

## The Pythagorean Theorem

### Background

Pythagoras of Samos was an Ionian Greek mathematician from the 6<sup>th</sup> century BC who is best known for the Pythagorean Theorem, a mathematical theory that says that if a right triangle has sides with length  $a$  and  $b$ , then the length of the hypotenuse,  $c$ , satisfies the formula:

$$a^2 + b^2 = c^2$$

### Project Description

For project 1, please write a C program called `pythag.c` that takes two command line arguments, which represent the length of two sides of a right triangle –  $a$  and  $b$  respectively. These arguments will be character strings, so you should use the “`atof`” function in `stdlib.h` to convert the character strings into double precision floating-point numbers. If your program is called with less than, or more than, two command line arguments, it should print an error message that specifies what arguments you expect, and return a return code that indicates that the program failed.

Your program should then calculate the length of the hypotenuse,  $c$ . You may use C library functions EXCEPT those in the `math.h` library. (This includes built-in functions like “`sqrt`”, “`pow`”, or logarithm and exponential functions.) Hint: Look at the web, such as [Wikipedia Square Root Computation](#) to get some ideas on how to compute the value for  $c$  without `math.h` library functions. Do not copy C code from the web! Write your own code from the ideas on the web. Your calculation should compute the value of  $c$  accurately to at least 6 decimal places. In other words, your answer should be the correct value for  $c$  +/- 0.000001.

Finally, your program should print out the values of  $a$ ,  $b$ , and  $c$ , using the `printf` statement:

```
printf("a=%f, b=%f, c=%f\n",a,b,c);
```

and then return a return code that indicates that the program worked.

### Project Submission

Upload your `pythag.c` file in the Project 1 submission area on MyCourses in the Content area under “Project Submissions”.

### Project Grading

After the due date, your submission will be graded as follows. The project is worth 100 points. Your code will be compared to all other student's code using an automated code plagiarism checker that can detect copied code even if you try to "fix" it after copying. Then, your code will be compiled on a Linux Lab machine. If there are compiler errors, the professor will attempt to fix your code. If your code can be fixed, the rest of the grading will be performed on that fixed code. Once compiled, your code will be tested with various simple values for  $a$  and  $b$ , and the results will be compared to correct results. Points will be deducted for the following reasons:

Problem	Deduction
Compiler error that can be fixed	-15 points per error
Unfixable compiler error, or >4 errors	-60 points
Compiler warning messages	-5 points per type of warning message
Illegible or Poorly formatted code	Up to -10 points
Incorrect results	-10 points per invocation (up to -40 points)
Late submission	-10 points per every 24 hours late
Code matches another student's code	-100 points

There are also some "advanced" test cases – values for  $a$  and  $b$  - which may be quite difficult for beginning students. Your code will be run on these advanced tests cases, and if your code produces the correct results for these test cases, you will receive up to 10 extra credit points per invocation. Extra credit points will be used to offset deductions for this project, but even with extra credit points, no grade will exceed 100.