

# 자료구조론 CC343\_2207

## Programming assignment 3

경기대학교 컴퓨터공학부

201511837 이상민

### Programming Exercises

1. Declare a structure that represents the following hierarchical information.

- (a) Student
  - (b) Roll Number
  - (c) Name
    - (i) First name
    - (ii) Middle Name
    - (iii) Last Name
  - (d) Sex
  - (e) Date of Birth
    - (i) Day
    - (ii) Month
    - (iii) Year
  - (f) Marks
    - (i) English
    - (ii) Mathematics
    - (iii) Computer Science

Source code :

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

```
struct Stdent {
```

```
    int number;
```

```
};
```

```
struct Roll_Number {
```

```
    int b;
```

```
};
```

```
struct Name {
```

```
    char Firstname[10];
```

```
    char Middlename[10];
```

```
    char Lastname[10];
```

```
};
```

```
struct sex {
```

```
    int gender;
```

```
};
```

```
struct Date_of_birth {
```

```
    int day;
```

```
    int month;
```

```
    int year;
```

```
};
```

```
struct Marks {
```

```
    int English;
```

```
    int Mathematics;
```

```
    int Computer;
```

```
};
```

2. Define a structure to store the name, an array marks[] which stores the marks of three different subjects, and a character grade. Write a program to display the details of the student whose name is entered by the user. Use the structure definition of the first question to make an array of students. Display the name of the students who have secured less than 40% of the aggregate.

이름을 저장할 구조, 세 개의 다른 과목의 마크를 저장하는 배열 표시[] 및 문자 등급을 정의한다. 사용자가 이름을 입력한 학생의 세부사항을 표시하는 프로그램을 작성한다. 첫 번째 문제의 구조 정의를 사용하여 학생 배열을 만드십시오. 골재량의 40% 미만을 확보한 학생의 이름을 표시한다.

```
Enter information of students:

For roll number1,
Enter first name: lee
Enter marks: 40

For roll number2,
Enter first name: kim
Enter marks: 80

For roll number3,
Enter first name: koi
Enter marks: 30

For roll number4,
Enter first name: jio
Enter marks: 20

For roll number5,
Enter first name: ijo
Enter marks: 90
Displaying Information:

Roll number: 3
First name: koi
Marks: 30.0

Roll number: 4
First name: jio
Marks: 20.0
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

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

```
struct student {
```

```
    char firstName[50];
```

```
    int roll;
```

```
    float marks;
```

```

} s[10];

int main() {

    int i;

    printf("Enter information of students:\n");

    // storing information

    for (i = 0; i < 5; ++i) {

        s[i].roll = i + 1;

        printf("\nFor roll number%d,\n", s[i].roll);

        printf("Enter first name: ");

        scanf("%s", s[i].firstName);

        printf("Enter marks: ");

        scanf("%f", &s[i].marks);

    }

    printf("Displaying Information:\n\n");

    // displaying information

    for (i = 0; i < 5; ++i) {

        if (s[i].marks < 40) {

            printf("\nRoll number: %d\n", i + 1);

            printf("First name: ");

            puts(s[i].firstName);

            printf("Marks: %.1f", s[i].marks);

            printf("\n");

        }

    }

    return 0;

}

```

3. Modify Question 2 to print each student's average marks and the class average (that includes average of all the student's marks).

질문 2를 수정하여 각 학생의 평균 점수와 학급 평균(모든 학생 점수의 평균 포함)을 인쇄하십시오.

```
Enter information of students:
For roll number1,
Enter first name: lee
Enter marks: 40

For roll number2,
Enter first name: kim
Enter marks: 80

For roll number3,
Enter first name: koi
Enter marks: 30

For roll number4,
Enter first name: jio
Enter marks: 20

For roll number5,
Enter first name: ijo
Enter marks: 90
Displaying Information:

Roll number: 3
First name: koi
Marks: 30.0

Roll number: 4
First name: jio
Marks: 20.0
average 52
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

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

```
struct student {
```

```
    char firstName[50];
```

```
    int roll;
```

```
    float marks;
```

```
} s[10];
```

```
int main() {
```

```
    int i;
```

```

int sum = 0;

printf("Enter information of students:\n");

// storing information
for (i = 0; i < 5; ++i) {

    s[i].roll = i + 1;

    printf("\nFor roll number%d,\n", s[i].roll);

    printf("Enter first name: ");

    scanf("%s", s[i].firstName);

    printf("Enter marks: ");

    scanf("%f", &s[i].marks);

}

printf("Displaying Information:\n\n");

// displaying information
for (i = 0; i < 5; ++i) {

    if (s[i].marks < 40) {

        printf("\nRoll number: %d\n", i + 1);

        printf("First name: ");

        puts(s[i].firstName);

        printf("Marks: %.1f", s[i].marks);

        printf("\n");

    }

    sum = sum + s[i].marks;

}

printf("average %d", sum / 5);

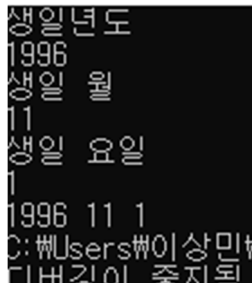
return 0;

}

```

4. Make an array of students as illustrated in Question 1 and write a program to display the details of the student with the given Date of Birth.

