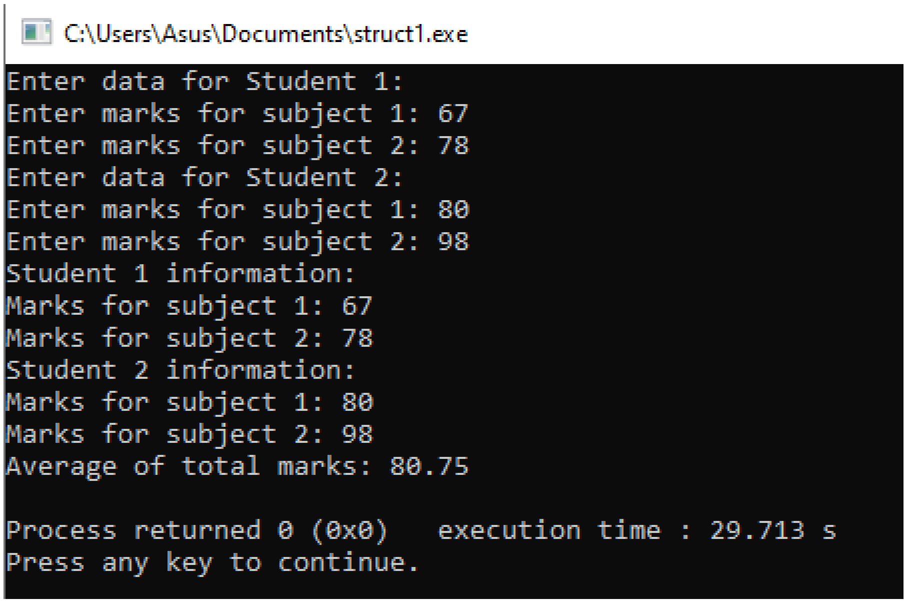
1. Create a " Student " structure with the member's name, age, and total marks. Write a C program to input data for two students, display their information, and find the average of total marks.
   Input:

```
#include <stdio.h>

// Function to input student information
void inputStudentData(int studentNumber, int *marks1, int
*marks2) {
    printf("Enter data for Student %d:\n", studentNumber);
    printf("Enter marks for subject 1: ");
    scanf("%d", marks1);
    printf("Enter marks for subject 2: ");
    scanf("%d", marks2);
}

// Function to display student information
void displayStudentData(int studentNumber, int marks1, int
marks2) {
    printf("Student %d information:\n", studentNumber);
    printf("Marks for subject 1: %d\n", marks1);
    printf("Marks for subject 2: %d\n", marks2);
}

int main() {
    int marks1_student1, marks2_student1;  // Marks for student 1
    int marks1_student2, marks2_student2;  // Marks for student 2

    // Input data for both students
    inputStudentData(1, &marks1_student1, &marks2_student1);
    inputStudentData(2, &marks1_student2, &marks2_student2);

    // Display information for both students
    displayStudentData(1, marks1_student1, marks2_student1);
    displayStudentData(2, marks1_student2, marks2_student2);

    // Calculate and display the average of total marks
    float average = (marks1_student1 + marks2_student1 +
marks1_student2 + marks2_student2) / 4.0;
    printf("Average of total marks: %.2f\n", average);

    return 0;
}
```

Output:



C:\Users\Asus\Documents\struct1.exe

```
Enter data for Student 1:
Enter marks for subject 1: 67
Enter marks for subject 2: 78
Enter data for Student 2:
Enter marks for subject 1: 80
Enter marks for subject 2: 98
Student 1 information:
Marks for subject 1: 67
Marks for subject 2: 78
Student 2 information:
Marks for subject 1: 80
Marks for subject 2: 98
Average of total marks: 80.75

Process returned 0 (0x0)   execution time : 29.713 s
Press any key to continue.
```

