

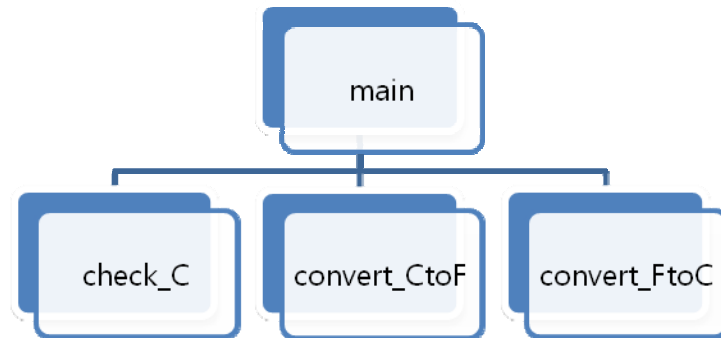
ASSIGNMENT #2

#PROBLEM 1

1. 프로그램 개요

섭씨 또는 화씨 온도를 입력 받아 섭씨는 화씨로 화씨는 섭씨로 바꾼 결과를 출력한다.

2. 프로그램의 구조



#1. 입력과 출력

입력 : 온도 - 실수형

: 입력한 온도의 단위 화씨(F) 또는 섭씨(C) - 문자

출력 : 온도 - 실수형

#2 사용할 변수, 상수 (type) identifier

(constant) CEL 1 - 섭씨를 나타낸다.

(constant) FAH 0 - 화씨를 나타낸다.

(int) C_F : 섭씨인지 화씨인지 나타내는 변수

(int) i : 종료 조건 (1이 아닌 경우 종료)

(double) temp : 입력 받은 온도

#3 main의 구조

```
main(){  
    변수 선언;  
    온도 입력  
    do{  
        조건 = check_C; /* 온도의 단위 체크 */  
        if (조건 == CEL) // (섭씨인가?)  
            covert_CtoF; // 섭씨 -> 화씨  
        else  
            convert_FtoC; //화씨 -> 섭씨  
        종료조건 입력;  
    }while (종료조건 == 1)  
}
```

#3 Function

(1) check_C

기능 : 사용자로부터 화씨인지 섭씨인지 입력 받고, 1또는 0을 리턴한다

입력 : (void), 사용자로부터 char을 입력 받는다.(C,c 또는 F,f)

출력 : (int), 화씨일 경우 FAH(0), 섭씨일 경우 CEL(1)

원하는 값을 입력 받기 위해 다음과 Loop를 구성하였다.

```
do{  
    check_c 입력 // 온도의 단위 입력  
}while( check_c != 'F' && check_c != 'f' && check_c != 'C' && check_c != 'c');
```

(2) convert_CtoF

기능 : 섭씨 온도를 화씨 온도로 바꾼다.

입력 : (double) C

출력 : (double) F

사용한 식 $F = C * 1.8 + 32.0$

(3) convert_FtoC

기능 : 화씨 온도를 섭씨 온도로 바꾼다.