문제 1에 설명된 대로 학생 배열을 만들고 주어진 생년월일로 학생의 세부사항을 표시하는 프로그램을 작성한다.



```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
struct Stdent {
```

```
    int number;
```

```
};
```

```
struct Roll_Number {
```

```
    int b;
```

```
};
```

```
struct Name {
```

```
    char Firstname[10];
```

```
    char Middlename[10];
```

```
    char Lastname[10];
```

```
};
```

```
struct sex {
```

```
    int gender;
```

```
};

typedef struct Date_of_birth {

    int day;

    int month;

    int year;

}date;

struct Marks {

    int English;

    int Mathematics;

    int Computer;

};

int main() {

    date d;

    printf("생일 년도 \n");

    scanf("%d", &d.day);

    printf("생일 월 \n");

    scanf("%d", &d.month);

    printf("생일 요일 \n");

    scanf("%d", &d.year);

    printf("%d %d %d", d.day, d.month, d.year);

};
```



5. Write a program to find smallest of three numbers using structures.

구조를 사용하여 세 숫자 중 가장 작은 숫자를 찾는 프로그램을 작성한다.

```
Enter three number 10 20 30
Smallest number is: 10
C:\Users\이상민\source\repos\
디버깅이 중지될 때 콘솔을 자동
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
typedef struct number {
```

```
    int num1;
```

```
    int num2;
```

```
    int num3;
```

```
}num;
```

```
int main() {
```

```
    num n;
```

```
    printf("Enter three number ");
```

```
    scanf("%d %d %d", &n.num1, &n.num2, &n.num3);
```

```
    if (n.num1 < n.num2) {
```

```
        if (n.num1 < n.num3) {
```

```
            printf("\nSmallest number is: %d\n", n.num1);
```

```
        }
```

```
    else {
```

```
        printf("\nSmallest number is: %d\n", n.num3);
```

```
    }
```

```
}
```

```
else {  
    if (n.num2 < n.num3) {  
        printf("\nSmallest number is: %d\n", n.num2);  
    }  
    else {  
        printf("\nSmallest number is: %d\n", n.num3);  
    }  
}  
}
```

6. Write a program to calculate the distance between the given points (6,3) and (2,2).

주어진 점(6,3)과 (2,2) 사이의 거리를 계산하는 프로그램을 작성한다.

```
Input x1: 6
Input y1: 3
Input x2: 2
Input y2: 2
Distance between the said points: 4.1231

C:\Users\이상민\source\repos\Project1\Debug\P
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
#include <math.h>
```

```
int main() {
```

```
    float x1, y1, x2, y2, gdistance;
```

```
    printf("Input x1: ");
```

```
    scanf("%f", &x1);
```

```
    printf("Input y1: ");
```

```
    scanf("%f", &y1);
```

```
    printf("Input x2: ");
```

```
    scanf("%f", &x2);
```

```
    printf("Input y2: ");
```

```
    scanf("%f", &y2);
```

```
    gdistance = ((x2 - x1) * (x2 - x1)) + ((y2 - y1) * (y2 - y1));
```

```
    printf("Distance between the said points: %.4f", sqrt(gdistance));
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```

7. Write a program to read and display the information about all the employees in a department.  
 Edit the details of the  $i^{th}$  employee and redisplay the information.

부서에 있는 모든 직원의 정보를 읽고 표시하는 프로그램을 작성한다.  $i$ 번째 직원의 세부사항을 편집하고 정보를 다시 표시하십시오.

```
Enter details :
Name ? : lee
ID ? : 15135
Salary ? : 234412

Enter details :
Name ? : ID ? : kim
Salary ? :
Enter details :
Name ? : ID ? : 213421
Salary ? : 1234

Entered detail is: Name: kimId: 213421Salary: 1234.000000
Entered detail is: Name: kimId: 213421Salary: 1234.000000
Entered detail is: Name: kimId: 213421Salary: 1234.000000
value change
Entered detail is: Name: leeld: 1000Salary: 1234.000000
Entered detail is: Name: leeld: 1000Salary: 1234.000000
Entered detail is: Name: leeld: 1000Salary: 1234.000000

C:\Users\이상민\source\repos\Project1\Debug\Project1.exe(9592)
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]->[
디버깅 옵션]
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

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

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

```
#pragma warning(disable : 4996)
```

```
/*structure declaration*/
```

```
struct employee {
```

```
    char    name[30];
```

```
    int     empId;
```

```

        float    salary;

};

int main()
{
    /*declare structure variable*/

    struct employee emp;

    /*read employee details*/
    for (int i = 0; i < 3; i++) {

        printf("\nEnter details :\n");

        printf("Name ?.");

        gets(emp.name);

        printf("ID ?.");

        scanf("%d", &emp.empId);

        printf("Salary ?.");

        scanf("%f", &emp.salary);

    }

    /*print employee details*/
    for (int i = 0; i < 3; i++) {

        printf("\nEnter detail is:");

        printf("Name: %s", emp.name);

        printf("Id: %d", emp.empId);

        printf("Salary: %f\n", emp.salary);

    }

