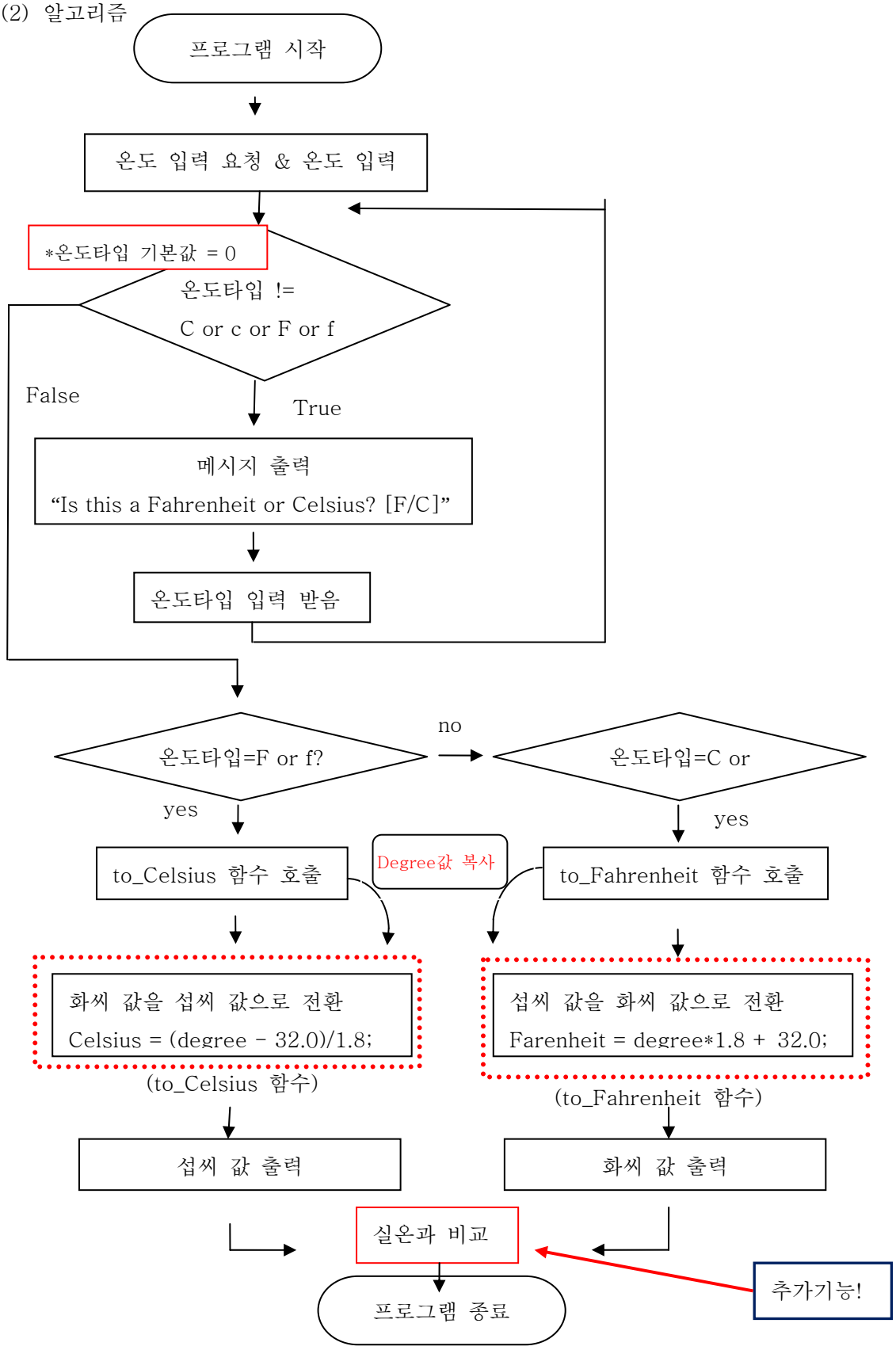
Assign #2

1. Problem #1

(1) 개요

이 프로그램은 변환할 온도값 하나를 입력받아서 이 값이 화씨일 경우 섭씨로 변환하고, 섭씨일 경우 화씨로 변환하여 출력한다.

