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.

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

/data/courses/ece_1111/current/exams/exam_01

Your directory should be your last name all lowercase (e.g, "picone"). Set the permissions using "chmod u+rwx,g-rwx,o-rwx <lastname>" so only you have read and write permission to this directory. Create two subdirectories within this directory: p01 and 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.

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

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".

Add Linux commands to this script that print out the size of the file. This is what your output should look like:

```
nedc_999_[1]: p01.sh foo.txt
the size of foo.txt in bytes is: 99
```

"99" is the actual file size of foo.txt.

Hint: Use the ls command and pipes.

Problem No. 2: Create a C program that does the following:

```
p02.exe 1 2 3 4
sum = 10.0000, sum is evenly divisible by 2
```

Your program should compile with a Make file and use the following files: (1) p02.cc – main program, (2) p02.h – header file in which you include the necessary header files and function definitions, and (3) p02_00.cc – implementation of your function to check the sum (see below).

Your program should sum the arguments and test if the result is evenly divisible by 2. It should work for any of these test cases as well as other cases like these:

```
p02.exe 27 54.333
sum = 81.33, sum is not evenly divisible by 2
```

```
p02.exe -10.0 10.5
sum = 0.5, sum is not evenly divisible by 2
```

```
p02.exe -1.0 1.0 -2.0 2.00000001 2
sum = 2.0000
sum is not evenly divisible by 2
```

Be careful how you handle floating-point and integer precision, especially for the last test case.

Your implementation should include a main program named p02.cc. You should declare a function named mycheck that takes a floating-point value as an argument and returns true or false depending on whether the argument is evenly divisible by 2. Your main program should test the return value of this function. The function should be stored in a file p02_00.cc. Everything should compile and link with a Make file.

The sum can be computed in your main program by looping over the command line arguments. Note that the variable “argc” contains the number of command line arguments, and the main program is always the first argument (argv[0]).

Summary: Work directly in your workspace so that you don’t lose any time transferring files. Your source code should be clean and commented. If I can’t read it and understand it, your grade will suffer. Comment as you go along – it will help you understand what you are doing.