

Homework #1

1. 두 개의 정수를 입력 받아 두 수의 공약수와 최소 공배수를 아래와 같이 출력되게 하시오.

```
C:\W > hw1_1.exe 16 24
두 수의 공약수: 1 2 4 8
최소공배수: 48
```

2. 아래와 같은 표가 있다. 표에 들어가 있는 숫자들에 있어 7로 나누어지는 부분을 @로 바꾸고, 나누어지지 않는 부분을 공백(' ')으로 바꿔서 이를 2차원 행렬에 넣어서 출력하시오.

77	28	31	392	224	84	161
60	84	53	476	32	255	7
515	147	76	455	12	243	21
9	224	141	287	56	42	441
87	161	142	224	8	9	231
243	21	23	35	111	87	357
255	7	90	70	63	224	147

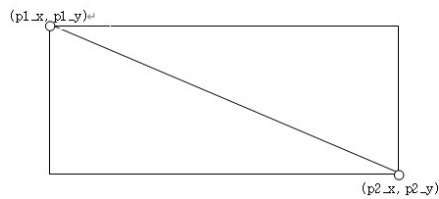
```
C:\W > hw1_1.exe
@@ @@@@
@ @ @
....

int main() {
    int a[7][7] = { {77, 28, 31, 392, 224, 84, 161}, ... };
    char b[7][7]; // 조건에 맞춰 문자를 입력 받을 char형식 2차원 배열

    // 조건에 맞춰 이차원 배열 b에 조건에 맞는 문자값 삽입

    // 출력
}
```

3. 좌표상의 2점의 정보를 가지고 있는 직사각형 class를 만들려고 한다. 서로의 점은 직사각형에서 대각을 이루어지게 되어있다고 가정한다.



직사각형 class 내에는 직사각형의 넓이를 구하는 멤버 함수와 직사각형 내의 대각선 길이를 구하는 멤버 함수가 포함되어 있다. 이와 같은 조건을 만족시키는 class를 가지고 아래 main() 함수를 작성하시오. (코드를 .h과 .cpp 파일을 분리하여 작성하시오.)

// Rect.h 직사각형 클래스 헤더파일

```
#ifndef RECT_H
#define RECT_H

class Rect {
private:
    float p1_x, p2_x, p1_y, p2_y;

public:
    Rect();
    Rect(float x1, float x2, float y1, float y2);
    float get_area(); // 직사각형의 넓이를 구하는 함수
    float get_diagonal_length(); // 직사각형의 내부 대각선 길이 구하는 함수
}

#endif
```

// main.cpp 구현부

```
#include "Rect.h"

int main() {
    float x1, x2, y1, y2; // 입력 받는 2개의 좌표정보, 입력 받는 부분은 생략되어있음

    Rect rect1(x1, x2, y1, y2); // 위 정보를 가지고 직사각형 객체 생성

    cout << .... // 직사각형의 넓이 출력
    cout << .... // 직사각형의 내부 대각선 길이 출력
}
```

4. 문자열을 매개변수로 받아, 해당 문자열에서 사용된 단어의 개수(중복된 단어 고려할 필요 X)와 문장의 개수를 카운트해주는 함수를 각각 만들어 아래와 같이 출력되게 하시오. (문장의 조건은 처음 시작은 대문자, 끝은 '.'으로 이루어진 것으로 고려, 문자열은 char* 사용)

```
int count_words(char *s) {  
    // 문자열에 포함된 단어의 개수를 반환해주는 함수  
    ....  
}
```

```
int count_sentences(char *s) {  
    // 문자열에 포함된 문장의 개수를 반환해주는 함수  
    ....  
}
```

- 첫 번째 문자열 -

"Object-oriented programming is a programming paradigm based on the concept of objects, which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods. A feature of objects is that an object's procedures can access and often modify the data fields of the object with which they are associated. In OOP, computer programs are designed by making them out of objects that interact with one another. There is significant diversity of OOP languages, but the most popular ones are class-based, meaning that objects are instances of classes, which typically also determine their type."

- 두 번째 문자열 -

"melon, apple, grape, kiwi, pear, orange"

- 세 번째 문자열 -

" "

C:\W > hw1_2.exe

첫 번째 문자열에 포함된 단어 수: 105

두 번째 문자열에 포함된 단어 수: 6

세 번째 문자열에 포함된 단어 수: 0