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+rwx,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:

The resources directory contains files with filenames of the form:

```
/data/courses/ece_1111/resources/data/aaaaahun/s006_2009_07_09/02_tcp_le/aaaaahun_s006_t000.csv_bi
```

Write a shell script that takes a path as the first argument (e.g., p01.sh <path>), and a match string as a second argument (e.g, "07_09"). Find all the files in the directory tree that contain the match string somewhere in full pathname and sums up the sizes of these files.

Your output should look exactly like this:

```
ece-000_[1]: p01.sh /data/courses/ece_1111/resources/data/ 07_09
found 99 files that total 99999 bytes
```

where "99" is the number of files that match your search criterion, and "99999" is the total size in bytes.

Test your program using this directory:

```
ece-000_[1]: d /data/courses/ece_1111/resources/data/aaaaahun/s006_2009_07_09/02_tcp_le/
total 12
drwxr-xr-x. 2 picone ece_1111 99 Aug 31 23:26 ./
drwxr-xr-x. 3 picone ece_1111 23 Aug 31 23:26 ../
-rw-r--r--. 1 picone ece_1111 69 Aug 31 23:26 aaaaahun_s006_t000.csv
-rw-r--r--. 1 picone ece_1111 72 Aug 31 23:26 aaaaahun_s006_t000.csv_bi
-rw-r--r--. 1 picone ece_1111 69 Aug 31 23:26 aaaaahun_s006_t000.edf
```

The total size should be $69 + 72 + 69 = 210\ bytes$.

(50 pts) Problem No. 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.

Create a function called:

```
unsigned char myrandom(long N);
```

This function should add the value of the math constant pi N times into a float variable called sum and extract 2 bits from the middle (b_4 and b_5). Return the value of these bits as an unsigned binary coded decimal (which has a range of [0,3]).

In your main program, create a long integer array, a, of length 4, and initialize it to the numbers $[29, 37, 67, 89]$. Create a long integer array, b, that is also length 4. Loop over the array a, and assign the array b using the following assignment statement:

```
b[myrandom(a[i])] = a[i];
```

where "i" is your loop counter.

Print the values of the arrays a and b to stdout:

```
for (long i = 0; i < 4; i++) {
   fprintf(stdout, "a[%1d] = %8ld, b[%1d] = %8ld\n", i, a[i], i, b[i]);
}
```

This is not the most efficient way to randomly shuffle an array, but it should give you some idea of how this can be done.