(2) 알고리즘



(3) 전체구조

① 온도값 입력

프로그램을 실행하면 사용자에게 온도값을 입력할 것을 요청하는 메시지를 출력한다. 사용자가 온도값을 입력하면 scanf를 통해 double형 변수 degree에 저장된다.

이 이후에 Do you want really (입력값) degree?[y/n] 문구를 출력하고 사용자가 잘못 입력했을 때 다시 입력 받을 수 있도록 기능을 추가하였다.

② 온도 타입 입력

온도 값을 입력 받은 뒤에는 그 온도가 어떤 타입인지 입력할 것을 요청하는 메시지가 출력된다. 화씨일 경우 F or f, 섭씨일 경우 C or c를 입력하면 된다. 입력이 잘못 됐을 경우 error문을 출력하고 아래와 같은 while문을 사용하여 다시 입력할 수 있도록 하였다.

while(DGR_type!='F' && DGR_type!='C' && DGR_type!='c' && DGR_type!='f') -> 재입력

③ 섭씨로 변환

void to_Celsius(double degree)

if문을 사용하여 사용자가 입력한 온도 타입이 F or f일 경우 섭씨로 전환하는 위의 함수를 호출한다. 위의 함수는 온도값 만을 인자로 갖는데

Celsius = (degree - 32.0)/1.8;
위와 같은 식을 통해서 섭씨로 전환하고 그 값을 입력받은 온도와 함께 출력한다.

④ 화씨로 변환

void to_ Fahrenheit(double degree)
if문을 사용하여 사용자가 입력한 온도 타입이 C or c일 경우 화씨로 전환하는 위의 함수를 호출한다. 위의 함수역시 온도값 만을 인자로 갖는데

Farenheit = degree*1.8 + 32.0;
위와 같은 식을 통해서 화씨로 전환하고 그 값을 입력받은 온도와 함께 출력한다.

⑤ 상온과 비교 (추가기능!)

상온은 섭씨로는 15, 화씨로는 59이다. 이 값을 입력 값과 비교하여 입력된 값이 상온보다 높은지 낮은지를 출력한다.

(4) 프로그램 실행방법 및 예제

```
shell.postech.ac.kr/hiscrepl ~/Assn2/1 {153} Assn2_1.exe
Enter the degree you want to convert:
```

프로그램을 실행하면 위와 같은 문구가 뜬다.

```
Enter the degree you want to convert: 10
Do you want really 10.00 degree?[y/n]: y
Is this a Fahrenheit or Celsius? [F/C] c
        10.00 F = 50.00 C
Cooler than Normal
```

위의 그림과 같이 원하는 온도를 입력하면 제대로 입력 받았는지 여부를 묻는 문장이 뜨고 y 를 선택하면 입력 값이 화씨인지 섭씨인지를 묻는 항목이 뜬다.
위의 그림은 섭씨를 선택했을 때의 실행화면이다.

```
Enter the degree you want to convert: 70
Do you want really 70.00 degree?[y/n]: y
Is this a Fahrenheit or Celsius? [F/C] f
                70.00 C = 21.11 F
Hotter than Normal
```

이 그림은 화씨를 선택했을 때의 구문이다. 보는 바와 같이 온도가 변환되고 변환된 값이 상온(Normal)보다 높은지 낮은지 여부가 출력된다

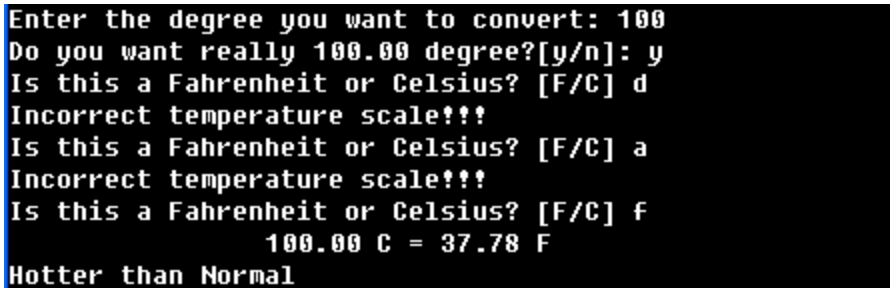
```
Enter the degree you want to convert: 20
Do you want really 20.00 degree?[y/n]: y
Is this a Fahrenheit or Celsius? [F/C] C
        20.00 F = 68.00 C
Hotter than Normal
```

```
Enter the degree you want to convert: 30
Do you want really 30.00 degree?[y/n]: y
Is this a Fahrenheit or Celsius? [F/C] f
                30.00 C = -1.11 F
Cooler than Normal
```