```

```

printf("value change");

strcpy(emp.name, "lee");

emp.empld = 1000;


for (int i = 0; i < 3; i++) {

    printf("\nEntered detail is:");

    printf("Name: %s", emp.name);

    printf("Id: %d", emp.empld);

    printf("Salary: %f\n", emp.salary);

}

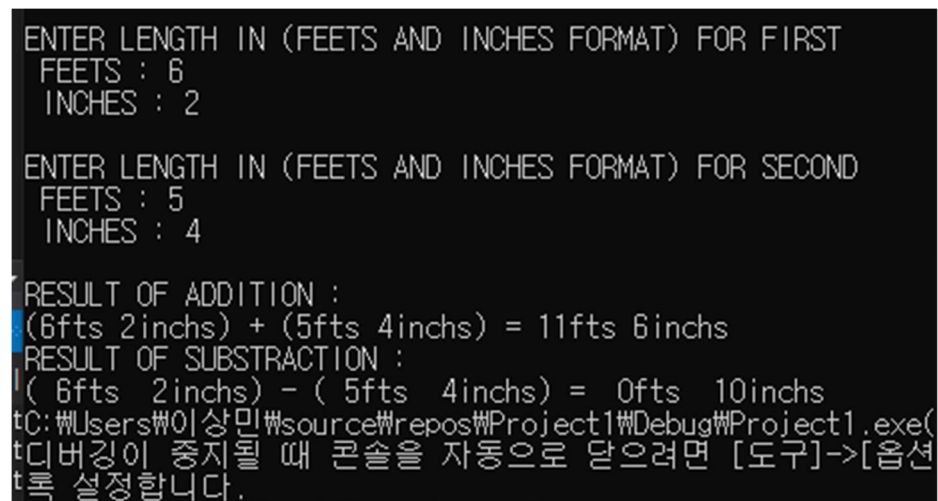
return 0;

}

```

8. Write a program to add and subtract height 6'2" and 5'4".

키 6'2"와 5'4"를 더하고 빼는 프로그램을 쓴다.



```

ENTER LENGTH IN (FEETS AND INCHES FORMAT) FOR FIRST
FEETS : 6
INCHES : 2

ENTER LENGTH IN (FEETS AND INCHES FORMAT) FOR SECOND
FEETS : 5
INCHES : 4

RESULT OF ADDITION :
(6fts 2inches) + (5fts 4inches) = 11fts 6inches
RESULT OF SUBTRACTION :
( 6fts 2inches) - ( 5fts 4inches) = 0fts 10inches
C:\Users\이상민\source\repos\Project1\Debug\Project1.exe(
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]
를 설정합니다.

```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
struct LENGTH
```

```

{
    int FEET;

    int INCH;

};

void add_length(struct LENGTH, struct LENGTH);

void sub_length(struct LENGTH, struct LENGTH);

void main()
{
    int I;

    struct LENGTH A, B;

    printf("\nEnter LENGTH IN (FEETS AND INCHES FORMAT) FOR FIRST\n");

    printf("FEETS : ");

    scanf("%d", &A.FEET);

    printf("INCHES : ");

    scanf("%d", &A.INCH);

    printf("\nEnter LENGTH IN (FEETS AND INCHES FORMAT) FOR SECOND\n");

    printf("FEETS : ");

    scanf("%d", &B.FEET);

    printf("INCHES : ");

    scanf("%d", &B.INCH);

    add_length(A, B);

    sub_length(A, B);

}

void add_length(struct LENGTH A, struct LENGTH B) {

    struct LENGTH C;

    int INCHS;

```

```

    INCHS = (A.FEET * 12 + A.INCH) + (B.FEET * 12 + B.INCH);

    C.FEET = INCHS / 12;

    C.INCH = INCHS % 12;

    printf("\nRESULT OF ADDITION :\n");

    printf("(%dfts %dinchs) + (%dfts %dinchs) = %dfts %dinchs", A.FEET, A.INCH, W

        B.FEET, B.INCH, C.FEET, C.INCH);

}

void sub_length(struct LENGTH A, struct LENGTH B)

{

    struct LENGTH C;

    int INCHS;

    INCHS = (A.FEET * 12 + A.INCH) - (B.FEET * 12 + B.INCH);

    C.FEET = INCHS / 12;

    C.INCH = INCHS % 12;

    printf("\nRESULT OF SUBTRACTION : \n");

    printf("(% dfts % dinchs) - (% dfts % dinchs) = % dfts % dinchs", A.FEET, A.INCH, W

        B.FEET, B.INCH, C.FEET, C.INCH);

}

```



9. Write a program to add and subtract 10hrs 20mins 50sec and 5hrs 30min 40sec.

10시간 20분 50초와 5시간 30분 40초를 더하고 빼는 프로그램을 쓴다.

```
Enter start time:
Enter hours, minutes and seconds: 10 20 50
Enter stop time:
Enter hours, minutes and seconds: 5 30 40

TIME DIFFERENCE: 10:20:50 - 5:30:40 = 4:50:10

TIME SUM: 10:20:50 + 5:30:40 = 15:51:30

C:\Users\이상민\source\repos\Project1\Debug\Project1.exe
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]을 설정합니다
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
struct TIME {
```

```
    int seconds;
```

```
    int minutes;
```

```
    int hours;
```

```
};
```

```
void Subtracttime(struct TIME t1, struct TIME t2, struct TIME* diff);
```

```
void Addtime(struct TIME start, struct TIME stop, struct TIME* diff);
```

```
int main() {
```

```
    struct TIME startTime, stopTime, diff;
```

```
    printf("Enter start time: \n");
```

```
    printf("Enter hours, minutes and seconds: ");
```

```
    scanf("%d %d %d", &startTime.hours, &startTime.minutes, &startTime.seconds);
```

