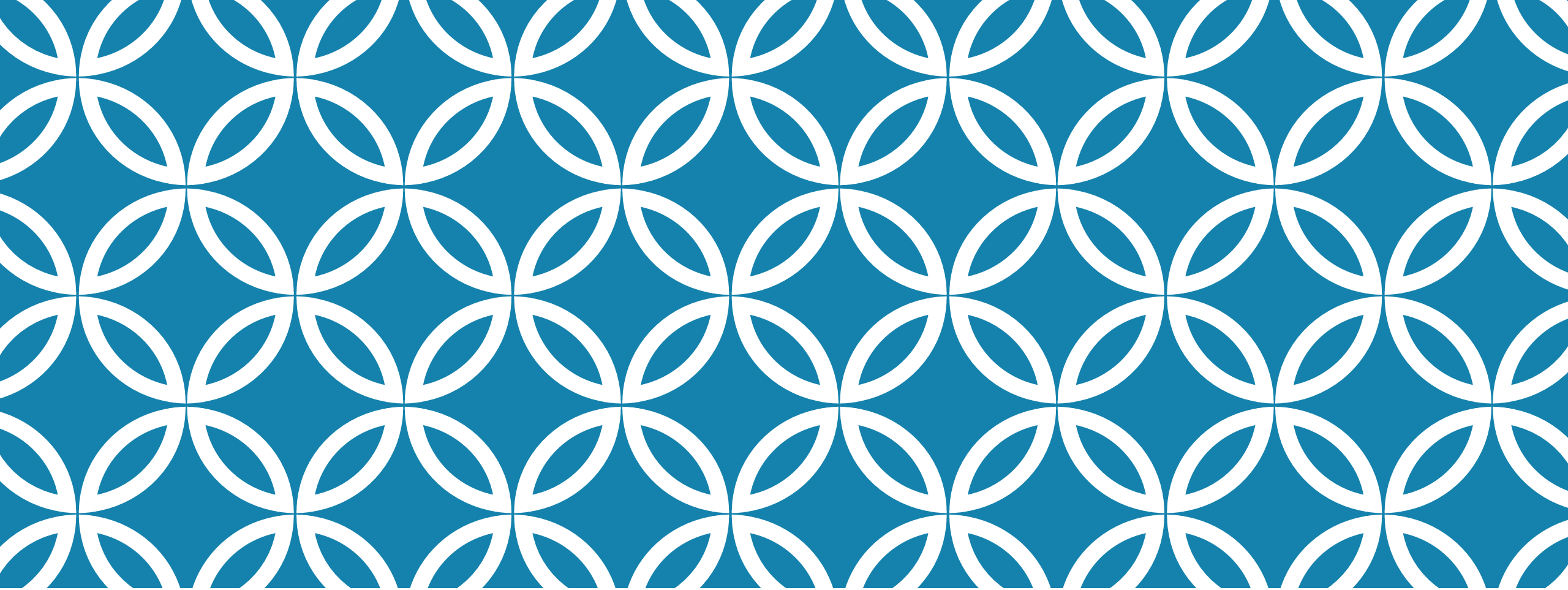


CCP PAYROLL SYSTEM

Project members:
Muhammad Adam (CT-190)
Anzal Ansari (CT-177)
Areeba Ilyas (CT-158)

INTRODUCTION TO PAYROLL SYSTEM

The Payroll system computes employee salaries by taking into account **working hours, overtime, tax deductions,** and **other components.** This project implements a simple payroll calculator in C that demonstrates the use of **arrays, loops** and basic **arithmetic** operations.

OBJECTIVES

The objective of this program is that it processes payroll for a set of employees. For each individual employee the program will

- Take input : employee name, hours worked, hourly pay
- Calculate
    1. Regular pay
    2. Overtime pay
    3. Gross pay
    4. Tax reduction (tax% of gross pay)
    5. Net pay (gross pay – tax reduction)

- Storing : we will use arrays to store all this information of all the employees
- Output : Display a clean , formatted payroll summary of all the employees

