Name: ____

Problem	Points	Score
1	50	
2	50	
Total	100	

Notes:

(1) 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.

(2) Your code and results should be placed in directories p01, p02 and p03.

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

/data/courses/ece_1111/current/exams/exam_02/lastname_firstname

Set the permissions using "chmod -R u+rwx,g-rwx,o-rwx <lastname_firstname>" so only you have read and write permission to this directory. Create two subdirectories within this directory: p01 and p02. Put ALL your code in these directories. Do not touch your files after the exam is over.

You must use a make file, a header file and a main program file named p01.cc (or p02.cc). All other code needs to go into an implementation file called p01_00.cc (or p02_00.cc).

Problem No. 1: In this section of the class, we learned about string manipulation functions such as those available in <string.h>. Using these tools, write a program to sort the command line arguments as strings in descending order. For example, your program should run something like this:

```
p01.exe aaa zzz mmm
```

would produce a listing of strings:

```
1: zzz
2: mmm
3: aaa
```

Your sort algorithm need not be efficient. You can do a brute force sort (a double loop over the data for example). You do not need to call a library function either to do the sort. You can code it manually.

Test your program using this command:

```
nedc_999_[1]: ls -1r /data/courses/ece_1111/current/exams/exam_02/picone_joseph/p01/*.dat
/data/courses/ece_1111/current/exams/exam_01/picone_joseph/p01/z27.dat
/data/courses/ece_1111/current/exams/exam_01/picone_joseph/p01/q_123.dat
/data/courses/ece_1111/current/exams/exam_01/picone_joseph/p01/m999.dat
/data/courses/ece_1111/current/exams/exam_01/picone_joseph/p01/a123.dat
```

Problem No. 2: Using the program you previously developed that generates and writes a sinewave to a BINARY file as a SHORT INTEGER (16 bits or two bytes), generate a sinewave of amplitude 10,000 to a file. Use a frequency of 100 Hz for the sinewave, a sample frequency of 100 Hz and a duration of 10 seconds (none of the results depend on the sample frequency or frequency of the sinewave). You will have 1,000 points in this file. The signal will range from [-10,000,10,000]. The file will be 2,000 bytes long. Call this file **p02.dat**.

Write a program, p02.cc, that reads the data from this file and computes the minimum and maximum values, and the average value. Also print the number of samples. Use this output format:

```
nedc_999[1]: p02.exe <filename>
file: %s
number of samples = %d
min_value = %15.4f
max_value = %15.4f
avg_value = %15.4f
```

For example:

```
nedc_999[1]: p02.exe p02.dat
file: p02.dat
number of samples = 1000
min_value = -10000.0000
max_value = 10000.0000
```

```
avg_value = 0.0000
```

Your program, **p02.cc**, must work on any binary file containing short integers. Use the file you generated as a test case. I will provide a previously unseen file for testing when I grade the exam. Its length and contents will be arbitrary. Demonstrate your program works by running it on your example file and putting the result in a comment at the top of your main program.

YOU CAN ONLY READ THE INPUT FILE ONCE – you are not allowed to do multiple passes over the input file.