

프로그래밍 보고서

10202 강상원

1. 목적 : 지리 좌표계의 두 좌표(경도 값, 위도 값)를 입력 받아, 두 위치 사이의 실제 거리를 구한다.
2. 사용언어 : C 언어
3. 과정 :
 - a. 두 좌표를 입력 받는다.
 - b. 각각의 소수점 좌표를, 도-분-초 좌표로 변환한다.
 - c. 계산식을 이용하여 두 좌표 사이의 거리를 구한다.
4. 좌표사이의 거리 구하는 식 :

지리좌표간 두점(A,B)의 거리

$X = A\text{경도} - B\text{경도}$, $Y = A\text{위도} - B\text{위도}$, $R(\text{지구의 반지름}) = 6,378.135\text{km}$

$$C = \cos(\text{도}) \times \left(\frac{2\pi R}{360} \right), \cos(\text{도}) = \frac{A\text{위도}^\circ + B\text{위도}^\circ}{2}$$

$$D = \frac{2\pi R}{360}$$

$$\text{두점(A,B)의 거리} = \sqrt{(X)^2 + (Y)^2}$$

$$= \sqrt{\left(\left((\text{도} \times C) + \left(\text{분} \times \left(\frac{C}{60} \right) \right) + \left(\text{초} \times \left(\frac{\left(\frac{C}{60} \right)}{60} \right) \right) \right) \right)^2 + \left(\left((\text{도} \times D) + \left(\text{분} \times \left(\frac{D}{60} \right) \right) + \left(\text{초} \times \left(\frac{\left(\frac{D}{60} \right)}{60} \right) \right) \right) \right)^2}$$

(참고: https://ko.wikipedia.org/wiki/지리좌표_거리#cite_note-2)

5. 소스코드 (C 언어)

```

1  #include <stdio.h>
2  #include <math.h>
3  int main(){
4      int sc_d0, sc_p0, sc_d00, sc_p00;
5      printf("Enter first location (Format : ddd.mmmmm/dd.mmmmm) : ");    //첫번째 위치값 입력
6      scanf("%d.%d/%d.%d", &sc_d0, &sc_p0, &sc_d00, &sc_p00);
7      int se_d0, se_p0, se_d00, se_p00;
8      printf("Enter second location (Format : ddd.mmmmm/dd.mmmmm) : ");    //두번째 위치값 입력
9      scanf("%d.%d/%d.%d", &se_d0, &se_p0, &se_d00, &se_p00);
10
11     // 입력 받은 두 소숫점 좌표를 도-분-초 좌표로 변형
12     /*
13     도 : 소숫점 좌표값의 정수값
14     분 : 소숫점 좌표값의 소숫점 이하값 * 60을 했을 때 결과값의 소숫점 앞 정수
15     초 : 분을 계산한 결과값의 소숫점 이하값 * 60을 했을 때의 앞의 4자리값.*/
16     float sc_m0, sc_l0, sc_s0, sc_l00, sc_m00, sc_s00;
17     sc_l0 = (float)sc_p0 * 1 / 1000000 * 60;
18     sc_m0 = (int)((float)sc_p0 * 1 / 1000000 * 60);
19     sc_s0 = (((float)sc_l0 - (float)sc_m0) * 60);
20     sc_l00 = (float)sc_p00 * 1 / 1000000 * 60;
21     sc_m00 = (int)((float)sc_p00 * 1 / 1000000 * 60);
22     sc_s00 = (((float)sc_l00 - (float)sc_m00) * 60);
23     float se_m0, se_l0, se_s0, se_l00, se_m00, se_s00;
24     se_l0 = (float)se_p0 * 1 / 1000000 * 60;
25     se_m0 = (int)((float)se_p0 * 1 / 1000000 * 60);
26     se_s0 = (((float)se_l0 - (float)se_m0) * 60);
27     se_l00 = (float)se_p00 * 1 / 1000000 * 60;
28     se_m00 = (int)((float)se_p00 * 1 / 1000000 * 60);
29     se_s00 = (((float)se_l00 - (float)se_m00) * 60);
30     // 두 위치 사이의 거리를 구하는 식
31     float ca_d0, ca_m0, ca_s0, ca_d00, ca_m00, ca_s00;
32     float r = 6378.135, d = 111.31945588668853276112184419674;    // r:지구의 반지름, d:2πr/360
33     double ck = cos((sc_d00 - se_d00) / 2) * d;
34     ca_d0 = sc_d0 - se_d0;
35     ca_m0 = sc_m0 - se_m0;
36     ca_s0 = sc_s0 - se_s0;
37     ca_d00 = sc_d00 - se_d00;
38     ca_m00 = sc_m00 - se_m00;
39     ca_s00 = sc_s00 - se_s00;
40     float calc_1 = ca_d0 * ck;
41     float calc_2 = ca_m0 * (ck / 60);
42     float calc_3 = ca_s0 * (ck / 3600);
43     float cal_1 = pow((calc_1 + calc_2 + calc_3), 2);
44     float calc_4 = ca_d00 * d;
45     float calc_5 = ca_m00 * (d / 60);
46     float calc_6 = ca_s00 * (d / 3600);
47     float cal_2 = pow((calc_4 + calc_5 + calc_6), 2);
48
49     printf("Distance of two points : %.4fkm (%.4fm)", (cal_1+cal_2)*10, (cal_1+cal_2)*10000);
50     return 0;
51 }

```

6. 보완할 점

- 함수를 사용하지 않았고, 변수명이 서로 비슷하여, 소스코드의 가독성이 떨어진다.

7. 느낀점 :

지리좌표 위치 사이의 거리를 구하는 공식을 직접 구해서 하는게 아닌, 이미 있는 공식을 사용하여 프로그래밍을 하였다는 점이 아쉬웠지만, 모르는 부분을 구글링을 통해 새로 알게 되어가면서 소스코드를 제작하는 과정이 재미있었고, C 언어 뿐만 아니라, Javascript 나, python 과 같은 타언어를 이용해서 똑 같은 소스코드를 만들어 보고 싶다는 생각이 들었다.