REQUIREMENTS AND TOOLS USED

Language used

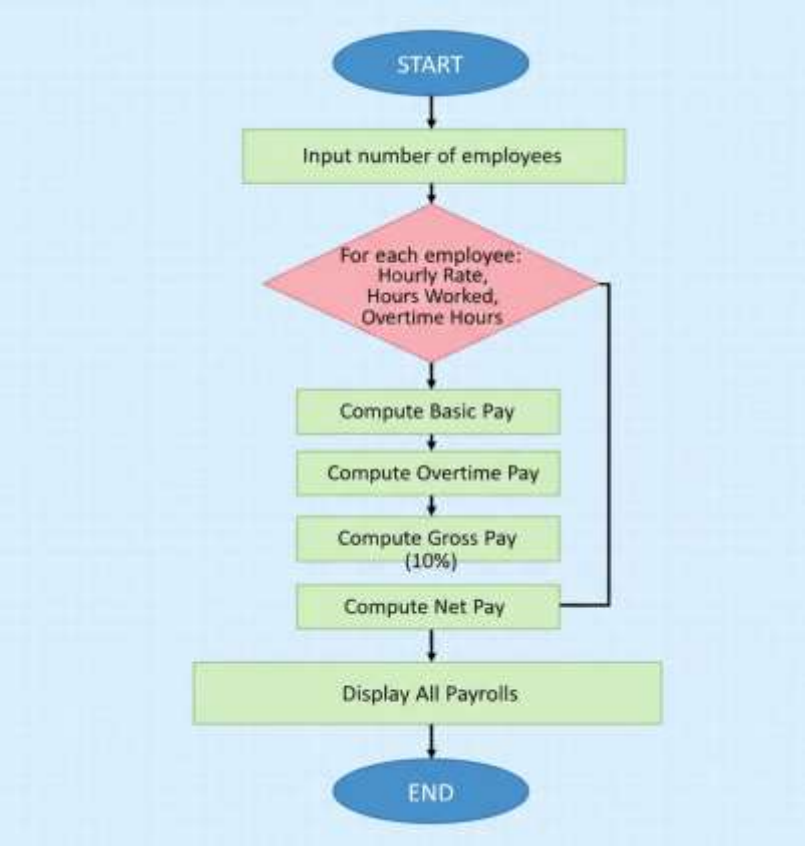
Compiler used:

GCC or any C complier

Implementation of

- Arrays
- Loops
- Input/Output

FLOWCHART OF PAYROLL SYSTEM



Here is a basic presentation of the code in a flowchart

ALGORITHM OF CODE

1.Start

2. Read n (number of employees)

3. For i =1 to n do:
   - Read Employee ID and Name
   - Read Hourly Rate, Hours Worked, Overtime Hours
   - BasicPay = HoursWorked × HourlyRate
   - OvertimePay = OvertimeHours × 1.5 × HourlyRate
   - GrossPay = basicPay + overtimePay
   - Tax = grossPay × 0.10
   - NetPay = grossPay - tax

4. End For

5. Display payroll summary for each employee

6. End

PROGRAM CODE OF PAYROLL SYSTEM

```c
//CCP PAYROLL SYSTEM BY ADAM(CT-190),AREEBA(CT-158) AND ANZAL(CT-177)
#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];
```

PROGRAM CODE OF PAYROLL SYSTEM

```
47      }
48      printf("\n===== PAYROLL SUMMARY =====\n");
49      for (i = 0; i < n; i++) {
50          printf("\nEmployee %d\n", i + 1);
51          printf("ID: %d\n", empID[i]);
52          printf("Name: %s\n", name[i]);
53          printf("Basic Pay: %.2f\n", basicPay[i]);
54          printf("Overtime Pay: %.2f\n", overtimePay[i]);
55          printf("Total Allowances: %.2f\n", allowanceTotal[i]);
56          printf("Performance Bonus: %.2f\n", performanceBonus[i]);
57          printf("Tenure Bonus: %.2f\n", tenureBonus[i]);
58          printf("Unpaid Leave Deduction: %.2f\n", unpaidDeduction[i]);
59          printf("Gross Pay: %.2f\n", grossPay[i]);
60          printf("Tax (10%%): %.2f\n", tax[i]);
61          printf("Net Pay: %.2f\n", netPay[i]);
62          printf("--------------------------\n");
63      }
64      printf("===== END OF PAYROLL =====\n");
65      return 0;
66  }
```

EXPLANATION OF KEY PARTS

1. Input section

- reads number of employees and per-employee data (ID, Name, Hourly Rate, Hours Worked, Overtime Hours).

2. Calculation section

- calculates basic pay, overtime pay (1.5×), gross pay, tax (10%) and net pay.

3. Output section

- prints a formatted payroll summary for all employees.

SAMPLE INPUT & OUTPUT (EXAMPLE)

Sample Input:

Number of employees: 1
Employee 1: ID=101, Name=Ahmed, HourlyRate=150, HoursWorked=20, OverTime=5, UnpaidHours=3, HouseAllowance=90, MedicalAllowance= 190, TransportAllowance= 100, Tenure= 4

Sample Output:

Employee 1: Net Pay= 9049.5, Overtime= 1125, Tax= 1005.5

CONCLUSION AND FUTURE WORK

Conclusion:

The CCP Payroll System offers a simple, maintainable implementation for calculating employee pay.

Future Enhancements:

- Add persistent storage (file I/O) to save/load employee records.
- Add validation and error handling for user inputs.
- Add support for allowances, deductions, and variable tax brackets.
- Create a GUI or web front-end for ease of use.