입력 : (double) F

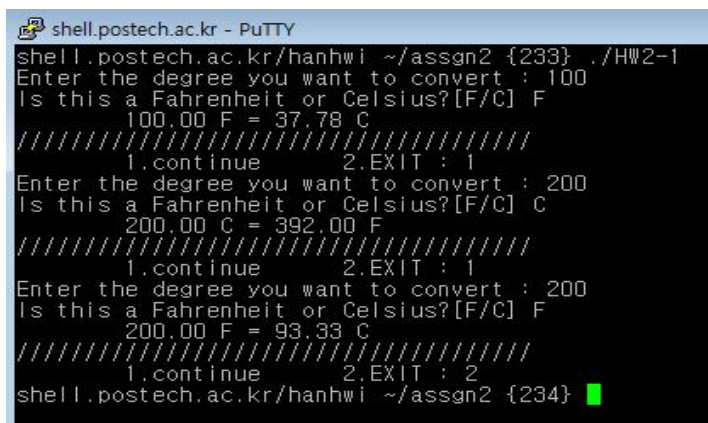
출력 : (double) C

사용한 식 $C = (F-32.0)/1.8$

3. 추가기능

입력된 온도를 변환하고 곧바로 종료하지 않고 종료 조건을 두어 사용자에게 종료를 결정하게 만들었다.

```
-----  
do{  
    scanf ("%d", &i);  
}while(i!=1); /* 1이 아닌 것을 입력받으면 종료 */  
-----
```



```
shell.postech.ac.kr - PuTTY  
shell.postech.ac.kr/hanhwi ~/assgn2 {233} ./HW2-1  
Enter the degree you want to convert : 100  
Is this a Fahrenheit or Celsius?[F/C] F  
100.00 F = 37.78 C  
////////////////////////////////////  
1.continue 2.EXIT : 1  
Enter the degree you want to convert : 200  
Is this a Fahrenheit or Celsius?[F/C] C  
200.00 C = 392.00 F  
////////////////////////////////////  
1.continue 2.EXIT : 1  
Enter the degree you want to convert : 200  
Is this a Fahrenheit or Celsius?[F/C] F  
200.00 F = 93.33 C  
////////////////////////////////////  
1.continue 2.EXIT : 2  
shell.postech.ac.kr/hanhwi ~/assgn2 {234} █
```

<변환 후, 종료 여부를 묻고 그에 따라 종료하거나 다시 온도를 입력 받는다.>

4. 사용방법

- (1)온도를 입력 받는다.
- (2)입력한 온도의 단위를 입력 받는다. 섭씨의 경우 C,c 화씨의 경우 F,f
- (3)변환 후 종료할 경우 1 입력 아닐 경우 1이 아닌 다른 정수 입력.

5. 문제해결

- (1)문제점 : 화씨 또는 섭씨를 결정할 때, 문자 대신 공백문자 또는 Enter 입력 후 문자 입력했을 때, 다음 루프는 키보드로부터 입력 받지 않고 넘어갔다.

해결 : 앞에 공백문자 또는 개행문자가 들어간 후 문자를

6. 실행 예시

```
shell.postech.ac.kr - PuTTY
shell.postech.ac.kr/hanhwi ~/assgn2 {235} ./HW2-1
Enter the degree you want to convert : 100
Is this a Fahrenheit or Celsius?[F/C] a
Incorrect temperature scale!!!
Is this a Fahrenheit or Celsius?[F/C] 1
Incorrect temperature scale!!!
Is this a Fahrenheit or Celsius?[F/C] 2
Incorrect temperature scale!!!
Is this a Fahrenheit or Celsius?[F/C] 3
Incorrect temperature scale!!!
Is this a Fahrenheit or Celsius?[F/C] 4
Incorrect temperature scale!!!
Is this a Fahrenheit or Celsius?[F/C] 6
Incorrect temperature scale!!!
Is this a Fahrenheit or Celsius?[F/C] f
100.00 F = 37.78 C
////////////////////////////////////
1.continue      2.EXIT : █
```

< 단위를 F,f,C,c를 제외하고 입력할 경우 >

```
shell.postech.ac.kr - PuTTY
shell.postech.ac.kr/hanhwi ~/assgn2 {244} ./HW2-1
Enter the degree you want to convert : 100
Is this a Fahrenheit or Celsius?[F/C] F
100.00 F = 37.78 C
////////////////////////////////////
1.continue      2.EXIT : 1
Enter the degree you want to convert : 100
Is this a Fahrenheit or Celsius?[F/C] C
100.00 C = 212.00 F
////////////////////////////////////
1.continue      2.EXIT : g
```

< 화씨에서 섭씨로, 섭씨에서 화씨로 변환 예시 >

결론

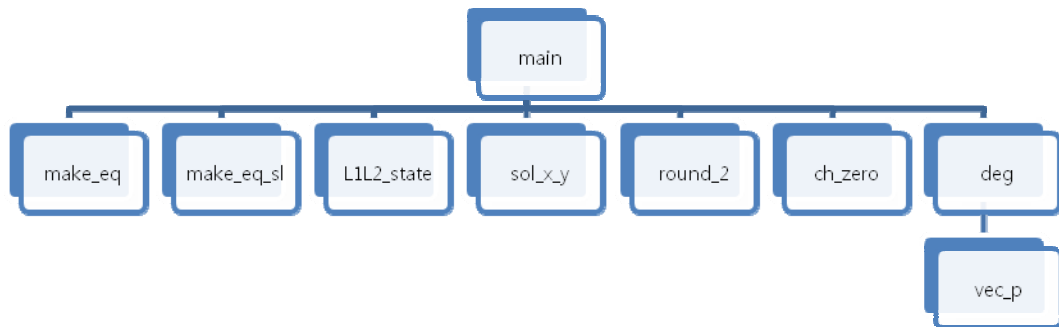
Scanf 가 char 의 자료를 받아 들일 때는 앞의 scanf 가 있다면 앞의 scanf 에서 생긴 'wn'를 읽어 버린다. "%c"를 " %c"로 공백문자를 format string 앞에 추가해 준다면 앞에서 넘어온 공백이나 'wn'을 scanf 가 읽지 않게 할 수 있다.

