# 창의적 소프트웨어 프로그래밍 hw3-3

**제출 기한**

9월 24일 23:59pm 까지

1. GitLab 웹 페이지에서hw3-3라는 폴더를 만들고 터미널에서 git pull
2. hw3-3 디렉토리에 Makefile, array\_2dim.cc, touch\_points.cc cal\_complex.cc를 작성
3. Makefile은 ‘make’명령으로 1~3 숙제가 모두 빌드
4. 최종 버전을 GitLab에 commit
5. **시간과 파일명, 입력과 출력 방식 반드시 지키기**

**과제 1.**

과제 : 2차원 int형 배열에 정수를 6개 입력받아 채우고, 작성 규칙에 따라 배열의 요소를 출력하고, 배열의 각 열의 정수의 합을 출력하는 프로그램 작성

**작성 규칙**

* 선언하는 배열은 행이3개 열이2개인 3x2 int형 배열이어야 함
* 배열의 각 원소들은 사용자로부터 입력 받아야 함.
* 입력 순서대로 1행1열 - 1행2열 – 2행1열 … 순서대로 숫자가 채워져야 함
* 입력이 끝나고 배열이 다 채워지면 다음내용을 차례대로 출력 하도록 함
  1. 각 행을 한 줄에 하나씩 출력(1행, 2행, 3행 순서)
  2. 1열의 모든 원소의 합과 2열의 모든 원소의 합을 차례대로 출력

파일명: array\_2dim (array\_2dim.cc)

입력: 6개의 정수 값

출력: 배열 각 행의 원소들과 각 열의 원소들의 합

|  |
| --- |
| **$./array\_2dim**  **1 2 3 4 5 6**  **1 2**  **3 4**  **5 6**  **1번째 열의 숫자의 합: 9**  **2번째 열의 숫자의 합: 12**  **$** |

**과제 2.**

과제: 3명의 학생의 이름과 각 학생이 각각 선택한 3가지 점의 좌표를 입력 받아 각 학생별로 원점(0,0)에서 가장 거리가 먼 점을 출력하는 프로그램 작성

**작성 규칙**

* 학생이 선택한 점은 아래 Point 구조체를 사용하여 저장한다.

( Point 구조체는 2차원 직교 좌표계의 한 점을 나타낸다 가정한다.)

* 아래 Person 구조체 변수를 3개 선언해야 하며, Person 구조체 변수들의 멤버는 사용자로부터 값을 입력 받아 채운다.
* 학생의 이름은 영어 알파벳으로 최대 6글자까지인 것으로 가정한다
* 입력은 아래 실행예와 같이 학생의 이름과 각 학생이 선택한 세 점의 x, y 실수 좌표값을 순서대로 입력받아야 함
* 입력이 끝나면 아래와 같이 각 학생별로 학생의 이름과 학생이 선택한 점 중 원점 (0,0)으로부터 거리가 가장 먼 점을 출력한다 .

Code skeleton

typedef struct

{

double x;

double y;

} Point;

typedef struct

{

char name[7]

Point points[3];

} Person;

int main(void){

//implement this function

}

파일명: touch\_points (touch\_points.cc)

입력: 7글자 미만 알파벳 문자열(이름)과 6개의 실수(각 학생이 선택한 점)를 차례대로 3번 입력

출력: 입력 받은 학생의 이름과 학생이 선택한 점 중 원점에서 가장 먼 점을 학생별로 출력

|  |
| --- |
| **$./touch\_points**  **John 1 2 3 4 5 6**  **Amy 1 1 -2 -1 0.1 0.1**  **Emma 1 1 2 2 10 1**  **John (5, 6)**  **Amy (-2, -1)**  **Emma (10, 1)**  **$** |

**과제 3**

과제: 사용자로부터 4개의 실수를 입력 받아 2개의 복소수를 만든 후, 두 복소수의 합과 곱을 출력하는 프로그램을 작성.

**배경 지식**

* 복소수는 a+bi와 같은 형태로 나타낸다.

( a를 복소수의 실수부, 실수 b를 복소수의 허수부라 부름)

* 복소수의 합은 (a+bi) + (c+di) = (a+c) + (b+d)i 와 같다.
* 복소수의 곱은 (a+bi) \* (c+di) = ac - bd + (ad+bc)i와 같다.

**작성 규칙**

* 복소수(complex number)를 나타내는 구조체를 직접 정의한다.
* printComplex() 함수는 하나의 복소수 구조체를 인자로 받아, 해당 복소수를 출력하는 함수이다.
* printAddComp() 함수는 두개의 복소수 구조체를 인자로 받아, 두 복소수의 합을 출력하는 함수이다.
* printMulComp() 함수는 두개의 복소수 구조체를 인자로 받아, 두 복소수의 곱을 출력하는 함수이다.
* main(), printAddComp(), printMulComp() 세 함수에서 복소수 값을 출력할 때는 반드시 printComplex() 함수를 호출하여 printComplex()내에서 복소수를 출력해야 한다.
* printAddComp(), printMulComp(), printComplex() 함수는 직접 선언하고 정의한다.
* 프로그램은 사용자로부터 4개의 실수를 입력 받아야 함
* 입력받은 실수 값은 입력 순서대로 복소수 구조체1의 실수부와 허수부, 복소수 구조체2의 실수부와 허수부에 할당한다.
* 입력이 끝나고 복소수 구조체를 다 채운 후 다음 내용을 차례로 출력한다.
  1. 입력 받은 두 복소수의 값을 차례대로 출력
  2. 두 복소수의 합을 출력
  3. 두 복소수의 곱을 출력

파일명: cal\_complex (cal\_complex.cc)

입력: 실수 4개(2개의 복소수의 실수부와 허수부)

출력: 입력 받은 각 복소수와 입력 받은 복소수의 합과 차를 출력

**$./cal\_comlex**

**1.0 2.0 3.0 4.0**

**복소수1:**

**1.000000 + 2.000000i**

**복소수2:**

**3.000000 + 4.000000i**

**두 복소수의 합:**

**4.000000 + 6.000000i**

**두 복소수의 곱:**

**-5.000000 + 10.000000i**

**$**