

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, followed 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:

Shell scripts in Unix can be used to process command line arguments. Starting with this script:

```
#!/bin/sh
for var in "$@"
do
    echo "$var"
done
```

write a shellscript that does the following:

for each command line argument:
 treat the argument as the root node in a directory
 search the directory tree for all files that have "aaa" in their filename
 count the total number of lines in each file that you find

For example, your command should pass these test cases:

```
ece-000_[1]: p01.sh /data/courses/ece_1111/current/exams/ex_01/picone_joseph/aaa
total lines = 2
ece-000_[1]: p01.sh /data/courses/ece_1111/current/exams/ex_01/picone_joseph/aba
total lines = 2
ece-000_[1]: p01.sh /data/courses/ece_1111/current/exams/ex_01/picone_joseph/aca
total lines = 0
ece-000_[1]: p01.sh /data/courses/ece_1111/current/exams/ex_01/picone_joseph/
total lines = 6
```

You might find this code useful:

```
num1=$1
num2=$2
sum=`expr $num1 + $num2`
echo $sum
```

This adds two numbers in shell.

(50 pts) Problem No. 2:

Write a program that does the following:

- Creates an array, a, of N elements, where N is specified from the command line
- Initializes the values of the array from 0 to N-1
- Prints the values to stdout:

```
ece-000_[1]: p02.exe 3
before:
a[0] = 0.0000
a[1] = 1.0000
a[2] = 2.0000
```

- Reverses the values of the array and prints the reversed values to stdout:

```
after:
a[0] = 2.0000
a[1] = 1.0000
a[2] = 0.0000
```

Note that your code must work for any values in a[]. To test your code, I will initialize the values of a[] to some arbitrary, or random, values.

To reverse the array, implement a function called myreverse:

```
bool myreverse(float* out, float* in, long N)
```

where N is the number of elements in the array. Be careful to do memory allocation in your main program for this problem – you don't have to do memory allocation in `myreverse()`.