```
    printf("Enter stop time: \n");
```

```
    printf("Enter hours, minutes and seconds: ");
```

```
scanf("%d %d %d", &stopTime.hours, &stopTime.minutes, &stopTime.seconds);
```

```
Subtracttime(startTime, stopTime, &diff);
```

```
printf("\nTIME DIFFERENCE: %d:%d:%d - ", startTime.hours, startTime.minutes, startTime.seconds);
```

```
printf("%d:%d:%d ", stopTime.hours, stopTime.minutes, stopTime.seconds);
```

```
printf("= %d:%d:%d\n", diff.hours, diff.minutes, diff.seconds);
```

```
Addtime(startTime, stopTime, &diff);
```

```
printf("\nTIME SUM: %d:%d:%d + ", startTime.hours, startTime.minutes, startTime.seconds);
```

```
printf("%d:%d:%d ", stopTime.hours, stopTime.minutes, stopTime.seconds);
```

```
printf("= %d:%d:%d\n", diff.hours, diff.minutes, diff.seconds);
```

```
return 0;
```

```
}
```

```
void Subtracttime(struct TIME start, struct TIME stop, struct TIME* diff) {
```

```
    if (stop.seconds > start.seconds) {
```

```
        --start.minutes;
```

```
        start.seconds += 60;
```

```
    }
```

```
    diff->seconds = start.seconds - stop.seconds;
```

```
    if (stop.minutes > start.minutes) {
```

```
        --start.hours;
```

```
        start.minutes += 60;
```

```
    }
```

```
    diff->minutes = start.minutes - stop.minutes;
```

```

        diff->hours = start.hours - stop.hours;
    }

void Addtime(struct TIME start, struct TIME stop, struct TIME* diff) {

    if (stop.seconds + start.seconds >= 60) {

        ++start.minutes;

        start.seconds -= 60;

    }

    diff->seconds = start.seconds + stop.seconds;

    if (stop.minutes + start.minutes >= 60) {

        ++start.hours;

        start.minutes -= 60;

    }

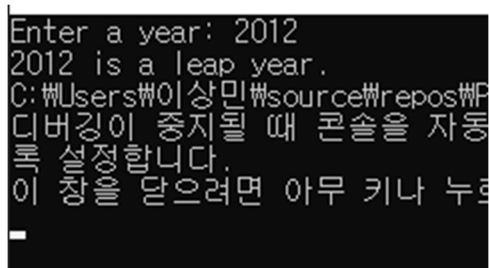
    diff->minutes = start.minutes + stop.minutes;

    diff->hours = start.hours + stop.hours;

}

```

10. Write a program using structure to check if the current year is leap year or not.  
구조를 이용하여 프로그램을 작성하여 당해 연도가 윤년인지 아닌지 확인한다.



```

Enter a year: 2012
2012 is a leap year.
C:\Users\이상민\source\repos\
디버깅이 중지될 때 콘솔을 자동
로 설정합니다.
이 창을 닫으려면 아무 키나 누르
_

```

```

#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>

typedef struct year {

    int year;

```

```
}year;

int main() {

    year y;

    printf("Enter a year: ");

    scanf("%d", &y.year);

    if (y.year % 4 == 0) {

        if (y.year % 100 == 0) {

            if (y.year % 400 == 0)

                printf("%d is a leap year.", y.year);

            else

                printf("%d is not a leap year.", y.year);

        }

        else

            printf("%d is a leap year.", y.year);

    }

    else

        printf("%d is not a leap year.", y.year);

    return 0;

}
```

11. Write a program using pointer to structure to initialize the members of an employee structure. Use functions to print the employee's information.

직원 구조의 구성원을 초기화하기 위해 포인터를 사용하여 프로그램을 작성한다. 기능을 사용하여 직원의 정보를 인쇄하십시오.

```
Enter age: 25
Enter weight: 45
Displaying:
Age: 25
weight: 45.000000
C:\Users\W01상민\source\repos\
디버깅이 중지될 때 콘솔을 자동
로 설정합니다.
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
struct employee
```

```
{
```

```
    int age;
```

```
    float weight;
```

```
};
```

```
int main()
```

```
{
```

```
    struct employee *employeePtr, employee1;
```

```
    employeePtr = &employee1;
```

```
    printf("Enter age: ");
```

```
    scanf("%d", &employeePtr->age);
```

```
    printf("Enter weight: ");
```

```
    scanf("%f", &employeePtr->weight);
```

```
    printf("Displaying:\n");
```

```
    printf("Age: %d\n", employeePtr->age);
```

```
    printf("weight: %f", employeePtr->weight);
```

```
    return 0;
```

```
}
```

12. Write a program to create a structure with the information given below. Then, read and print the data.

아래와 같은 정보로 구조를 만드는 프로그램을 작성한다. 그런 다음 데이터를 읽고 인쇄하십시오.

Employee[10]

- (a) Emp\_Id
- (b) Name
  - (i) First Name
  - (ii) Middle Name
  - (iii) Last Name
- (c) Address
  - (i) Area
  - (ii) City
  - (iii) State
- (d) Age
- (e) Salary
- (f) Designation

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
struct employee{
```

```
    int emp_id;
```

```
    int age;
```

```
    double salary;
```

```
    int designation;
```

```
};
```

```
struct name {
```

```

char FirstName[10];

char MiddleName[10];

char LastName[10];

};

struct Address {

    char Area[10];

    char City[10];


    char State[10];

};

```

13. Define a structure date containing three integers— day, month, and year. Write a program using functions to read data, to validate the date entered by the user and then print the date on the screen. For example, if you enter 29,2,2010 then that is an invalid date as 2010 is not a leap year. Similarly 31,6,2007 is invalid as June does not have 31 days.

