

Name: \_\_\_\_\_

Problem	Points	Score
1	50	
2	50	
Total	100	

Notes:

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

`/data/courses/ece_1111/current/exams/ex_01`

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+rw,g-rwx,o-rwx <lastname>`" so only you have read and write permission to this directory. Create subdirectories within this directory: p01, p02, ... You will use these for problems 1 and 2, ... respectively. Put ALL your code in these directories. Do not touch your files after the exam is over.

- (2) Your code must be nicely formatted and well commented, use Makefiles, etc. as we have done all semester. Failure to do these things will significantly lower your grade.
- (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.

**(50 pts) Problem No. 1:**

Write a shell script that processes all the files in this directory:

```
/data/courses/ece_1111/current/exams/ex_01/picone_joseph/p01
```

whose names end in “.txt” and adds up the decimal numbers contained in those files. When I run your script, it should accept a pathname as an argument, print the name of each file and the value contained in the file, and then print the total sum:

```
ece-000_[1]: p01.sh /data/courses/ece_1111/current/exams/ex_01/picone_joseph/p01
1: /data/courses/ece_1111/current/exams/ex_01/picone_joseph/p01/f004.txt, value = -2.0000
2: /data/courses/ece_1111/current/exams/ex_01/picone_joseph/p01/01/f001.txt, value = 1.0000
3: /data/courses/ece_1111/current/exams/ex_01/picone_joseph/p01/01/02/f002.txt, value = 2.0000
4: /data/courses/ece_1111/current/exams/ex_01/picone_joseph/p01/03/f003.txt, value = -1.0000
sum = 0.0000
```

Your program should work for any set of files and run correctly from your directory p01. You can assume each file contains exactly one line with one number on it.

Hint: Use the find command to generate a list of files and pass this list as arguments to a program that loops over its command line arguments and sums up the numbers in each file. Your solution should use p01.sh (a shellscript that calls your C program) and p01.exe (a C program that adds the numbers up).

If you can produce the correct answer and sort the file list in lexical order, I’ll give you 10 extra points. Lexical order for the list above means the first file processed above would be f002.txt.

**(50 pts) Problem No. 2:**

Random number generation is one of the more interesting and difficult things to do on a computer. In this exercise, we will write a simple piece of code that generates random values of “0” or “1”. Make sure you use a makefile, put your driver program in p02.cc, your function declaration in p02.h, and your function implementation in p02\_00.cc.

Write a program that does the following:

- (1) Loops for 10 times and calls a function “my\_random” that generates a random number using the algorithm described below. Print the loop counter value and the random number generated. Pass the for loop index as an argument to the function described below (what we call the seed).
- (2) The function my\_random uses this interface:

```
float my_random(long seed);
```

Your function should do the following:

- (a) Multiply the math constant M\_PI by the seed and store the result in a 32-bit float.
- (b) Extract the third least significant bit from this product.
- (c) Also extract the least significant bit from the same product.
- (d) Compute the “exclusive or” of these two bits and return this value as a floating-point number (it will be “1” or “0” of course).

Your output should be (note these are not the actual values, just an example):

```
ece-000_[1]: p02.exe
0: 0.0000
1: 1.0000
...
```