Aditya Agrawal
aagraw14@ucsc.edu

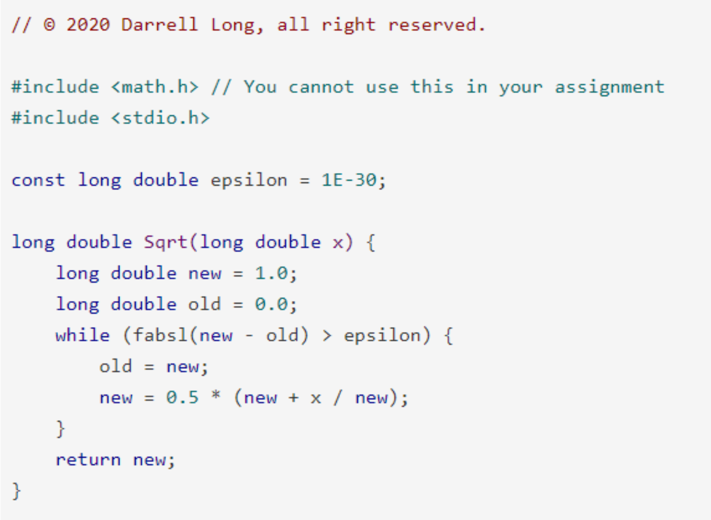
# CSE13s Spring 2021
# Assignment 2: A Small Numerical Library
# Design Document

This program will implement the mathematical functions arcsin, arcos, arctan, and log without using the provided math.h library. Only the basic addition, subtraction, multiplication, and division operations will be used as well as helper functions implemented by me for square root, absolute value, and exp(). The values generated by my program will then be compared to the results obtained from the same functions in the math.h library.

Helper Methods:

```
double Abs(double x){
    if x is above 0
        return x
    else return -x
}
```

Sqrt() based on code from Prof. Long posted to Piazza:



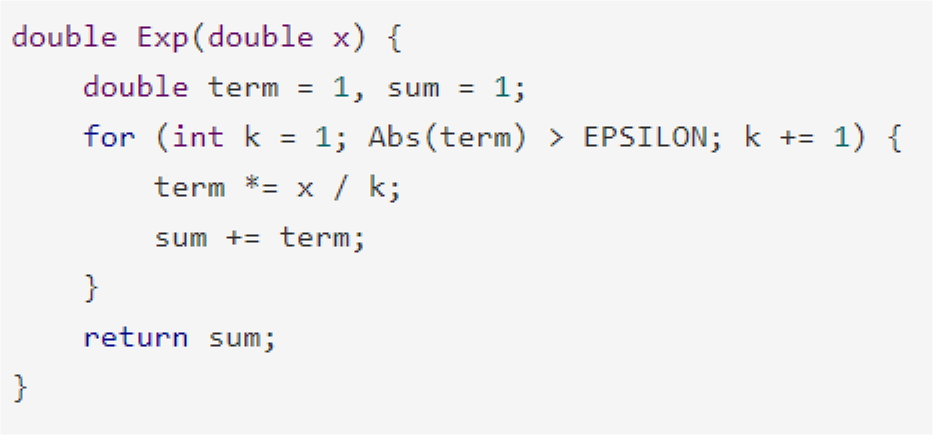
```c
// © 2020 Darrell Long, all right reserved.

#include <math.h> // You cannot use this in your assignment
#include <stdio.h>

const long double epsilon = 1E-30;

long double Sqrt(long double x) {
    long double new = 1.0;
    long double old = 0.0;
    while (fabsl(new - old) > epsilon) {
        old = new;
        new = 0.5 * (new + x / new);
    }
    return new;
}
```

Exp() based on code posted on Piazza:



```
double Exp(double x) {
    double term = 1, sum = 1;
    for (int k = 1; Abs(term) > EPSILON; k += 1) {
        term *= x / k;
        sum += term;
    }
    return sum;
}
```

arcsin:
Using Newton's method of

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

arcsin(x) = y
sin(y) = x
will solve for y by setting equal to 0 and using newton's method
sin(y) - x = 0
derivative (sinx)' = cosx

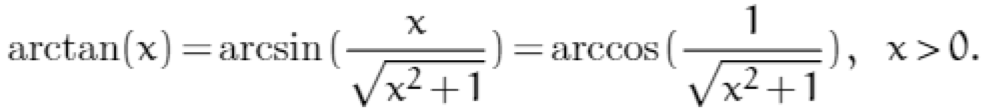
```
double arcsin(double x) {
        double old = -5
        double new = x
        while( Abs(old - new) > epsilon ){
                old = new
                new = old - ( (sin(old) - x) / (cos(old)) )
        }
        return new
}
```

arccos:

$$\arccos(x) = \frac{\pi}{2} - \arcsin(x)$$

```
double arccos(double x) {
        double output = pi/2
        output -= arcsin(x)
        return output
}
```

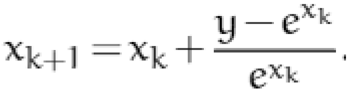
arctan:



$$\arctan(x) = \arcsin\left(\frac{x}{\sqrt{x^2+1}}\right) = \arccos\left(\frac{1}{\sqrt{x^2+1}}\right), \quad x > 0.$$

```
double arctan(double x){
        double numerator = x
        double denominator = Sqrt(x * x + 1)
        return arcsin(numerator/denominator)
}
```

log(essentially ln):



$$x_{k+1} = x_k + \frac{y - e^{x_k}}{e^{x_k}}.$$

```
double log( double x ) {
        double old = 0
        double new = 1
        while( Abs( new - old) > epsilon ) {
                old = new
                new = old + ( (x - Exp(old)) / Exp(old) )
        }
        return new
}
```

All pictures and screenshots taken from piazza or assignment doc provided by Prof. Long.

Afterthoughts/corrections: The only problem I ran into was with the arcSin function, in the -0.9 to -0.5 range of x there was a major difference between my function and the library function. I changed the initial guess to the value of x that was passed into the function which seemed to resolve the problem which was most likely due to the initial guess being too far away from the solution. I also changed the value of the variable new to be -5 so that in the case that the user passes in 0 the loop will still execute at least once.