```c
/*
    HW5
    Hamza AlHasan-1181636
    Oct 17 2020
*/
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

#define k 470 // N/m
#define L0 0.1 // m
#define g 9.8 // m/s^2
#define eps 1e-16 // epsilon -the error-
#define N_of_it 100000 // number of iterations in the method

//The function that we want to find its zero
double f(double theta, double m){//theta in radians
    return (double)( 2*k*L0*(tan(theta)-sin(theta)) - m*g );
}

//Using secant method to find the zero of f(x) with error = eps
double secant_method(double x0, double x1, double m){// return the zero of the function
    int i=0;
    double x, zero;
    while(1==1)//fabs(f(x,m))>=eps && i<N_of_it
    {
        x=x1-(f(x1,m)*(x1-x0))/(f(x1,m)-f(x0,m));
        x0=x1;
        x1=x;
        //printf("iteration number: %d \tf(%.15lf) = %.15lf\n",i,x,f(x,m));
        i++;
        if(i>=N_of_it || fabs(f(x,m))<eps){
            zero=x;
            //printf("\n The zero of the function = %.15lf\n",x);
            break;
        }
    }
    return zero;
}

double bisection_method(double x0, double x1, double m)
{
    if (f(x0,m)*f(x1,m)>=0)
    {
        printf("You have to choose an initial interval such that f(x0)f(x1)<0\n");
        return 0;
    }
    int i=0;
    double x, zero;
    while (1==1)
    {
        // Find middle point
        x = (x0+x1)/2;
        // Decide the side to repeat the steps
        if(f(x,m)*f(x0,m) < 0)
            x1 = x;
        else
            x0 = x;
        i++;
        //printf("iteration number: %d \tf(%.15lf) = %.15lf\n",i,x,f(x));

        // Check if middle point is root
        if(fabs(f(x,m))<=eps || i>=N_of_it){
            zero=x;
            break;
        }
    }
    return zero;
}

int main()
{
    // Initial values
    double x0=0.1, x1=1.21, zero;
    FILE *outputR = fopen("thetaR_VS_mass.txt", "w");//thetaR: angles in radians
    FILE *outputD = fopen("thetaD_VS_mass.txt", "w");//thetaD: angles in degrees

    // 0 for secant method - 1 for bisection method
    int check=1;

    for(double m=0.01; m<=16; m+=0.01){
        if(check==0) zero=secant_method(x0,x1,m);
        else if(check==1) zero=bisection_method(x0,x1,m);
        fprintf(outputR,"%f\t%f\n",zero,m);
        fprintf(outputD,"%f\t%f\n",zero*180/M_PI,m); //M_PI is pi=3.14***
    }
    return 0;
}
```