Name: _____

Problem	Points	Score
1	40	
2	40	
3	20	
Total	100	

Notes:

(1) The first step in this exam is to create a workspace in the following directory:

/data/courses/ece_1111/current/resources/data

Your directory should be your last name all lowercase, followed by an underscore, following by your first name (e.g, "picone_joseph"). Set the permissions using "chmod u+rwx,g-rwx,o-rwx <lastname>" so only you have read and write permission to this directory. Create three subdirectories within this directory: p01, p02 and p03. You will use these for problems 1, 2 and 3 respectively. Put ALL your code in these directories. Do not touch your files after the exam is over.

Failure to follow these instructions will result in a grade of 0. This preamble is part of demonstrating you have basic Linux literacy.

(2) Your code must be nicely formatted and well commented, or I will deduct at least 10 points per problem.

(3) For this exam you are allowed to open a terminal window on your computer, you are allowed to web surf with Google, but you cannot use online chat or other interactive services.

Problem No. 1: The directory tree:

/data/courses/ece_1111/current/resources/data

contains over 1000 files. One of these files contains the words "ece_1111 is a fun class" spelled using mixed case (e.g., EcE_1111 or fUN or CLaSS). Find the full pathname of the file (e.g. /data/courses/..../<filename>.txt) containing these words. Find the line number in the file in which this occurs. You must do this by running the command from this directory:

/data/courses/ece_1111/current/exams/exam_01/lastname_firstname/p01

Show the results of you command in a file named p01.txt. Put your Linux command in a file named p01.sh. I should be able to type "p01.sh" from the command line and see your result.

Your script, p01.sh, must take a directory path as an argument. The following two commands should produce your answer:

```
sd /data/courses/ece_1111/current/exams/exam_01/lastname_firstname/p01
p01.sh /data/courses/ece_1111/current/resources/data > p01.txt
```

I should be able to run your command from any directory.

Note also that you cannot generate any intermediate files. Use a single command line, though you can use multiple commands on that line.

Problem 2: Random number generation is one of the more interesting and difficult things to do on a computer. We will write a simple piece of code that uses roundoff errors to generate random numbers.

Declare a float variable called val, and set it to the math constant M_PI. Write a loop to add val to another float, sum, 10,000 times. Be sure to initialize sum to zero. Then write another loop to subtract val 10,000 times from sum. Compute the error.

Extract the four least significant bits from sum and print them to stdout using this format:

b0 = 1 -or- 0 (depending on its value)
b1 = 1 -or- 0 (depending on its value)
b2 = 1 -or- 0 (depending on its value)
b3 = 1 -or- 0 (depending on its value)

Problem No. 3: We learned that binary coded decimals are used to represent numbers as integers. We also discussed the relationship between ASCII characters and decimal values. Write a C program that adds the decimal values of ASCII characters using command line arguments. Your program should accept the characters from the command line:

my_prog.exe a b c d

and produce the following output:

the sum of the command line values is 394

For example, the characters abcd sum to 97+98+99+100 = 394. Your output should be an integer value – no decimal point. Your program should work for any values entered on the command line – don't hardcode the example above or limit it to 4 numbers. It should work for any number of arguments.