#PROBLEM 2

1. 프로그램 개요

두 직선의 관계를 나타내고 두 직선의 교차할 경우 교차점을 구한다.

2. 프로그램 구조



#1. 입력, 출력

입력 : 임의의 점 4개의 좌표 (x,y) – (type)float

직선의 기울기 – (type) float

출력 : 직선 1과 직선 2의 방정식

//(x+ay+b=0의 형태, x의 계수가 0일 경우 0x+ay+b = 0)

직선 1과 직선 2의 상태

//(intersect, perpendicular, coincident, parallel)

교차점 출력 (intersect,perpendicular 일 경우)

두 직선 사이의 예각 (intersect 일 경우)

#2. 사용할 변수,상수

점 : (float) x1,y1,x2,y2

기울기 : (float) slope

벡터 : (float) vx1,vy1,vx2,vy2, mag1,mag2

조건 : (char) use_slope (P/S) , (char) save (Y/N) , int (exit) (1 == 종료)

각 : (float) degree

#3. Main의 구조

while (exit)

{

/* 두 직선이 교차할 경우, 그 결과를 LINE 1의 생성에 다시 이용 할 수 있다. */

/*===== LINE 1 =====*/

if(save == N) /* 교차점을 사용하지 않는다. */

```

        점 p1 입력;
    else
        기존의 intersect point 출력;
        do{
            Slope 사용 여부 확인; // use_slope에 P또는 S 입력
            while(slope P 또는 S가 아니다.);
            if(use_slope==P) // point 사용
            do{
                점 p2 입력;
            } (점1 == 점2) // 두 점이 같으면 직선이 아니다.
        else
            기울기 입력;
/* ===== LINE 2 =====*/
        점 p3 입력;
        do{
            점 p4 입력;
        }while(p3 == p4);
/* ===== 두 직선의 식 출력 =====*/
        If (use_slope == P)
            make_eq(x1,y1,x2,y2,&a1,&b1,&c1); //LINE1
        else /* 한점과 기울기를 이용하여 식 만든다.*/
            make_eq_sl(slope,x1,y1,&a1,&b1,&c1); // LINE1

            make_eq(x3,y3,x4,y4,&a2,&b2,&c2); // LINE 2
/* ===== 두직선의 상태 =====*/
        state=L1L2_state(a1,b1,c1,a2,b2,c2);
        if (state) // intersect 또는 perpendicular 이면
        {
            sol_x_y(a1,b1,c1,a2,b2,c2);
            if (state) // intersect 이면
            {
                deg(vx1,vy1,vx2,vy2,mag1,mag2) //각 출력
            }
        }
    }
/*===== 종료 =====*/
f
If (exit != 1 && sate == intersect || state == perpendicular)
{
do{
    교차점 재사용 확인 // save 에 Y 또는 N 입력
    while( save 가 Y또는 N이 아니다);
}

```

#4 Function

1) make_eq

직선의 두 점을 입력받아서 직선의 방정식의 계수를 찾는다.

입력 : float 형 x1,y1,x2,y2 | float 형 포인터 a,b,c

출력 : 방정식을 화면에 출력 | a,b,c에 값을 돌려준다.

식) 1) X의 계수가 1인 경우

y1-y2가 0이 아니다.

방정식은 다음과 같이 된다.

$$\text{ratioinv} = (x1-x2) / (y1-y2)$$

$$X - \text{ratioinv} * Y - \text{ratioinv} * y1 - x1 = 0$$

2) X의 계수가 0인 경우

y1-y2가 0이 된다.

x 축과 평행하므로

$$0X - 1.00Y + Y1 = 0$$

2) make_eq_sl

직선의 점과 기울기를 받아서 직선의 방정식의 계수를 구한다.

입력 : float 형 x1,y1 | float 형 slope | float 형 포인터 a,b,c

출력 : 방정식을 화면에 출력 | a,b,c에 식의 계수를 돌려준다.