일, 월, 년 등 세 개의 정수를 포함하는 구조 날짜를 정의한다. 사용자가 입력한 날짜를 확인하고 화면에 날짜를 인쇄하기 위해 기능을 사용하여 프로그램을 작성한다. 예를 들어, 29,2,2010을 입력하면 2010년은 윤년이 아니므로 그 날짜는 유효하지 않다. 마찬가지로 6월에도 31일이 없기 때문에 31,6,2007년은 무효다.



```

Enter year month day : 2010 2 29
Date is invalid.2010 2 29 is not a leap year.
C:\Users\이상민\source\repos\Project1\Debug\Project1
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->
로 설정합니다

```

```

#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>

void Leapyear(int year, int month, int day);

void Checkdate(int year, int month, int day);

typedef struct Date {

    int year;

    int month;

    int day;

```

```
}date;
```

```
void Leapyear(int year, int month, int day) {  
    if (year % 4 == 0) {  
        if (year % 100 == 0) {  
            if (year % 400 == 0)  
                printf("%d %d %d is a leap year.", year, month, day);  
            else  
                printf("%d %d %d is not a leap year.", year, month, day);  
        }  
        else  
            printf("%d %d %dis a leap year.", year, month, day);  
    }  
    else  
        printf("%d %d %d is not a leap year.", year, month, day);  
}
```

```
void Checkdate(int year, int month, int day) {  
    int is_leap = 0, is_valid = 1;  
    if (year >= 1800 && year <= 9999)  
    {  
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))  
        {  
            is_leap = 1;  
        }  
        if (month >= 1 && month <= 12)  
        {
```



```
    if (month == 2)
    {
        if (is_leap && day == 29)
        {
            is_valid = 1;
        }
        else if (day > 28)
        {
            is_valid = 0;
        }
    }
    else if (month == 4 || month == 6 || month == 9 || month == 11)
    {
        if (day > 30)
        {
            is_valid = 0;
        }

        else if (day > 31)
        {
            is_valid = 0;
        }
    }
    else
        is_valid = 0;
}
```

```
        else

            is_valid = 0;

        if (is_valid)

            printf("Date is valid.");

        else

            printf("Date is invalid.");

    }

}

int main() {

    date d;

    printf("Enter year month day : ");

    scanf("%d %d %d", &d.year, &d.month, &d.day);

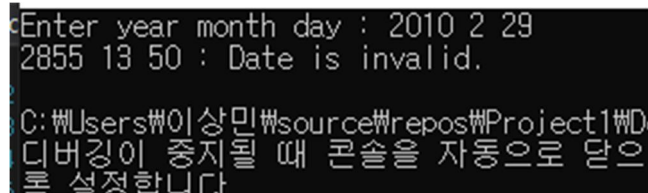
    Checkdate(d.year, d.month, d.day);

    Leapyear(d.year, d.month, d.day);

}
```

14. Using the structure definition of the above program, write a function to increment the date. Make sure that the incremented date is a valid date.

상기 프로그램의 구조 정의를 이용하여 날짜를 증분하는 함수를 작성한다. 증분 날짜가 유효한 날짜인지 확인하십시오.



```
Enter year month day : 2010 2 29
2855 13 50 : Date is invalid.

C:\Users\이상민\source\repos\Project1\WD
디버깅이 중지될 때 콘솔을 자동으로 닫으
로 설정합니다
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
#include <time.h>
```

```
void Incrementdate(int year, int month, int day);
```

```
typedef struct Date {
```

```
    int year;
```

```
    int month;
```

```
    int day;
```

```
}date;
```

```
void Incrementdate(int year, int month, int day) {
```

```
    int is_leap = 0, is_valid = 1;
```

```
    srand(time(NULL));
```

```
    year += (rand() % 1000) + 1;
```

```
    month += (rand() % 12) + 1;
```

```
    day += (rand() % 31) + 1;
```

```
    printf("%d %d %d : ", year, month, day);
```

```
    if (year >= 1800 && year <= 9999)
```

```
{
```

```
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
```

```
{  
  
    is_leap = 1;  
  
}  
  
if (month >= 1 && month <= 12) {  
  
    is_valid = 1;  
  
}  
  
else  
  
    is_valid = 0;  
  
  
if (month == 2)  
{  
  
    if (is_leap && day == 29)  
    {  
  
        is_valid = 1;  
  
    }  
  
    else if (day > 28)  
    {  
  
        is_valid = 0;  
  
    }  
  
}  
  
else if (month == 4 || month == 6 || month == 9 || month == 11)  
{  
  
    if (day > 30)  
    {  
  
        is_valid = 0;  
  
    }  
  
}
```

```

        else if (day > 31)
        {
            is_valid = 0;
        }
    }
    else
        is_valid = 0;
}

if (is_valid == 1)
    printf("Date is valid.\n");
else
    printf("Date is invalid.\n");
}

int main() {
    date d;

    printf("Enter year month day : ");

    scanf("%d %d %d", &d.year, &d.month, &d.day);

    Incrementdate(d.year, d.month, d.day);
}

