

ASSESSMENT 2 (JAN-14)

=====

1. Payroll Management System

Specifications:

Variables: Employee ID, name, designation, and salary.

Static & Const: Static variable for total employees; const for maximum number of employees.

Switch Case: Menu for adding, calculating, and displaying payroll.

Looping Statements: Loop through employee records.

Pointers: Pointer for salary calculation and updates.

Functions: Separate functions for each payroll operation.

Arrays: Store employee details.

Structures: Structure for employee details.

Nested Structures: Nested structure for personal and salary details.

Unions: Union for different types of allowances.

Nested Unions: Nested unions for various bonus types.

Output Expectations: Display employee details and payroll summary.

Menu Example:

1. Add Employee
2. Calculate Payroll
3. Display Payroll
4. Exit

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

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

```
#define MAX_EMPLOYEES 50
```

```
// Static variable for total employees
```

```
static int totalEmployees = 0;
```

```
// Structure for salary details
```

```
typedef struct {
```

```
    float basic;
```

```
    float hra;
```

```
    float da;
```

```
} SalaryDetails;
```

```
// Structure for employee details
```

```
typedef struct {
```

```
    int id;
```

```
    char name[50];
```

```
    char designation[50];
```

```
    SalaryDetails salary;
```

```
} Employee;
```

```
// Union for different allowances
```

```
typedef union {
```

```
    float travelAllowance;
```

```
    float medicalAllowance;
```

```
} Allowances;
```

```
// Nested union for various bonuses
```

```
typedef union {
```

```
    float yearlyBonus;  
    float performanceBonus;  
} Bonuses;
```

```
// Array to store employee records  
Employee employees[MAX_EMPLOYEES];
```

```
// Function prototypes  
void addEmployee();  
void calculatePayroll();  
void displayPayroll();
```

```
int main() {  
    int choice;  
  
    while (1) {  
        // Display menu  
        printf("\nPayroll Management System\n");  
        printf("1. Add Employee\n");  
        printf("2. Calculate Payroll\n");  
        printf("3. Display Payroll\n");  
        printf("4. Exit\n");  
        printf("Enter your choice: ");  
        scanf("%d", &choice);  
  
        switch (choice) {  
            case 1:  
                addEmployee();  
                break;  
            case 2:  
                calculatePayroll();  
                break;  
            case 3:  
                displayPayroll();  
                break;  
            case 4:  
                printf("Exiting system.\n");  
                return 0;  
            default:  
                printf("Invalid choice. Please try again.\n");  
        }  
    }  
}
```

```
// Function to add an employee  
void addEmployee() {  
    if (totalEmployees >= MAX_EMPLOYEES) {  
        printf("Employee limit reached. Cannot add more employees.\n");  
        return;  
    }  
}
```

```
Employee e;  
printf("Enter Employee ID: ");  
scanf("%d", &e.id);  
printf("Enter Employee Name: ");
```

```

scanf("%s", e.name);
printf("Enter Employee Designation: ");
scanf("%s", e.designation);
printf("Enter Basic Salary: ");
scanf("%f", &e.salary.basic);
printf("Enter HRA: ");
scanf("%f", &e.salary.hra);
printf("Enter DA: ");
scanf("%f", &e.salary.da);

employees[totalEmployees++] = e;
printf("Employee added successfully!\n");
}

// Function to calculate payroll
void calculatePayroll() {
    if (totalEmployees == 0) {
        printf("No employees available to calculate payroll.\n");
        return;
    }

    Allowances allowance;
    Bonuses bonus;

    for (int i = 0; i < totalEmployees; i++) {
        // Pointer to employee salary
        float *salary = &employees[i].salary.basic;

        allowance.travelAllowance = 0.1 * (*salary); // Example: 10% of basic salary
        bonus.yearlyBonus = 0.2 * (*salary);        // Example: 20% of basic salary

        printf("Payroll calculated for Employee ID %d\n", employees[i].id);
        printf("Travel Allowance: %.2f\n", allowance.travelAllowance);
        printf("Yearly Bonus: %.2f\n", bonus.yearlyBonus);
    }
}

// Function to display payroll
void displayPayroll() {
    if (totalEmployees == 0) {
        printf("No employee records available.\n");
        return;
    }

    printf("\nEmployee Details and Payroll:\n");
    for (int i = 0; i < totalEmployees; i++) {
        printf("ID: %d, Name: %s, Designation: %s\n",
            employees[i].id, employees[i].name, employees[i].designation);
        printf("Basic Salary: %.2f, HRA: %.2f, DA: %.2f\n",
            employees[i].salary.basic, employees[i].salary.hra, employees[i].salary.da);
    }
}

```