위의 두 그림과 같이 섭씨나 화씨를 대문자로 입력해도 된다.

```
Enter the degree you want to convert: a
You made a mistake!!
Enter the degree you want to convert: 10
Do you want really 10.00 degree?[y/n]: n
Enter the degree you want to convert: 20
Do you want really 20.00 degree?[y/n]: y
```

온도를 입력할 때 숫자가 아닌 문자를 입력하면 에러 메시지가 뜬다(추가기능!)



화씨냐 섭씨냐 선택할 경우 F, f, C, c 가 아닌 경우 에러 메시지가 뜬다

(5) 토론

① scanf 버퍼처리

scanf를 통해서 온도 타입을 입력 받을 때 온도 타입을 잘못 입력하면 다시 입력을 받아야 하는데 입력을 하지 않아도 거듭 입력되는 문제가 발생하였다. 알아본 결과 '₩n'이 버퍼에 남아 있어서 문제가 되는 것이었다.

while(trash!='₩n') trash = getchar();

그래서 위와 같은 코드를 온도 타입을 입력 받는 while문 안에 추가해서 '₩n'값을 버퍼에서 없앴다.

(6) 결론

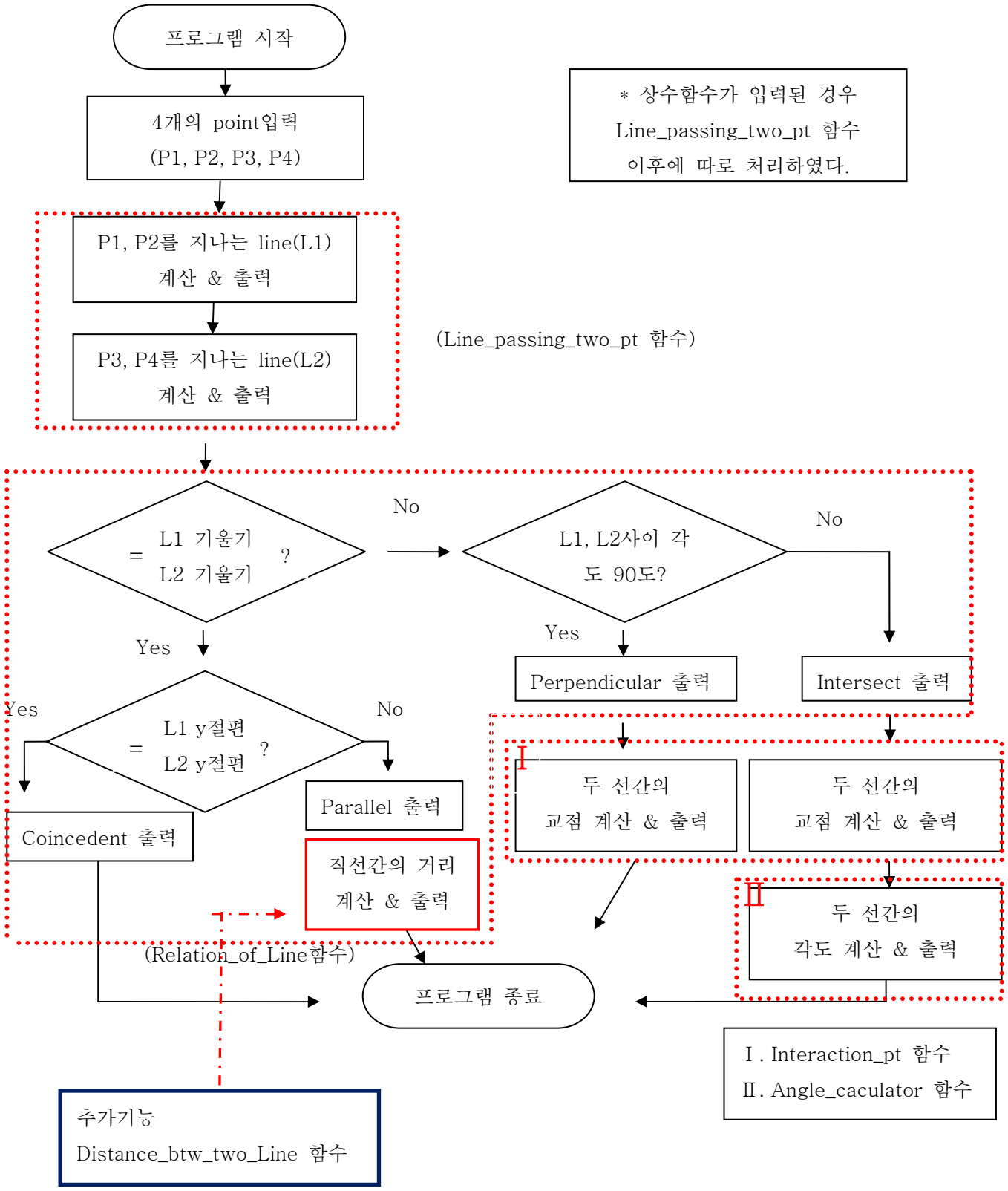
첫 번째 문제는 크게 어렵지는 않았다. 가장 어려운 부분이 scanf버퍼 처리부분이었는데 큰 어려움 없이 해결할 수 있었다.

