TITLE : Simple Payroll System

STUDENT NAME :	ID
KRUSH GABANI	24AIML009
VRAJ GOTI	24AIML011
HEPIN BHUVA	24AIML004

Objective : To create a program that converts temperatures between Celsius, Fahrenheit, and Kelvin using C.

Course Learning Outcome (CLO):By the end of this project, students will be able to develop a program that converts temperature units by applying mathematical formulas using C

Program Overview:The Simple Payroll System is a C program designed to automate the process of calculating employee salaries based on their hours worked and hourly rate. It also manages employee records by saving and retrieving data from files. This system is modular, user-friendly, and teaches key programming concepts such as arithmetic operations, file handling, and menu-driven control.

Program Overview: The Simple Payroll System is a C program designed to automate the process of calculating employee salaries based on their hours worked and hourly rate. It also manages employee records by saving and retrieving data from files. This system is modular, user-friendly, and teaches key programming concepts such as arithmetic operations, file handling, and menu-driven control.

Key Function :

1. Input Function: Collect employee details (name, ID, hours worked, hourly rate).Example: void getEmployeeDetails()
2. Calculation Function: Calculate salary using the formula:Salary = Hours Worked * Hourly Rate.Example: float calculateSalary()
3. File Writing Function: Save employee details and salary to a file.Example: void saveToFile()

4. File Reading Function: Read and display employee records from the file.Example: void displayRecords()
5. Output Function: Display the calculated salary on the screen.Example:
void displaySalary()

#CODE :

```
#include <stdio.h>
#include <stdlib.h>
// Function to calculate salaryfloat calculateSalary(float hoursWorked, float hourlyRate) {
    return hoursWorked * hourlyRate;
}
```

```
int main() {
  FILE *file; // File pointer for payroll records
   char employeeName[50];
  float hoursWorked, hourlyRate, salary;
  char choice;
  // Open the file in append mode     file = fopen("payroll.txt", "a+");
  if (file == NULL) {
       printf("Error opening file!\n");
       return 1; // Exit if file fails to open
    }
do {
      // Input employee details
       printf("Enter employee name: ");
       scanf(" %[^\n]", employeeName); // Read string with spaces
       printf("Enter hours worked: ");
       scanf("%f", &hoursWorked);
       printf("Enter hourly rate: ");
       scanf("%f", &hourlyRate);
```

```
 // Calculate salary        salary = calculateSalary(hoursWorked, hourlyRate);
       // Save details to file
     fprintf(file, "Employee Name: %s, Hours Worked: %.2f, Hourly Rate: %.2f, Salary: %.2f\n",
employeeName, hoursWorked, hourlyRate, salary);
      // Display salary on console
      printf("\nEmployee Name: %s\n", employeeName);
      printf("Hours Worked: %.2f\n", hoursWorked);
      printf("Hourly Rate: %.2f\n", hourlyRate);
      printf("Calculated Salary: %.2f\n", salary);
// Ask if user wants to add another employee
      printf("\nDo you want to add another employee? (y/n): ");
      scanf(" %c", &choice);
   } while (choice == 'y' || choice == 'Y');
// Continue if user enters 'y' or 'Y'    fclose(file);
// Close the file
  printf("\nPayroll records have been saved to 'payroll.txt'.\n");
  return 0;
}
```

OUTPUT :

```
PS C:\Users\VRAJ GOTI\OneDrive\Desktop\VRAJ>
cd "c:\Users\VRAJ GOTI\OneDrive\Desktop\VRAJ\" ; if ($?) { gcc 2.c -o 2 } ; if ($?) { .\2 }
Enter employee name: VRAJ GOTI
Enter hours worked: 50.00
Enter hourly rate: 20.00

Employee Name: VRAJ GOTI
Hours Worked: 50.00
Hourly Rate: 20.00
Calculated Salary: 1000.00

Do you want to add another employee? (y/n): y
Enter employee name: KRUSH GABANI
Enter hours worked: 35.00
Enter hourly rate: 15.00

Employee Name: KRUSH GABANI
Hours Worked: 35.00
Hourly Rate: 15.00
Calculated Salary: 525.00

Do you want to add another employee? (y/n): n

Payroll records have been saved to 'payroll.txt'.
PS C:\Users\VRAJ GOTI\OneDrive\Desktop\VRAJ>
```