BCIT FIRST YEAR SECTION D

MADE BY: ANZAL(CT-25177), ADAM(CT-25190) AND AREEBA(CT25158)

Payroll Management System

Index

1. Project Title:

Payroll Management System

2. Project Description:

The Payroll Management System is a computerized software-based solution that calculates employees' monthly salaries by processing their salary structure, allowances, deductions, working hours, tenure bonuses, unpaid leaves and tax amounts. Payroll automation eliminates manual calculation errors, ensures accuracy, generates salary slips and efficiently maintains employee payment records.

3. Project Methodology:

3.1 Dataset / Input Data:

* Employee ID, Name
* Basic Salary

* Allowances (House Rent, Medical, Transport)
* Deductions (Tax, Fund)
* Hours Worked
* Unpaid Time Off
* Bonus Rate per Hour
* Tenure (Years of Service)

3.2 Tools and Technologies:

* Language: C Programming
* IDE: Dev C++ / CodeBlocks / Visual Studio Code
* Platform: Windows OS
* Compiler: GCC

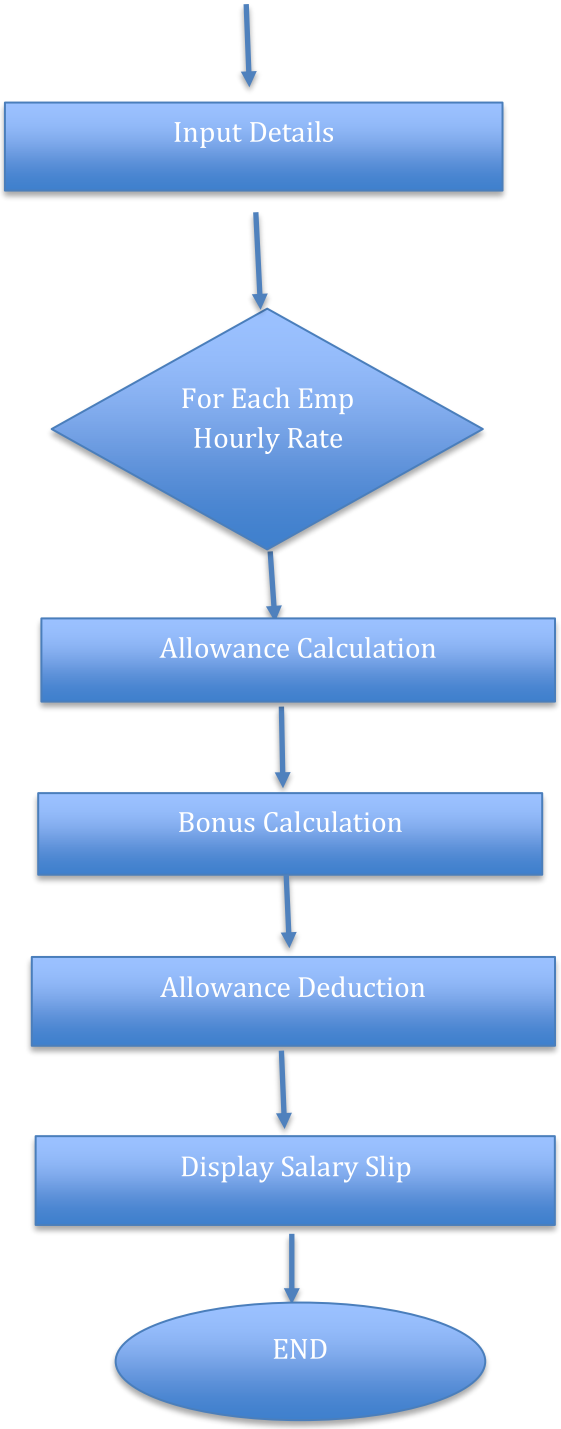
3.3 Algorithm:

1. Start
2. Input employee details
3. Calculate total allowances
4. Calculate performance bonus based on hours worked
5. Calculate tenure bonus depending on years of service
6. Calculate unpaid leave deduction based on hourly rate
7. Calculate gross salary
8. Subtract deductions from gross salary
9. Display payroll slip with final net pay
10. End

3.4 Objectives:

* Automate salary calculation process
* Provide accurate and fast payroll slip generation
* Reduce human calculation error
* Include bonus and deduction logic dynamically

3.5 Flowchart:



3.6 Expected Outcomes:

* Accurate payroll slip generation for each employee

* Automatic net salary computation

* Professional structured salary breakdown

3.7 Goals:

* Build a clear logical payroll model
* Make calculation transparent & replicable
* Implement in modular C program format

4. Justification – Why It Is a Complex Computing Problem:

Payroll contains multiple salary influencing variables. Each factor can change calculation logic. Accounting rules, working hours, deduction criteria & tenure bonuses require conditional logic, mathematical processing and systematic breakdown which cannot be reliably handled through manual repetitive calculations.