2. Proplem #2

(1) 개요

이 프로그램은 입력 받은 point간의 관계를 구하는 프로그램이다. P1, P2, P3,P4 4개의 point를 입력 받는다. P1, P2는 L1위에 존재하며 P3,P4는 L2위에 존재한다. 이를 바탕으로 L1과 L2를 x+ ay+ b=0의 형태로 나타내고 L1과 L2의 상태(intersect, parallel, coincident, perpendicular)를 나타내고 intersect와 perpendicular일 경우 교차점의 좌표 P(x,y)를 출력하고 intersect일 경우에는 L1과 L2사이의 예각을 출력한다.

(2) 알고리즘



(3) 전체 구조

⑥ 좌표값 입력

프로그램을 실행하면 사용자에게 point1 ~ 4의 x값과 y값을 입력할 것을 요청하는 메시지를 출력한다. 사용자가 좌표값을 입력하면 scanf를 통해 int형 변수 point1의 x값 y값은 각각 p1_x, p1_y point2의 x값 y값은 각각 p2_x, p2_y 마찬가지로 point3, 4의 값들도p3_x, p3_y, p4_x, p4_y에 저장된다.

⑦ 두 점이 이루는 직선 계산

int Line_passing_two_pt t(int pt1_x, int pt1_y, int pt2_x, int pt2_y ,double *a, double *b)
가장 먼저 두 점의 y좌표가 같은지를 판단한다. 같은 경우는 상수함수인 경우기 때문에 1을 반환한다. (상수함수만 따로 처리하기 위해)
상수함수가 아닌 경우 직선의 방정식을 x+ ay+ b=0꼴로 나타내기 위해 a, b 두 값을 반환 해서 a, b를 포인터로 사용하였다. 이 값들을 구하기 위해서 아래와 같은 식을 사용하였다.

```
*a = (pt1_x - pt2_x)/(double)(pt2_y - pt1_y);
*b = (pt2_x*pt1_y - pt1_x*pt2_y)/(double)(pt2_y - pt1_y);
```

모든 좌표의 x, y 값이 모두 int형이기 때문에 a, b값을 구할 때 double형으로 casting하였다.
결과값은 “x+ ay+ b=0”형태로 출력한다.

⑧ 상수함수(y축에 수직인 직선) 처리

x값에 상관 없이 y값이 일정한 상수함수의 경우 다른 것과 동일하게 처리할 수 없어서 따로 처리하였다. 두 직선 모두 상수함수인 경우, 하나만 상수함수인 경우, 둘 다 아닌 경우로 나누어 앞의 두 경우에는 각 상황에 맞게 따로 위치관계, 교점 등을 계산하여 출력하였다.

