found.\n", name);
    }
}

// Function to display employees by salary range
void displayEmployeesBySalaryRange(struct Employee employees[],
int count, float minSalary, float maxSalary) {
    int found = 0;

    printf("\nEmployees with salary between %.2f and %.2f:\n",
minSalary, maxSalary);
    for (int i = 0; i < count; i++) {
        if (employees[i].salary >= minSalary &&
employees[i].salary <= maxSalary) {
            printf("ID: %d, Name: %s, Salary: %.2f\n",
                   employees[i].id, employees[i].name,
employees[i].salary);
            found = 1;
        }
    }

    if (!found) {
        printf("No employees found in the given salary
range.\n");
    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output
Hello World