```

15. Modify the above program to add a specific number of days to the given date.

상기 프로그램을 수정하여 지정된 날짜에 특정 일수를 추가한다.

```
Enter Date : 2010 2 29
Add X : 30
2 9 34
C:\Users\이상민\source\repos\Project1
디버깅이 중지될 때 콘솔을 자동으로 닫
록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
#include <stdbool.h>
```

```
bool isLeap(int y);
```

```
int offsetDays(int d, int m, int y);
```

```
void revoffsetDays(int offset, int y, int* d, int* m);
```

```
void addDays(int d1, int m1, int y1, int x);
```

```
bool isLeap(int y)
```

```
{
```

```
    if (y % 100 != 0 && y % 4 == 0 || y % 400 == 0)
```

```
        return true;
```

```
    else
```

```
        return false;
```

```
}
```

```
int offsetDays(int d, int m, int y)
```

```
{
```

```
    int offset = d;
```

```
    switch (m - 1)
```

```
{  
  
case 11:  
    offset += 30;  
  
case 10:  
    offset += 31;  
  
case 9:  
    offset += 30;  
  
case 8:  
    offset += 31;  
  
case 7:  
    offset += 31;  
  
case 6:  
    offset += 30;  
  
case 5:  
    offset += 31;  
  
case 4:  
    offset += 30;  
  
case 3:  
    offset += 31;  
  
case 2:  
    offset += 28;  
  
case 1:  
    offset += 31;  
  
}
```

```
if (isLeap(y) && m > 2)
```

```

        offset += 1;

    return offset;
}

void revoffsetDays(int offset, int y, int* d, int* m)
{
    int month[13] = { 0, 31, 28, 31, 30, 31, 30,
                     31, 31, 30, 31, 30, 31 };

    if (isLeap(y))
        month[2] = 29;

    int i;
    for (i = 1; i <= 12; i++)
    {
        if (offset <= month[i])
            break;

        offset = offset - month[i];
    }

    *d = offset;
    *m = i;
}

void addDays(int d1, int m1, int y1, int x)
{
    int offset1 = offsetDays(d1, m1, y1);

    int remDays = isLeap(y1) ? (366 - offset1) : (365 - offset1);

    int y2, offset2;

```



```

    if (x <= remDays)
    {
        y2 = y1;

        offset2 = offset1 + x;
    }

    else
    {
        x -= remDays;

        y2 = y1 + 1;

        int y2days = isLeap(y2) ? 366 : 365;

        while (x >= y2days)
        {
            x -= y2days;

            y2++;

            y2days = isLeap(y2) ? 366 : 365;
        }

        offset2 = x;
    }

    int m2, d2;

    revoffsetDays(offset2, y2, &d2, &m2);

    printf("%d %d %d  ", d2, m2, y2);
}

int main()
{

```

```

    int y, m, d, x;

    printf("Enter Date : ");

    scanf("%d %d %d", &d, &m, &y);

    printf(" Add X : ");

    scanf("%d", &x);

    addDays(d, m, y, x);

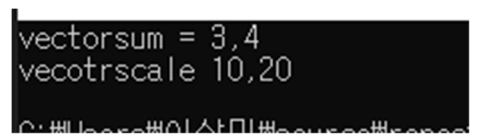
    return 0;

}

```

16. Write a program to define a structure vector. Then write functions to read data, print data, add two vectors and scale the members of a vector by a factor of 10.

구조 벡터를 정의하는 프로그램을 작성한다. 그런 다음 데이터를 읽고, 데이터를 인쇄하고, 벡터를 두 개 추가하고, 벡터의 구성원을 10배만큼 스케일링한다.



```

vectorsum = 3,4
vecotrscale 10,20
C:\Users\user>

```

```

#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>

void Vectordadd(int x1, int x2, int y1, int y2);

int Vectorscale(int x1, int y1, int scale);

typedef struct vector {

    int x1;

    int x2;

    int y1;

    int y2;

}vector;

```

```

void Vectordadd(int x1, int x2, int y1, int y2) {

    printf("vectorsum = %d,%d\n", x1 + x2, y1 + y2);

    return 0;

}

int Vectorscale(int x1, int y1, int scale) {

    x1 = x1 * scale;

    y1 = y1 * scale;

    printf("vecotrscale %d,%d \n", x1, y1);

}

int main() {

    vector v;

    v.x1 = 1;

    v.y1 = 2;

    v.x2 = 1;

    v.y2 = 3;

    Vectordadd(v.x1, v.y1, v.x2, v.y2);

    Vectorscale(v.x1, v.y1, 10);

    return 0;

}

```

17. Write a program to define a structure for a hotel that has members— name, address, grade, number of rooms, and room charges. Write a function to print the names of hotels in a particular grade. Also write a function to print names of hotels that have room charges less than the specified value.