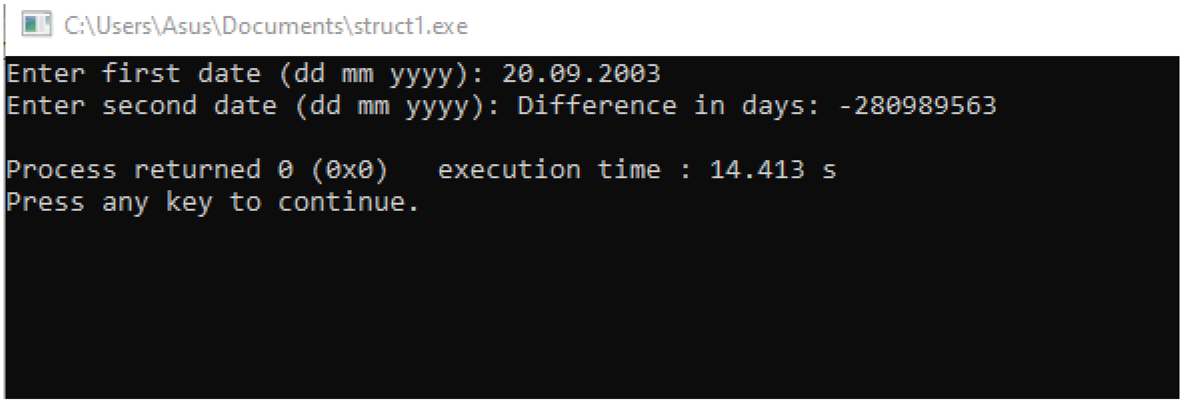
2. Define a " Date " structure with members day, month, and year. Write a C program to input two dates and find the difference in days between them.

   Input:

```
#include <stdio.h>

struct Date {
    int day;
    int month;
    int year;
};

// Function to calculate the difference in days between two dates
int differenceInDays(struct Date date1, struct Date date2) {
    // Days in each month (ignoring leap years for simplicity)
    int daysInMonth[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30,
31};

    // Calculate total days for each date
    int days1 = date1.day;
    for (int i = 1; i < date1.month; ++i) {
        days1 += daysInMonth[i];
    }
    days1 += date1.year * 365;

    int days2 = date2.day;
    for (int i = 1; i < date2.month; ++i) {
        days2 += daysInMonth[i];
    }
    days2 += date2.year * 365;

    // Calculate and return the difference in days
    return days2 - days1;
}
```

```
int main() {
    struct Date date1, date2;

    // Input for the first date
    printf("Enter first date (dd mm yyyy): ");
    scanf("%d %d %d", &date1.day, &date1.month, &date1.year);

    // Input for the second date
    printf("Enter second date (dd mm yyyy): ");
    scanf("%d %d %d", &date2.day, &date2.month, &date2.year);

    // Calculate and display the difference in days
    int difference = differenceInDays(date1, date2);
    printf("Difference in days: %d\n", difference);

    return 0;
}
```

Output:



```
Enter first date (dd mm yyyy): 20.09.2003
Enter second date (dd mm yyyy): Difference in days: -280989563

Process returned 0 (0x0)   execution time : 14.413 s
Press any key to continue.
```

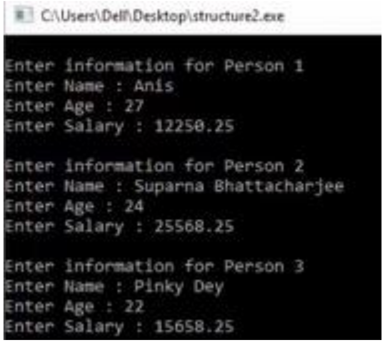
3. Create an " Employee " structure to store employee details such as employee ID, name, and salary. Write a program to input data for three employees, find the highest salary employee, and display their information.

