Experiment No. 9

Simulate a simple HR database system Using Structures in C

Aim: Design a C program that manages basic employee records for a company using structures. The program should simulate a simple HR database system by allowing storage and display of employee details such as:

- Employee Name
- Employee ID
- Department
- Salary

The system must support handling multiple employee records efficiently

Learning Outcomes:

After completing this experiment, students will be able to:

- Understand how to use structures in C to organize and store related data efficiently.
- Implement an array of structures to manage multiple employee records.
- Develop logic to store and retrieve structured data for multiple records.
- Simulate a basic HR database system and understand how real-world data management works in programming.

Theory:

What Is a Structure in C?

A structure in C is a user-defined data type that allows you to group different types of data under a single name.
It is useful when you want to store information about an entity that has multiple attributes.
For example, an employee has:

- Name (string)
- Employee ID (integer)
- Department (string)
- Salary (float)

These are different types of data — a structure helps to store them together in one unit.

Syntax of Structure

```
struct structure_name {
   data_type member1;
```

```
    data_type member2;
    ...
    data_type memberN;
};
```

Example Structure

```
struct Employee {
    char name[50];
    int empID;
    char department[30];
    float salary;
};
```

This structure stores all the necessary information about an employee.

How to Declare Variables of Structure

```
struct Employee e1, e2;   // Individual variables
struct Employee employees[50];  // Array of structures for multiple employees
```

Use of Structures in HR Simulation

In your HR database simulation, structures help store multiple employee records efficiently.

Importance of Using Structures in This HR Management System

- Structures are particularly valuable in this program because they enable:
- Efficient organization of related data by grouping different employee attributes under one unit.
- Management of multiple records through arrays of structures, allowing scalability for numerous employees.
- Simplified data processing, making input and output operations more structured and manageable.
- Practical representation of real-world HR systems, providing a foundation for building more advanced database applications.

Program Code:

Students are expected to implement this experiment by writing the complete C program for the Simulate a simple HR database system Using Structures in C.

Output:

Students are expected to execute the program and provide various sample outputs for the same code to demonstrate its functionality with different input values.

Conclusion: In this experiment, Simulated a simple HR database system Using Structures in C.