```c
/* Finding where n will be constant using double precision
 *
 * Author: Dustin
 * ME2045
 * September 21
 *
 * Use the following to compile
 * gcc divergentDouble.c -std=c99 -O3 -lm -o divergentDouble.exe
 *
 */
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>
#include <math.h>
#include "timer.h"

int main(){
        // Timer Variables
                double start, finish, flop, elapsedTime;
        // Double Precision
        double i,k,q,increment;
        q = 0.0;
        k = 1.0;
        i = 1.0;

        GET_TIME(start); //start the timer

        // while next value subtracted by previous value doesnt equal to zero, continue
        while (k-q != 0.0){
                q = k;
                increment = 1.0/i;
                k = k + increment;

                // Output every 50,000 steps
                if(remainder(i,50000) == 0){
                        printf("i = %lf, k = %0.55lf\n",i,k);
                 }

                // When divergent series stops increasing, this is the final iteration
                if (k-q == 0.0){
                        printf("At n = %lf, the seires stops increasing at k = %0.55lf\n",i-1,q);
                return 0;
                }
                i++;
        }
        GET_TIME(finish);
        elapsedTime = finish - start;
        printf("elapsed wall time = %.6lf seconds\n",elapsedTime);
        return EXIT_SUCCESS;
}
```