⑨ 두 직선의 관계 조사

void Relation_of_Line(double *a_1, double *b_1, double *a_2, double *b_2)

이 기능은 위의 함수가 담당한다. Line_passing_two_pt에서 계산된 line1과 line2의 a, b값이 각각 &a_1, &b_1, &a_2, &b_2에 저장되는데 이 함수는 이 값들의 주소를 인자로 갖는다.
이 함수가 판별하는 직선의 관계는 Coincident, Parallel, Perpendicular, Intersect 네 가지다.

- Coincident: *a_1 = *a_2 & *b_1 = *b_2인 경우
- Parallel: *a_1 = *a_2 &* b_1 ≠ *b_2인 경우
- Perpendicular: *a_1 × *a_2 = -1 인 경우
- Intersect: *a_1 ≠ *a_2 & *a_1 × *a_2 ≠ -1 인 경우

이렇게 네 경우를 판단하여 결과를 출력한다.

⑩ 두 직선의 교점 조사

void Intersection_pt(double *a_1, double *b_1,double *a_2, double *b_2)

이 기능은 위의 함수가 담당한다. 두 직선의 관계가 Perpendicular이나 Intersect인 경우 Relation_of_Line함수에서 이 함수가 호출된다. 교점의 x, y 좌표는 아래의 식으로 구하고 결과값을 출력한다.

```
x = (*a_1 * *b_2 - *b_1 * *a_2)/(*a_2 - *a_1);
y = (*b_1 - *b_2)/(*a_2 - *a_1);
```

⑪ 두 직선이 이루는 각 조사

double Angle_calculator(double *a_1, double *a_2)

이 기능은 위의 함수가 담당한다. 두 직선의 관계가 Intersect인 경우 Relation_of_Line함수에서 이 함수가 호출된다. 직선이 이루는 각을 계산하기 위해서 arctangent함수(atan2)를 사용하였다. 이를 위해서 math.h파일을 include했다.

Line1의 angle = atan2(1,-*a_1)
Line2의 angle = atan2(1,-*a_2)

이 때 반환되는 값이 radian값이기 때문에 $\frac{180}{\pi}$≒57.2958을 곱하여 degree(°)로 변환하였다.

이 값이 음수면 -1을 곱하여 양수(예각)로
이 값이 90보다 크면 180에서 이 값을 빼서 예각으로 변환하여 출력하였다.

⑫ 두 직선 사이의 거리 조사 (추가기능!)

void Distance_btw_two_Line(double *a_1, double *b_1,double *a_2, double *b_2)

직선이 서로 평행할 경우 위의 함수로 두 직선간의 거리를 계산하였다. 두 직선이
L1 : x+ a_1y+ b_1=0
L2 : x+ a_2y+ b_2=0 일 때,

L1은 (0 , -b_1)점을 지난다. 이 점에서 L2까지의 거리는

$$\text{Distance} = \frac{a_2 \times -b_1 + b\_2}{\sqrt{1 + a\_1^2}}$$

위의 식으로 계산할 수 있다. 이때 제곱근은 math.h에 저장돼 있는 sqrt()함수를 사용하였다.

(4) 프로그램 실행방법 및 예제

```
shell.postech.ac.kr/hiscrepl ~/Assn2/2 {190} assn2-2.exe
Enter the value x and y of P1:
```

프로그램을 실행하면 위와 같은 문구가 뜬다.

```
Enter the value x and y of P1: 2 3
Enter the value x and y of P2: 4 7
Enter the value x and y of P3: 1 0
Enter the value x and y of P4: 3 8

L1 : x + -0.50y + -0.50 = 0
L2 : x + -0.25y + -1.00 = 0
L1 and L2 are intersect.
Intersection Point P(x,y) : 1.50, 2.00
Degree : 12.53
```

안내에 따라 알맞은 좌표 값을 입력하면 그림과 같이

두 점으로 이루어지는 두 직선의 방정식과 이들의 위치관계가 출력된다.

Intersect!

```
Enter the value x and y of P1: 0 0
Enter the value x and y of P2: 1 1
Enter the value x and y of P3: 0 1
Enter the value x and y of P4: 1 2

L1 : x + -1.00y + 0.00 = 0
L2 : x + -1.00y + 1.00 = 0
L1 and L2 are Parallel.
Distance between two Lines is 0.71
```

평행한 경우 두 직선 간의 거리가 출력된다

Parallel!

```
Enter the value x and y of P1: 1 5
Enter the value x and y of P2: 2 7
Enter the value x and y of P3: -1 1
Enter the value x and y of P4: -2 -1

L1 : x + -0.50y + 1.50 = 0
L2 : x + -0.50y + 1.50 = 0
L1 and L2 are Coincident.
```

두 직선이 일치할 경우 해당 메시지를 출력한다.