5. Industrial / Commercial Product Potential:

This Payroll System can be used in small offices, colleges, software houses and business organizations to prepare monthly salary sheets. It can be expanded into a full HR management module by adding attendance system, employee database storage, multi employee batch payroll processing and salary exporting to PDF / CSV files.

6. Program Implementation:

The following C program calculates payroll for multiple employees, including basic pay, overtime pay, gross pay, tax deduction, and net pay.

```
#include <stdio.h>

int main() {
   int n, i;
   int empID[50], tenure[50];
   char name[50][50];
   float hourlyRate[50], hoursWorked[50], overtimeHours[50],
unpaidHours[50];
```

```
    float houseAllowance[50], medicalAllowance[50],
transportAllowance[50];
    float basicPay[50], overtimePay[50], allowanceTotal[50],
performanceBonus[50];
    float tenureBonus[50], unpaidDeduction[50], grossPay[50], tax[50],
netPay[50];
    printf("===== PAYROLL SYSTEM =====\n");
    printf("Enter number of employees: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        printf("\n--- Enter details for Employee %d ---\n", i + 1);
        printf("Enter Employee ID: ");
        scanf("%d", &empID[i]);
        printf("Enter Employee Name: ");
        scanf(" %[^\n]", name[i]);
        printf("Enter Hourly Rate: ");
        scanf("%f", &hourlyRate[i]);
        printf("Enter Hours Worked: ");
        scanf("%f", &hoursWorked[i]);
        printf("Enter Overtime Hours: ");
        scanf("%f", &overtimeHours[i]);
        printf("Enter Unpaid Leave Hours: ");
        scanf("%f", &unpaidHours[i]);
        printf("Enter House Allowance: ");
        scanf("%f", &houseAllowance[i]);
        printf("Enter Medical Allowance: ");
        scanf("%f", &medicalAllowance[i]);
        printf("Enter Transport Allowance: ");
        scanf("%f", &transportAllowance[i]);
        printf("Enter Tenure (years of service): ");
        scanf("%d", &tenure[i]);}
        // Calculations
        basicPay[i] = hoursWorked[i] * hourlyRate[i];
        overtimePay[i] = overtimeHours[i] * (hourlyRate[i] * 1.5);
        allowanceTotal[i] = houseAllowance[i] + medicalAllowance[i] +
transportAllowance[i];
        performanceBonus[i] = (hoursWorked[i] > 160) ? (hoursWorked[i]
- 160) * (hourlyRate[i] * 0.25) : 0;
        tenureBonus[i] = tenure[i] * (hourlyRate[i] * 10);
        unpaidDeduction[i] = unpaidHours[i] * hourlyRate[i];
        grossPay[i] = basicPay[i] + overtimePay[i] + allowanceTotal[i] + performanceBonus[i] + tenureBonus[i];
        grossPay[i] -= unpaidDeduction[i];
        tax[i] = grossPay[i] * 0.10;
        netPay[i] = grossPay[i] - tax[i];
    }
    printf("\n===== PAYROLL SUMMARY =====\n");
    for (i = 0; i < n; i++) {
        printf("\nEmployee %d\n", i + 1);
        printf("ID: %d\n", empID[i]);
        printf("Name: %s\n", name[i]);
        printf("Basic Pay: %.2f\n", basicPay[i]);
        printf("Overtime Pay: %.2f\n", overtimePay[i]);
        printf("Total Allowances: %.2f\n", allowanceTotal[i]);
        printf("Performance Bonus: %.2f\n", performanceBonus[i]);
        printf("Tenure Bonus: %.2f\n", tenureBonus[i]);
        printf("Unpaid Leave Deduction: %.2f\n", unpaidDeduction[i]);
        printf("Gross Pay: %.2f\n", grossPay[i]);
        printf("Tax (10%%): %.2f\n", tax[i]);
        printf("Net Pay: %.2f\n", netPay[i]);
        printf("--------------------------\n");
    }
    printf("===== END OF PAYROLL =====\n");
    return 0;
}
```

7. Conclusion:

The Payroll Management System streamlines employee salary management by integrating multiple salary-related parameters into a single automated platform. It reduces human errors, speeds up payroll processing, and ensures accurate computation of allowances, bonuses, deductions, and net pay. This system is practical for organizations of any size and provides a foundation for further enhancements such as attendance tracking, reporting, and integration with financial systems. Ultimately, it enhances operational efficiency and ensures transparency and reliability in employee salary management.