CIS 330: Project #1C
Assigned: April 9th, 2018
Due April 16th, 2018
(which means submitted by 6am on April 17th, 2018)
Worth 2% of your grade

Please read this entire prompt. It is your responsibility to understand the directions it contains.

Assignment: Download the file "Proj1C.tar". This file contains a C-based project. You will build a Makefile for the project, and also extend the project.

What's in the tar file?
When you untar the file ("tarball"), you will see:
- math330.h: the header file for the "math330 library"
- .c files in the trig and exp directories: the source files for the "math330 library"
- cli.c: a program that uses the "math330 library"

What is your assignment?

(1) Build a Makefile for math330
(2) Extend the math330 library

Details for both below.

IMPORTANT: this project will be graded on ix-dev.cs.uoregon.edu. If your project does not work there, you will likely be docked significantly.

== Build a Makefile for math330 ==

Your Makefile should:

(1) create an include directory
(2) copy the math330.h header file into the include directory
(3) create a lib directory
(4) compile the .c files in trig and exp as object files (.o)
(5) make a static library using the trig and exp object files
(6) install the library to the lib directory
(7) create a bin directory. "bin" is short for binary and it is where binary files are stored for later execution. In this context "binary" file means a program that is created by a compiler. ("Binary" often means an executable file that is not a script.)
(8) compile the "cli" program against the include and library directories. The cli program should be compiled so that it is created in the bin directory. Use gcc to compile the program.

Some extra guidance:

(1) I want you to have a Makefile rule to make the program bin/cli

(2) I want you to have a Makefile rule to build the library. Call it libmath330.a
(3) I want you to have Makefile rules for individual source code.
(4) Never use include "file.h". Always use <file.h>

When your Makefile does all of these things, then you have completed the first step.

NOTE: "mkdir include" fails if you call it twice in a row, since the 2nd invocation will complain that "include" already exists. But "mkdir –p" will not fail on the 2nd invocation. Do "man mkdir" to learn more. (And this tip is helpful if you are developing your Makefile and you start running it more than one time.)

== Extend the math330 library ==

You should:

(1) add 3 new functions: arccos, arcsin, and arctan (each in their own .c source file)
(2) Extend the "cli" program to support these functions
(3) Extend your Makefile to support the new functions

What to turn in:

When you are done, create a new tarball:

```
% ls   # demonstrate that the current working directory contains Proj1C
Proj1C
% tar cvf Proj1C.tar Proj1C  # command for tarring up Proj1C.tar
```

Then submit Proj1C.tar

Before you submit, you test your code on ix-dev.

```
% scp Proj1C.tar username@ix-dev.cs.uoregon.edu
% ssh –l username ix-dev.cs.uoregon.edu
% tar xvf Proj1C.tar   # shell is now on ix
% cd Proj1C
% make
% ./bin/cli cos 60
```

Brent will go over scp and ssh in lab on Friday.

The project will be graded on ix-dev.

Note 1: students from previous terms don't like this project. It involves a lot of googling since lecture hasn't perfectly spelled out how to do Makefiles. Also, if you get a link error for an unresolved symbol for sin or cos, then I recommend you do a Google search.

Note 2: I penalize heavily for non-working code. I strongly prefer working code late over non-working code on time. Translation: expect <50% credit for non-working code.