식) $y = \text{slope} * (x-x1) + y1$

1) slope = 0

$$0X - 1.00Y - y1 = 0$$

2) slope != 0

$$X - 1/\text{slope} * Y + 1/\text{slope} * y1 - x1$$

3) L1L2_state

계수를 이용하여 두 직선의 관계를 파악한다.

입력 : float 형 a1,b1,c1, a2,b2,c2

출력 : int 형

식) 두 직선의 기울기는 계수 a,b의 비가 같으면 같다.

$$a1:b1 = a2:b2 \Rightarrow a1b2 = b1a2 \text{ 인지 확인 필요하다.}$$

수직인 두 직선의 기울기의 곱은 -1이다.

$$b1*b2/a1*a2 = -1$$

1) slope of L1 = slope of L2 -> $a1b2 == b1a2$

A. coincident -> $c1 == c2$

B. parallel -> $c1 != c2$

2) slope of L1 != slope of L2 -> $a1b2 != b1a2$

A. perpendicular -> 두 기울기의 곱은 -1

$$\rightarrow a1a2 == -b1b2$$

B. intersect -> $a1a2 != -b1b2$

4) sol_x_y

두 직선의 계수를 받아서 교차점을 구한다.

입력 : float형 a1,b1,c1,a2,b2,c2 | float형 포인터 px,py

출력 : 교차점을 화면에 표시하고 좌표를 px, py에 전한다.

5) Deg

두 직선의 방향 벡터를 입력받아 구한다.

입력 : float형 vx1,vy1,vx2,vy2, mag1,mag2

출력 : float형 degree

식) 두 방향 벡터의 내적은 다음과 같다.

$$vx1*vx2+vy1*vy2 = |v1||v2|\cos X$$

$$\cos X = (vx1*vx2 + vy1*vy2)/(|v1||v2|) = Z$$

$$X = (\arccos)Z$$

예각을 구해야 하므로 if X > 90이 넘을 경우

degree = 180 - X로 반환한다.

6) vec_p

직선의 두 좌표를 입력받고 벡터의 x,y 성분을 구하고 크기를 리턴한다.

degree를 구하기 위해 사용된다.

입력 : float 형 x1,y1,x2,y2 | float형 포인터 vx1,vy1

출력 : float 형 mag | vx1,vy1에 값전달.

7) ch_zero

0에 가까운 수를 0으로 만든다.

입력 : float x 출력 : float x

8) round_2

소수 셋째 자리에서 반올림한다.

입력 : float x 출력 : float x

3. 추가기능

두 직선을 입력받아 직선의 관계를 구한 후, 다시 구할 지 종료할 지를 사용자에게 입력 받는다.

재 입력시 만약 교차점이 있었다면 그 교차점을 입력에 이용할 것이지를 결정할 수 있다.

재 입력시 교차점과 새로운 점 또는 기울기로 직선을 만들 수 있다.


```

shell.postech.ac.kr/hanhwi ~/assn2/d {25} ./a.out
Enter the value x and y of P1 : ex (1,2) : 1 2
Enter the value x and y of P2 : ex (1,2) : 1 3
Enter the value x and y of P3 : ex (1,2) : 44 5
Enter the value x and y of P4 : ex (1,2) : 4 4
x + 0.00 y + -1.00 = 0
x + -40.00 y + 156.00 = 0
L1 and L2 are intersect
Intersect point P(x,y)=(1.00,3.92)
Degree : 88.57
=====
If you want to quit (1)
I want to try more (other integer) : n
Do you use intersect point as new points of new Line
You use P1(1.00, 3.92)
Do you use slope of line (S) or another point (P) : f
Enter the value x and y of P2 : ex (1,2) : 1 1
Enter the value x and y of P3 : ex (1,2) : 2 2
Enter the value x and y of P4 : ex (1,2) : 3 3
x + 0.00 y + -1.00 = 0
x + -1.00 y + 0.00 = 0
L1 and L2 are intersect
Intersect point P(x,y)=(1.00,1.00)
Degree : nan
=====
If you want to quit (1)
I want to try more (other integer) : f
Do you use intersect point as new points of new Line
You use P1(1.00, 1.00)
Do you use slope of line (S) or another point (P) : c
You must enter S or P
S
Enter the slope what you want to make : 1
Enter the value x and y of P3 : ex (1,2) : 2 2
Enter the value x and y of P4 : ex (1,2) : 3 3
x + -1.00 y + 0.00 = 0
x + -1.00 y + 0.00 = 0
L1 and L2 are coincident
=====
If you want to quit (1)

