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:

Shell scripts in Unix can be used to combine a sequence of commands into a single script or command. Start with this simple shell script:

```
#!/bin/sh
echo $1
# end of script
```

“$1” is the first argument in the command line.

Create a script with the name p01.sh by copying the above code into a file named p01.sh. Give the file executable permission so I can run it just by typing “p01.sh”.

Assume all the input files are text files. Add Linux commands to this script that iterate over all the command line arguments and prints out the number of lines in each file. For example:

```
ece-000_[1]: p01.sh /data/courses/ece_1111/current/exams/ex_01/picone_joseph/p01/*.txt
/data/courses/ece_1111/current/exams/ex_01/picone_joseph/p01/x1.txt 3
/data/courses/ece_1111/current/exams/ex_01/picone_joseph/p01/x2.txt 7
/data/courses/ece_1111/current/exams/ex_01/picone_joseph/p01/x3.txt 6
```

Your output should be exactly as shown above – the filename and the line count. You need to process one file at a time in a loop. Calling something like “ls -l $*” will not meet my requirements. Your script should have a loop, call a command for each file in the loop, capture the output of that command, and “pretty print” the output.

(50 pts) Problem No. 2:

Consider the mathematical definition of a polynomial:

$$x(t) = a_1 * t^3 + a_2 * t^2 + a_3 * t + a_4$$

Write a program that prints out values of this function at specific points in time. Your interface should be:

p02.exe start_time stop_time step_size a1 a2 a3 a4

where t starts at $start\_time$ seconds, is incremented by step_size seconds until t reaches $stop\_time$.

For example,

p02.exe 100.0 200.0 10.0 1.0 2.0 3.0 4.0

would print values of the above function for $t = 100, 110, \dots, 200$.

Be careful to compute t using an integer counter rather than "$t += step\_size$" to avoid roundoff errors. Figure out how many points you have to process and perform an integer loop over that number of points. Your loop should terminate when t reaches $stop\_time$.

Print values to stdout using the following format:

```
a1 = 1.0000
a2 = 2.0000
a3 = 3.0000
a4 = 4.0000
time = 100.0000 value = x.xxxx
time = 110.0000 value = x.xxxx
```

Use a make file, a header file and place your main program in a file *p02.cc*.