이름, 주소, 등급, 객실 수, 객실 요금 등 회원을 가진 호텔의 구조를 정의하는 프로그램을 작성한다. 특정 등급의 호텔 이름을 인쇄하는 기능을 작성한다. 또한 객실료가 지정된 값보다 적은 호텔의 이름을 인쇄하는 기능도 작성한다.

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
#include<stdlib.h>
```

```
struct hotel {
```

```
    char name[20];
```

```
    char city[10];
```

```
    char grade[10];
```

```
    int rc, nr;
```

```
};
```

```
int main() {
```

```
    struct hotel ht[20], t;
```

```
    int i, n, j, c;
```

```
    char gr;
```

```
    printf("enter no.of hotelsn :");
```

```
    scanf("%d", &n);
```

```
    for (i = 0; i < n; i++)
```

```
    {
```

```
        printf("enter name of hotel n : ");
```

```
        scanf("%s", &ht[i].name);
```

```

        printf("enter name of city n : ");

        scanf("%s", &ht[i].city);

        printf("enter the grade n : ");

        scanf("%s", &ht[i].grade);

        printf("enter room charge n : ");

        scanf("%d", &ht[i].rc);

        printf("enter no of rooms n : ");

        scanf("%d", &ht[i].nr);

    }

    for (i = 0; i < n; i++)

        for (j = 0; j < n - i; j++)

            {

                t = ht[j];

                ht[j] = ht[j + i];

                ht[j + 1] = t;

            }

    printf("enter a grade to print the hotels n\n");

    gr = getchar();

    printf("hotel name / city / grade / roomcharge no of room\n");

    for (i = 0; i < n; i++)

        if (gr == ht[i].grade)

            printf("%s / %s / %s / %d / %d \n", ht[i].name, ht[i].city, ht[i].grade, ht[i].rc,

ht[i].nr);

    printf("enter a room charge to print hotels less than given charge n : ");

    scanf("%d", &c);

```

```
printf("\n");

printf("hotel name  / city / grade / roomcharge no of rooms\n");

for (i = 0; i < n; i++)

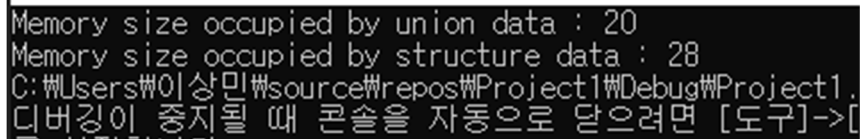
    if (c >= ht[i].rc)

        printf("%s / %s / %s / %d / %d /", ht[i].name, ht[i].city, ht[i].grade, ht[i].rc,
ht[i].nr);

}
```

18. Write a program to define a union and a structure both having exactly the same members. Using the sizeof operator, print the size of structure variable as well as union variable and comment on the result.

조합과 조합을 정의하는 프로그램을 작성하라. 측정 시스템의 크기를 사용하여 구조 변수의 크기 및 조합 변수를 인쇄하고 결과에 대해 설명하십시오.



```
Memory size occupied by union data : 20
Memory size occupied by structure data : 28
C:\Users\이상민\source\repos\Project1\Debug\Project1.exe
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[
```

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

```
union uData {
```

```
    int i;
```

```
    float f;
```

```
    char str[20];
```

```
};
```

```
struct sData {
```

```
    int i;
```

```
    float f;
```

```
    char str[20];
```

```
};
```

```
int main() {
```

```
    union uData uData;
```

```
    struct sData sData;
```

```
    printf("Memory size occupied by union data : %d \n", sizeof(uData));
```

```
    printf("Memory size occupied by structure data : %d", sizeof(sData));
```

```
    return 0;
```

```
}
```

19. Declare a structure time that has three fields—hr, min, sec. Create two variables start\_time and end\_time. Input their values from the user. Then while start\_time does not reach the end\_time, display GOOD DAY on the screen.

3개의 필드(hr, min, sec)가 있는 구조 시간을 선언한다. start\_time 및 end\_time 변수 두 개를 생성하십시오. 사용자로부터 값을 입력하십시오. 그런 다음 start\_time이 end\_time에 도달하지 않는 동안 화면에 GOOD DAY를 표시하십시오.

```
-----Enter START TIME-----
20 30 40
-----Enter END TIME-----
25 35 45
GOOD DAY
C:\Users\상민\source\repos
디버깅이 중지될 때 콘솔을 자
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
typedef struct time {
```

```
    char hour[10];
```

```
    char min[10];
```

```
    char sec[10];
```

```
}time;
```

```
int main() {
```

```
    time t;
```

```
    printf("-----Enter START TIME-----\n");
```

```
    scanf("%s %s %s", &t.hour[0], &t.min[0], &t.sec[0]);
```

```
    printf("-----Enter END TIME-----\n");
```

```
    scanf("%s %s %s", &t.hour[1], &t.min[1], &t.sec[1]);
```

```
    if (t.hour[0] == t.hour[1] && t.min[0] == t.min[1] && t.sec[0] == t.sec[1])
```

```
        printf("GOOD DAY");
```

```
    else
```

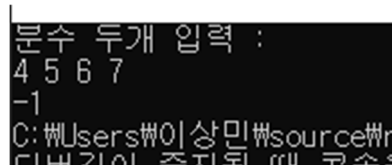
```
        printf("NO");
```

```
}
```



20. Declare a structure fraction that has two fields— numerator and denominator. Create two variables and compare them using function. Return 0 if the two fractions are equal, -1 if the first fraction is less than the second and 1 otherwise. You may convert a fraction into a floating point number for your convenience.

분자와 분모 두 개의 필드가 있는 구조 분수를 선언한다. 두 개의 변수를 만들어 함수를 사용하여 비교한다. 두 분율이 같으면 0을 반환하고, 첫 번째 분율이 두 번째 분율보다 작으면 -1을 반환하고, 그렇지 않으면 1을 반환한다. 편의를 위해 분수 값을 부동 소수점 번호로 변환할 수 있다.