Coincident!

```
Enter the value x and y of P1: 0 0
Enter the value x and y of P2: 1 1
Enter the value x and y of P3: 0 0
Enter the value x and y of P4: 1 -1

L1 : x + -1.00y + 0.00 = 0
L2 : x + 1.00y + -0.00 = 0
L1 and L2 are perpendicular.
Intersection Point P(x,y) : 0.00, 0.00
```

두 직선이 수직인 경우 수직이라는 메시지를 출력하고 교점도 표시한다.

Perpendicular!

```
Enter the value x and y of P1: 1 1
Enter the value x and y of P2: 1 1
Enter the value x and y of P3: 1 1
Enter the value x and y of P4: 1 1

You did Wrong!! Enter the points again!!
Enter the value x and y of P1:
```

(에러처리)

(P1과 P2, P3와 P4 가 일치하면 에러 메시지 출력

```
You did Wrong!! Enter the points again!!
Enter the value x and y of P1:
```

(에러처리)

입력한 좌표가 문자일 경우도 에러 메시지가 뜬다.

(5) 토론

① 포인터의 활용

이 프로그램에선 처음으로 pointer를 사용해 보았다. 직선을 나타내기 위해선 최소한 두 개의 변수가 필요했는데 함수는 기본적으로 하나의 값만을 return할 수 있기 때문에 하나의 함수에서 두 개의 변수 값을 조작하기 위해선 pointer를 사용해야 했다.

② 'math.h' header file include

두 직선이 이루는 각을 계산하기 위해서 arctangent함수가 필요했는데 이를 위해서 math.h 헤더 파일을 include해야 했다. 유닉스에선 -lm이라는 명령어를 넣어야 제대로 컴파일이 되었다. 이 헤더파일에 대해 조사해보니 sin, cos, log, exp, 절대값 등 수학적 처리와 관련된 주요 함수가 다 있다는 데에 놀랐다. 다른 헤더파일에 대해서도 더 알아보아야겠다는 생각이 들었다.

③ 두 직선간이 수직인 겨우 처리 문제

두 직선이 수직인 경우에는 기울기의 곱이 -1이 된다. 그런데 C프로그램에서는 부동소수점 표현의 한계에 의해서 3과 $-\frac{1}{3}$을 곱해도 -1이 되지 않는다. 2진수로는 $-\frac{1}{3}$을 정확하게 표현할 수 없기 때문이다. 그래서 일단 math.h에 있는 arctangent함수인 atan2를 사용하여 두 직선이 이

루는 각을 계산하고 이를 소수점 3째 자리에서 반올림하여 정밀도를 높였다. 계산식은 아래와 같다.

Degree = (int) (degree * 100 + 0.5) / 100.0

④ 문자 입력 처리

좌표를 입력받을 때 문자를 입력했을 때 처리는 scanf의 return값을 사용하여 처리하였다. Scanf의 경우 성공적으로 입력 받은 데이터의 개수를 return하는데 한번에 입력 받는 값이 두 개이므로 scanf에서 return받은 값이 2이하일 경우 다시 입력 받도록 하였다.

(6) 결론

이번 어싸인에서는 포인터를 사용하는 데는 익숙하지 않아 조금 헤매긴 했었지만 1학년 때 전산수업을 들었었기 때문에 큰 무리 없이 과제를 완료할 수 있었다. 그래도 그 때는 별 생각 없이 사용했던 scanf함수나 header file에 대해 이것저것 알아보았더니 새로운 것을 알 수 있었다.

그리고 어싸인 공지를 제대로 읽지 않아서 곤욕을 치르었다. 처음엔 상수함수의 경우를 에러처리하려고했는데 제출하기 전에 다시 공지를 읽어보니 0x꼴로 출력하는 것이었다. 나중에 급하게 고치느라 프로그램이 지저분하게 되었다. 다음부터는 공지를 제대로 읽고 계획을 짜서 코딩해야겠다.