```

< 교차점과 기울기, 교차점과 또 다른 점을 이용하여 L1 생성 >

4. 실행방법

- (1) 점 4개를 입력.
- (2) 두 직선의 방정식과 직선의 관계, 각, 교차점이 출력된다.
- (3) 다시 입력할 것인가 ? 정수 입력(1일 경우 종료)
- (4) 교차점을 사용 할 것인가 ? (교차점이 출력될 경우) [Y/N]
- (5) 기울기를 사용 할 것인가 ? (교차점을 사용할 경우) [S/P]

5. 문제 해결

문제점 : 직선끼리의 상태를 구하는 과정에서 float 형끼리의 비교에서는 연산에서 오차가 발생해서 ==가 성립이 안됐다.

해결 : 두 값의 차이가 0.0000001 에 들어오는 경우로 조건을 바꾸어 해결하였다.

문제점 :화면에 좌표나 식을 표현할 때 -0.00이 출력되는 경우가 있었다.

해결 : 변수를 + 0.0000001 , - 0.0000001 사이에 들어오면 0.0으로 변환하는 ch_zero 함수를 만들어서 해결.

문제점 : 교차점을 재사용할 때, 화면에 보이는 교차점의 값이 입력되지 않는다.

해결책 : 화면에서는 셋째 자리에서 반올림한 값을 출력 따라서 소수 셋째 자리에서 반올림하는 round_2함수를 만들어서 교차점을 사용할 때 round_2 함수를 이용하여 반올림하여 사용하였다.

6. 실행 예시

```

shell.postech.ac.kr/hanhwi ~/assgn2/d {212} ./a.out
Enter the value x and y of P1 : ex (1,2) : 0 1
Enter the value x and y of P2 : ex (1,2) : 1 1
Enter the value x and y of P3 : ex (1,2) : 2 2
Enter the value x and y of P4 : ex (1,2) : 2 3
  0x + -1.00 y + 1.00 = 0
  x + 0.00 y + -2.00 = 0
L1 and L2 are perpendicular
Intersect point P(x,y)=(2.00,1.00)
=====
If you want to quit (1)
  I want to try more (other integer) : 0
Do you use intersect point as new points of new Line : y
You use P1(2.00, 1.00)
Do you use slope of line (S) or another point (P) : S
Enter the slope what you want to make : 1.00
Enter the value x and y of P3 : ex (1,2) : 1 1
Enter the value x and y of P4 : ex (1,2) : 2 2
  x + -1.00 y + -1.00 = 0
  x + -1.00 y + 0.00 = 0
L1 and L2 are parellel
=====
If you want to quit (1)
  I want to try more (other integer) : 1
shell.postech.ac.kr/hanhwi ~/assgn2/d {213} █

```

```

If you want to quit (1)
  I want to try more (other integer) : 1
shell.postech.ac.kr/hanhwi ~/assgn2/d {210} ./a.out
Enter the value x and y of P1 : ex (1,2) : 1 1
Enter the value x and y of P2 : ex (1,2) : 2 2
Enter the value x and y of P3 : ex (1,2) : 0 0
Enter the value x and y of P4 : ex (1,2) : 0 1
  x + -1.00 y + 0.00 = 0
  x + 0.00 y + 0.00 = 0
L1 and L2 are intersect
Intersect point P(x,y)=(0.00,0.00)
Degree : 45.00
=====
If you want to quit (1)
  I want to try more (other integer) : 2
Do you use intersect point as new points of new Line : n
Enter the value x and y of P1 : ex (1,2) : 1 1
Enter the value x and y of P2 : ex (1,2) : 2 2
Enter the value x and y of P3 : ex (1,2) : 0 0
Enter the value x and y of P4 : ex (1,2) : 0 1
  x + -1.00 y + 0.00 = 0
  x + 0.00 y + 0.00 = 0
L1 and L2 are intersect
Intersect point P(x,y)=(0.00,0.00)
Degree : 45.00
=====
If you want to quit (1)
  I want to try more (other integer) : 1
shell.postech.ac.kr/hanhwi ~/assgn2/d {211} █

```

7. 결론

float 형의 경우에는 두 값이 같음을 비교 할 때는 ==을 사용할 경우에는 그 값이 연산을 거치지 않은 값이어야 한다. 그렇지 않을 경우에는 약간의 오차로 다르게 나올 수 있다. 부등호를 이용하여 같은지 비교하여야 한다.