Binary Files

On the course homepage, you'll find a source file, readAndPrint.c, and a binary input file, doubles.bin. The input file isn't a text file, so you shouldn't try to view it in your web browser (it'll look like garbage) or download a copy via cut-and-paste (you'll probably get garbage). It will probably be safer to just download these files with the following curl commands:

curl -O https://www.csc2.ncsu.edu/courses/csc230/exercise/exercise18/readAndPrint.c
curl -O https://www.csc2.ncsu.edu/courses/csc230/exercise/exercise18/doubles.bin
curl -O https://www.csc2.ncsu.edu/courses/csc230/exercise/exercise18/expected.txt

The input file is a sequence of doubles, written out in binary, 8 bytes each.

This source file is almost empty (except for comments). You will add code to open the binary input file (in binary mode) and open a text output file, output.txt. Your program will always open a file named "doubles.bin" as the input file and write to the file named "output.txt" as the output file. When we test it, we'll replace the "doubles.bin" file with a few different files, to try it out on different inputs.

You will read all the doubles from doubles.bin and store them in an array. Then, you will iterate over the array, printing the cosine of each double to your output file (each value on a line by itself, with 4 fractional digits of precision).

How can you tell when you've reached the end of the input file? You'll be reading each double as 8 bytes, read from the file into the memory location used to to store a double, so fread() would be a good tool for this job. Have a look at what the return value of fread() tells you. You should be able to figure out how to use this to tell when you've reached the end-of-file.

Finally, close both of your files and you're done. If your program is working, you should be able to run it like this:

```
$ ./readAndPrint
```

It will read input from doubles.bin and create a new output file named output.txt. If you look in this output file, you should see:

```
$ cat output.txt
1.0000
0.2527
0.2935
-0.1256
0.5772
-0.5801
0.8914
0.8750
0.8729
-0.9970
-0.7691
0.4634
-0.4401
0.9408
0.8605
0.5519
0.9097
0.9970
-0.7696
0.7848
```

When your program is working, submit your readAndPrint.c source file under the exercise_18 assignment in Moodle.