**★★★★☆**

**題組：Contest Volumes**

**題號：11817: Tunnelling the Earth**

**解題者：林琦**

**解題日期：2025年5月1日**

**題意：**

從地球上一個地點到另一地點，走地球表面（大圓路徑）與穿越地球內部（直線）之間的距離差，並輸出這個差值（四捨五入後的整數）。輸入是兩點的經緯度，正值表示北緯和東經，負值則表示南緯和西經。地球半徑固定為 6371009 公尺。

**題意範例：**

1

43.466667 -80.516667 30.058056 31.228889

🡺 802333

**解法：**

先把輸入的經緯度轉成弧度，接著用球面餘弦公式計算地表的大圓距離，再把經緯度轉成三維座標算出穿越地球的直線距離，最後兩者相減並四捨五入輸出整數結果。

**解法範例：**

　$d\_{1}=R×acos(\sin(θ\_{1})×\sin(θ\_{2})+\cos(θ\_{1}×)\cos(θ\_{2}×\cos(∆λ)))$

　$d\_{2}=\sqrt{\left(x\_{1}-x\_{2}\right)^{2}+\left(y\_{1}-y\_{2}\right)^{2}+\left(z\_{1}-z\_{2}\right)^{2}}$

**討論：**

 (1) 地球是完美球體的假設：實際上地球是略為扁平的橢球體，這樣的形狀會造成距離計算上的誤差，尤其在極地與赤道地區，誤差會更為明顯。

**程式：**

#include <iostream>

#include <cmath>

#define M\_PI 3.14159265358979323846

using namespace std;

double toRadians(double degree) {

    return degree \* M\_PI / 180.0;

}

int main(){

    int T;

    cin >> T;

    for (int i = 0; i < T; i++) {

        double lat1, lon1, lat2, lon2, new\_lat1, new\_lon1, new\_lat2, new\_lon2, d1, d2;

        // 地球半徑

        double R = 6371009;

        cin >> lat1 >> lon1 >> lat2 >> lon2;

        new\_lat1 = toRadians(lat1);

        new\_lon1 = toRadians(lon1);

        new\_lat2 = toRadians(lat2);

        new\_lon2 = toRadians(lon2);

        double sin\_new\_lat1 = sin(new\_lat1);

        double sin\_new\_lat2 = sin(new\_lat2);

        double cos\_new\_lat1 = cos(new\_lat1);

        double cos\_new\_lat2 = cos(new\_lat2);

        // 計算 Δλ

        double delta\_lambda = new\_lon2 - new\_lon1;

        // 取 cos(Δλ)

        double cos\_delta\_lambda = cos(delta\_lambda);

        double angle = sin\_new\_lat1 \* sin\_new\_lat2 + cos\_new\_lat1 \* cos\_new\_lat2 \* cos\_delta\_lambda;

        d1 = R \* acos(angle);  // 正確的大圓距離

        // 起點轉三維

        double x1 = R \* cos(new\_lat1) \* cos(new\_lon1);

        double y1 = R \* cos(new\_lat1) \* sin(new\_lon1);

        double z1 = R \* sin(new\_lat1);

        // 終點轉三維

        double x2 = R \* cos(new\_lat2) \* cos(new\_lon2);

        double y2 = R \* cos(new\_lat2) \* sin(new\_lon2);

        double z2 = R \* sin(new\_lat2);

        // 計算穿越地球的直線距離

        d2 = sqrt((x1 - x2)\*(x1 - x2) + (y1 - y2)\*(y1 - y2) + (z1 - z2)\*(z1 - z2));

        int diff = round(d1 - d2);

        cout << diff << endl;

    }

}