   Input:

```
#include <stdio.h>

struct Employee {
    char name[50];
    int employeeId;
    float salary;
};

// Function to find the employee with the highest salary
int findHighestSalaryEmployee(struct Employee employees[], int size) {
    float maxSalary = employees[0].salary;
    int maxIndex = 0;

    for (int i = 1; i < size; ++i) {
        if (employees[i].salary > maxSalary) {
            maxSalary = employees[i].salary;
            maxIndex = i;
        }
    }

    return maxIndex;
}

// Function to display employee information
```

```
void displayEmployeeInformation(struct Employee employee) {
    printf("Employee Information:\n");
    printf("Name: %s\n", employee.name);
    printf("Employee ID: %d\n", employee.employeeId);
    printf("Salary: %.2f\n", employee.salary);
}

int main() {
    struct Employee employees[3];

    // Input data for three employees
    for (int i = 0; i < 3; ++i) {
        printf("Enter data for Employee %d:\n", i + 1);
        printf("Name: ");
        scanf("%s", employees[i].name);  // Assuming single-word names for simplicity
        printf("Employee ID: ");
        scanf("%d", &employees[i].employeeId);
        printf("Salary: ");
        scanf("%f", &employees[i].salary);
    }

    // Find the employee with the highest salary
    int highestSalaryIndex = findHighestSalaryEmployee(employees, 3);

    // Display information for the employee with the highest salary
    printf("\nEmployee with the highest salary:\n");
    displayEmployeeInformation(employees[highestSalaryIndex]);
    return 0;
}
```

Output:



```
Enter information for Person 1
Enter Name : Anis
Enter Age : 27
Enter Salary : 12250.25

Enter information for Person 2
Enter Name : Suparna Bhattacharjee
Enter Age : 24
Enter Salary : 25568.25

Enter information for Person 3
Enter Name : Pinky Dey
Enter Age : 22
Enter Salary : 15658.25
```

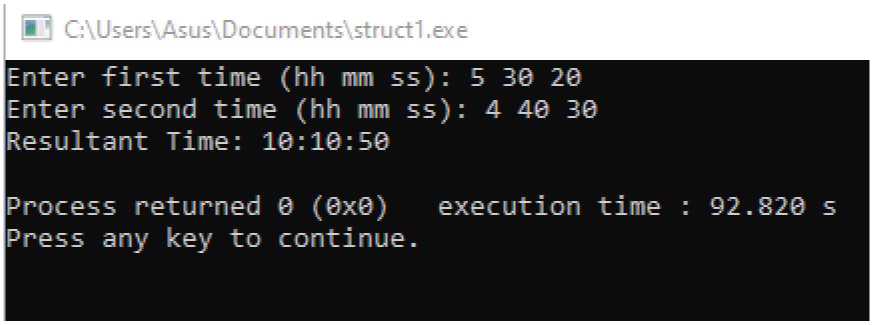
4. Define a structure named Time with members hours, minutes, and seconds. Write a C program to input two times, add them, and display the result in proper time format.

   Input:

```
#include <stdio.h>

struct Time {
    int hours;
    int minutes;
    int seconds;
};

// Function to add two times
struct Time addTimes(struct Time time1, struct Time time2) {
    struct Time result;

    // Add seconds
    result.seconds = time1.seconds + time2.seconds;

    // Adjust minutes if seconds exceed 60
    result.minutes = result.seconds / 60;
    result.seconds %= 60;

    // Add minutes
    result.minutes += time1.minutes + time2.minutes;

    // Adjust hours if minutes exceed 60
    result.hours = result.minutes / 60;
    result.minutes %= 60;

    // Add hours
    result.hours += time1.hours + time2.hours;

    return result;
}

// Function to display time in proper format
void displayTime(struct Time time) {
    printf("Resultant Time: %02d:%02d:%02d\n", time.hours,
time.minutes, time.seconds);
}

int main() {
    struct Time time1, time2, result;

    // Input for the first time
    printf("Enter first time (hh mm ss): ");
    scanf("%d %d %d", &time1.hours, &time1.minutes,
&time1.seconds);
```

```
    // Input for the second time
    printf("Enter second time (hh mm ss): ");
    scanf("%d %d %d", &time2.hours, &time2.minutes, &time2.seconds);

    // Add the two times
    result = addTimes(time1, time2);

    // Display the resultant time
    displayTime(result);

    return 0;
}
```

Output:



```
Enter first time (hh mm ss): 5 30 20
Enter second time (hh mm ss): 4 40 30
Resultant Time: 10:10:50

Process returned 0 (0x0)   execution time : 92.820 s
Press any key to continue.
```