EMPLOYEE MANAGEMENT SYSTEM

Employee Management System project developed using C.

CODE 1- SOURCE CODE

main.c

```c
#include <stdio.h>

// Defining the Date structure
struct Date {
    int dd, mm, yy;
};

// Defining the Employee structure
struct Employee {
    int Empno;
    char EmpName[50];
    struct Date hiredate;
    float BasicSalary;
    float NetSalary;
};

// Function to calculate Netsalary
void calculateNetSalary(struct Employee *emp) {
    float DA = 0.4 * emp->BasicSalary;
    float TA = 0.1 * emp->BasicSalary;
    float PF = 0.05 * emp->BasicSalary;

    emp->NetSalary = emp->BasicSalary + DA + TA - PF;
}

// Function to accept employee data
void AcceptEmployeeData(struct Employee *emp) {
    printf("Please Enter Employee Number: ");
    scanf("%d", &emp->Empno);

    printf("Please Enter Employee Name: ");
    scanf("%s", emp->EmpName);

    printf("Please Enter Hire Date (DD MM YY): ");
    scanf("%d %d %d", &emp->hiredate.dd, &emp->hiredate.mm, &emp->hiredate.yy);

    printf("Please Enter Basic Salary: ");
    scanf("%f", &emp->BasicSalary);

    calculateNetSalary(emp);
}

// Function to display employee data
void DisplayEmployeeData(struct Employee emp) {
    printf("\nEmployee Number: %d\n", emp.Empno);
    printf("Employee Name: %s\n", emp.EmpName);
    printf("Hire Date: %02d/%02d/%02d\n", emp.hiredate.dd, emp.hiredate.mm, emp.hiredate.yy);
    printf("Basic Salary: %.2f\n", emp.BasicSalary);
    printf("Net Salary: %.2f\n", emp.NetSalary);
}
```

```c

int main() {
    int n;
    printf("Please Enter the number of employees: ");
    scanf("%d", &n);

    struct Employee emp[n];

    for (int i = 0; i < n; i++) {
        printf("\nPlease Enter details for Employee %d:\n", i + 1);
        AcceptEmployeeData(&emp[i]);
    }

    printf("\nBelow are the Employee Details:\n");
    for (int i = 0; i < n; i++) {
        DisplayEmployeeData(emp[i]);
    }

    return 0;
}
```

CODE 1-OUTPUT

Output

```
/tmp/LZBvHIYILo.o
Please Enter the number of employees: 3
Please Enter details for Employee 1:
Please Enter Employee Number: 1001
Please Enter Employee Name: gagan
Please Enter Hire Date (DD MM YY): 12 12 20
Please Enter Basic Salary: 70000
Please Enter details for Employee 2:
Please Enter Employee Number: 1002
Please Enter Employee Name: darshan
Please Enter Hire Date (DD MM YY): 23 07 18
Please Enter Basic Salary: 120000
Please Enter details for Employee 3:
Please Enter Employee Number: 1003
Please Enter Employee Name: salman
Please Enter Hire Date (DD MM YY): 23 05 19
Please Enter Basic Salary: 110000
Below are the Employee Details:

Employee Number: 1001
Employee Name: gagan
Hire Date: 12/12/20
Basic Salary: 70000.00
Net Salary: 101500.00

Employee Number: 1002
Employee Name: darshan
Hire Date: 23/07/18
Basic Salary: 120000.00
Net Salary: 174000.00

Employee Number: 1003
Employee Name: salman
Hire Date: 23/05/19
Basic Salary: 110000.00
Net Salary: 159500.00
```

CODE 2-SOURCE CODE

main.c

```c
#include <stdio.h>

// Defining the Date structure
struct Date {
    int dd, mm, yy;
};

// Defining the Employee structure
struct Employee {
    int Empno;
    char EmpName[50];
    struct Date hiredate;
    float BasicSalary;
    float NetSalary;
};

// Function to calculate Netsalary
float calculateNetSalary(float basicSalary) {
    float DA = 0.4 * basicSalary;
    float TA = 0.1 * basicSalary;
    float PF = 0.05 * basicSalary;

    return basicSalary + DA + TA - PF;
}

// Function to accept employee data
struct Employee acceptEmployeeData() {
    struct Employee emp;

    printf("Please Enter Employee Number: ");
    scanf("%d", &emp.Empno);

    printf("Please Enter Employee Name: ");
    scanf("%s", emp.EmpName);

    printf("Please Enter Hire Date (DD MM YY): ");
    scanf("%d %d %d", &emp.hiredate.dd, &emp.hiredate.mm, &emp.hiredate.yy);

    printf("Please Enter Basic Salary: ");
    scanf("%f", &emp.BasicSalary);

    emp.NetSalary = calculateNetSalary(emp.BasicSalary);

    return emp;
}

```

```c
// Function to display employee data
void displayEmployeeData(struct Employee emp) {
    printf("\nEmployee Number: %d\n", emp.Empno);
    printf("Employee Name: %s\n", emp.EmpName);
    printf("Hire Date: %02d/%02d/%02d\n", emp.hiredate.dd, emp.hiredate.mm, emp.hiredate
        .yy);
    printf("Basic Salary: %.2f\n", emp.BasicSalary);
    printf("Net Salary: %.2f\n", emp.NetSalary);
}

int main() {
    int n;
    printf("Please Enter the number of employees: ");
    scanf("%d", &n);

    struct Employee emp[n];

    for (int i = 0; i < n; i++) {
        printf("\nPlease Enter details for Employee %d:\n", i + 1);
        emp[i] = acceptEmployeeData();
    }

    printf("\nBelow are the Employee Details:\n");
    for (int i = 0; i < n; i++) {
        displayEmployeeData(emp[i]);
    }

    return 0;
```

CODE 2-OUTPUT

Output

```
/tmp/uKtawwyT3T.o
Please Enter the number of employees: 3
Please Enter details for Employee 1:
Please Enter Employee Number: 1001
Please Enter Employee Name: gagan
Please Enter Hire Date (DD MM YY): 12 12 20
Please Enter Basic Salary: 70000
Please Enter details for Employee 2:
Please Enter Employee Number: 1002
Please Enter Employee Name: darshan
Please Enter Hire Date (DD MM YY): 23 07 18
Please Enter Basic Salary: 120000
Please Enter details for Employee 3:
Please Enter Employee Number: 1003
Please Enter Employee Name: salman
Please Enter Hire Date (DD MM YY): 23 05 19
Please Enter Basic Salary: 110000
Below are the Employee Details:

Employee Number: 1001
Employee Name: gagan
Hire Date: 12/12/20
Basic Salary: 70000.00
Net Salary: 101500.00

Employee Number: 1002
Employee Name: darshan
Hire Date: 23/07/18
Basic Salary: 120000.00
Net Salary: 174000.00

Employee Number: 1003
Employee Name: salman
Hire Date: 23/05/19
Basic Salary: 110000.00
Net Salary: 159500.00
```

CONCLUSION

The development of the Employee Management System project has achieved the following key points and goals:

- Efficiently manages employee records by storing essential information.
- Implements modular functionalities for input, calculation, and display of employee data.
- Enhances code reusability and maintainability through the use of composition for date handling.
- Provides flexibility in module design by supporting both "pass by value" and "pass by reference" mechanisms.
- In conclusion, the Employee Management System project is a versatile and efficient solution for managing employee records and performing essential operations on them, ensuring data accuracy and code maintainability.