Assignment 2

Submit Assignment

Due Jun 29 by 11:59pm **Points** 100
Submitting a text entry box or a file upload **Available** until Jul 14 at 11:59pm

For this assignment you will extend **avg** (from assignment #1) to average numbers from multiple files.

You will develop a program **avg_many** that is given a list of files on the command line. You will need to average up all of the floating point numbers in all of the files to produce a single average.

Assume that you have text files with numbers separated by newlines. You want to read the numbers in from all the files. Since you want to be flexible on the type and size of numbers your program can handle, you should use double floating point.

If a file appears multiple times on the command line, those numbers will be processed multiple times.

You should use **fork()** to start a process for each of the files so that we can compute sums and counts in parallel. To communicate between processes you will use **pipe()**. You need to wait for all the processes to finish using **wait()**.

You must be able to handle files with any number of numbers (including no numbers). You can assume that the files are well formed: they will contain only valid numbers separated by newlines.

The **code**/ipc directory contains examples of using fork, pipe and wait, which you can consult for help.

For example, **numbers.txt** contains a list of numbers that average to 2.1. running

```
./avg_many numbers.txt
```

will output

```
2.1000
```

```
./avg_many numbers.txt numbers.txt
```

will also output

```
2.1000
```

```
./avg_many numbers.txt morenumbers.txt
```

will output (assuming morenumbers.txt just has the number 1.0)

```
1.9166
```

(trailing zeros are ok.)

Grading

overall requirements of assignment are satisfied	20%
compiles cleanly	5%
output correctly for given test case	10%
outputs correctly for tester's test cases	10%
is fork() used correctly	10%
is wait*() used correctly	10%
is pipe() used correctly	10%
program flow is understandable	5%
code is commented and indented (use of white space)	10%
number of numbers that can be processed is not limited	5%
submission of **avg_many.c** inside a zip file called **proj2.zip**	5%

Submission

Upload a zip file called proj2.zip that contains a single file: avg_many.c