## STRUCT 1

main.c

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

#define NUM_EMPLOYEES 5

struct emp
{
    char empFName[50];
    char empLName[50];
    int baseSalary;
    int overTimeHours;
};

int main()
{
    struct emp employees[NUM_EMPLOYEES];
    int i;

    // Reading employee details
    for (i = 0; i < NUM_EMPLOYEES; i++)
    {
        printf("Enter first name: ");
        scanf("%s", employees[i].empFName);

        printf("Enter last name: ");
        scanf("%s", employees[i].empLName);

        printf("Enter base salary: ");
        scanf("%d", &employees[i].baseSalary);

        printf("Enter the overtime hours: ");
        scanf("%d", &employees[i].overTimeHours);
        printf("\n");
    }

    printf("\nEmployee Monthly Salaries:\n");
    // Calculating and displaying salaries
    for (i = 0; i < NUM_EMPLOYEES; i++)
    {
        int monthlySalary = employees[i].baseSalary + (employees[i].overTimeHours * 20);
        printf("%s %s %d\n", employees[i].empFName, employees[i].empLName, monthlySalary);
    }

    return 0;
}
```

Gabon_Johnrey_Struct1_Prog1
Sources
main.c

```
Enter first name: John
Enter last name: Smith
Enter base salary: 3000
Enter the overtime hours: 10

Enter first name: Mary
Enter last name: Johnson
Enter base salary: 3500
Enter the overtime hours: 15

Enter first name: David
Enter last name: Wilson
Enter base salary: 4000
Enter the overtime hours: 5

Enter first name: Sarah
Enter last name: Brown
Enter base salary: 3800
Enter the overtime hours: 8

Enter first name: Michael
Enter last name: Davis
Enter base salary: 4200
Enter the overtime hours: 12


Employee Monthly Salaries:
John Smith 3200
Mary Johnson 3800
David Wilson 4100
Sarah Brown 3960
Michael Davis 4440
```

## STRUCT 2

Gabon_Johnrey_Struct2_Prog1
Sources
main.c

main.c

```c
#include <stdio.h>
#include <string.h>
#define NUM_STUDENTS 10
struct std {
    char stdFName[50];
    char stdLName[50];
    int testScore;
    char grade;
};
// Function to calculate grade based on test score
char calculateGrade(int score) {
    if (score >= 90) return 'A';
    else if (score >= 80) return 'B';
    else if (score >= 70) return 'C';
    else if (score >= 60) return 'D';
    else return 'F';
}
int main() {
    struct std students[NUM_STUDENTS];
    int i, highestScore = -1;
    char topStudentFName[50], topStudentLName[50];
    // Reading students' data
    for (i = 0; i < NUM_STUDENTS; i++) {
        printf("Enter first name of student %d: ", i + 1);
        scanf("%s", students[i].stdFName);
        printf("Enter last name of student %d: ", i + 1);
        scanf("%s", students[i].stdLName);
        printf("Enter test score of student %d (0-100): ", i + 1);
        scanf("%d", &students[i].testScore);
        // Validate test score input
        while (students[i].testScore < 0 || students[i].testScore > 100) {
            printf("Invalid score! Enter test score of student %d (0-100): ", i + 1);
            scanf("%d", &students[i].testScore);
        }
        // Assign grade based on test score
        students[i].grade = calculateGrade(students[i].testScore);
        // Check for highest score
        if (students[i].testScore > highestScore) {
            highestScore = students[i].testScore;
            strcpy(topStudentFName, students[i].stdFName);
            strcpy(topStudentLName, students[i].stdLName);
        }
    }
    // Output students' data
    printf("\nStudent Grades:\n");
    for (i = 0; i < NUM_STUDENTS; i++) {
        printf("%s, %s: Test Score = %d, Grade = %c\n",
               students[i].stdLName, students[i].stdFName,
               students[i].testScore, students[i].grade);
    }
    // Output highest score details
    printf("\nHighest Test Score:\n");
    printf("Name: %s, %s\n", topStudentLName, topStudentFName);
    printf("Score: %d\n", highestScore);
    return 0;
}
```

```
Enter first name of student 1: John
Enter last name of student 1: Smith
Enter test score of student 1 (0-100): 95

Enter first name of student 2: Mary
Enter last name of student 2: Johnson
Enter test score of student 2 (0-100): 87

Enter first name of student 3: David
Enter last name of student 3: Wilson
Enter test score of student 3 (0-100): 73

Enter first name of student 4: Sarah
Enter last name of student 4: Brown
Enter test score of student 4 (0-100): 92

Enter first name of student 5: Michael
Enter last name of student 5: Silva
Enter test score of student 5 (0-100): 89

Enter first name of student 6: Emma
Enter last name of student 6: Taylor
Enter test score of student 6 (0-100): 88

Enter first name of student 7: James
Enter last name of student 7: Anderson
Enter test score of student 7 (0-100): 78

Enter first name of student 8: Lisa
Enter last name of student 8: Martinez
Enter test score of student 8 (0-100): 91

Enter first name of student 9: Robert
Enter last name of student 9: Thomas
Enter test score of student 9 (0-100): 83

Enter first name of student 10: Jennifer
Enter last name of student 10: White
Enter test score of student 10 (0-100): 79
```

```
Student Grades:
Smith, John: Test Score = 95, Grade = A
Johnson, Mary: Test Score = 87, Grade = B
Wilson, David: Test Score = 73, Grade = C
Brown, Sarah: Test Score = 92, Grade = A
Silva, Michael: Test Score = 89, Grade = B
Taylor, Emma: Test Score = 88, Grade = B
Anderson, James: Test Score = 78, Grade = C
Martinez, Lisa: Test Score = 91, Grade = A
Thomas, Robert: Test Score = 83, Grade = B
White, Jennifer: Test Score = 79, Grade = C

Highest Test Score:
Name: Smith, John
Score: 95

Process returned 0 (0x0)   execution time : 97.987 s
Press any key to continue.
```