```
분수 두개 입력 :
4 5 6 7
-1
C:\Users\이상민\source\프로그래밍\제20과제\제20과제.c
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
typedef struct fraction {
```

```
    float numerator;
```

```
    float denominator;
```

```
}fraction;
```

```
int main() {
```

```
    fraction f1, f2;
```

```
    printf("분수 두개 입력 :%n");
```

```
    scanf("%f %f %f %f", &f1.numerator, &f1.denominator, &f2.numerator, &f2.denominator);
```

```
    if (f1.numerator / f1.denominator > f2.numerator / f2.denominator)
```

```
        printf("1");
```

```
    else if (f1.numerator / f1.denominator == f2.numerator / f2.denominator)
```

```
        printf("0");
```

```
    else
```

```
        printf("-1");
```

```
}
```

21. Declare a structure POINT. Input the coordinates of a point variable and determine the quadrant in which it lies. The following table can be used to determine the quadrant

구조 POINT를 선언한다. 점 변수의 좌표를 입력하고 그것이 놓여 있는 사분면을 결정한다. 다음 표를 사용하여 사분면을 결정할 수 있다.

Quadrant	X	Y
1	Positive	Positive
2	Negative	Positive
3	Negative	Negative
4	Positive	Negative

```
Enter x point y point
2 3
1
C:\Users\이상민\source\repos\Pro
디버깅이 중지될 때 콘솔을 자동으
록 설정합니다.
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
typedef struct POINT {
```

```
    int x;
```

```
    int y;
```

```
}point;
```

```
int main() {
```

```
    point p;
```

```
    printf("Enter x point y point \n");
```

```
    scanf("%d %d", &p.x, &p.y);
```

```
    if (p.x >= 0 && p.y >= 0)
```

```
        printf("1");
```

```
    else if (p.x < 0 && p.y >= 0)
```

```
        printf("2");
```

```
    else if (p.x < 0 && p.y < 0)
```

```
        printf("3");
```

```
    else if (p.x >= 0 && p.y < 0)
```

```
        printf("4");
```

```
}
```

22. Write a program to calculate the area of one of the geometric figures—circle, rectangle or a triangle. Write a function to calculate the area. The function must receive one parameter which is a structure that contains the type of figure and the size of the components needed to calculate the area must be a part of a union. Note that a circle requires just one component, rectangle requires two components and a triangle requires the size of three components to calculate the area.

원형, 직사각형 또는 삼각형 등 기하학적 그림 중 하나의 영역을 계산하는 프로그램을 작성한다. 함수를 작성하여 면적을 계산한다. 함수는 그림의 유형과 면적 계산에 필요한 구성 요소의 크기를 포함하는 하나의 매개변수를 수신해야 한다. 원에는 한 개의 성분만 있고, 직사각형은 두 개의 성분이 필요하며, 삼각형은 면적을 계산하기 위해 세 개의 성분 크기가 필요하다.

```
삼각형 면적을 위한 세 변수입력 : 10 20 30
삼각형 면적 : 0.000
원 면적을 위한 하나의 변수 입력 : 20
원 면적 : 1256.000
직사각형 면적을 위한 두 변수입력 : 20 20
직사각형 면적 : 400.000

C:\Users\이상민\source\repos\Project1\Debug\Project1
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
#include<math.h>
```

```
double trianglearea(double a, double b, double c);
```

```
double circlearea(double radius);
```

```
double rectanglearea(double width, double height);
```

```
int main() {
```

```
    double a, b, c;
```

```
    double triangle_area, circle_area, rectangle_area;
```

```
    printf("삼각형 면적을 위한 세 변수입력 : ");
```

```
    scanf("%lf %lf %lf", &a, &b, &c);
```

```
    triangle_area = trianglearea(a, b, c);
```

```
printf("삼각형 면적 : %.3lf\n", triangle_area);
```

```
printf("원 면적을 위한 하나의 변수 입력 : ");
```

```
scanf("%lf", &a);
```

```
circle_area = circlearea(a);
```

```
printf("원 면적 : %.3lf\n", circle_area);
```

```
printf("직사각형 면적을 위한 두 변수입력 : ");
```

```
scanf("%lf %lf", &a, &b);
```

```
rectangle_area = rectanglearea(a, b);
```

```
printf("직사각형 면적 : %.3lf\n", rectangle_area);
```

```
return 0;
```

```
}
```

```
double rectanglearea(double width, double height) {
```

```
    return width * height;
```

```
}
```

```
double circlearea(double radius) {
```

```
    return radius * radius * 3.14;
```

```
}
```

```
double trianglearea(double a, double b, double c) {
```

```
    double l = (a + b + c) / 2;
```

```
    return sqrt(l*(l - a)*(l - b)*(l - c));
```

```
}
```