Name:

|  |  |  |
| --- | --- | --- |
| Problem | Points | Score |
| 1 | 30 |  |
| 2 | 20 |  |
| 3 | 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, 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 the process of demonstrating you have basic Linux literacy.

1. Your code must be nicely formatted and well commented, or I will deduct at least 10 points per problem.
2. 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.

**Problem No. 1**: The directory tree:

*/data/courses/ece\_1111/current/resources/reports/354*

contains 1000 files with an extension “\*.txt”. Each filename has a basename in format “########\_s##.txt” (e.g., “00009677\_s01.txt”). Write a Unix shellscript that counts the number of files that are from the year 2009 (e.g., “/354/0354987/s01\_2009\_12\_23/00002539\_s01.txt”) and have the number 25 in the basename (e.g., “00002539\_s01.txt”). Note that if the number “25” appears elsewhere in the full pathname, such as in a directory name, it should not be counted. Only files with “25” in the basename should be counted.

Put this shellscript in a directory *p01,* and name it *p01.sh*. It should take as an argument the directory path:

*cd <your exam directory>*

*./p01/p01.sh /data/courses/ece\_1111/current/resources/reports/354*

It should have executable permissions. Place your output from this command in a file named *p01/p01.txt*.

**Problem No. 2**: We have seen that the Unix command “grep” is extremely useful. Consider the following command:

*grep -I picone /etc/passwd*

This outputs all lines with “picone” somewhere in them:

*ece-000\_[1]: grep picone /etc/passwd*

*picone:x:1002:1002:Joseph Picone, picone@temple.edu:/home/picone:/bin/bash*

Modify this command so that the output only contains the name of my home directory (“*/home/picone*”). Place your command and its output in a file called *p02/p02.txt*.

**Problem No. 3:**

Write a C program that reads a text file line by line, sums up the character values on the line, and prints out the line number and the sum. For example:

./p03/p03.exe p03.txt

Should produce this result:

*ece-000\_[1]: more p03.txt*

*1 2 3*

*4 5 6*

*ece-000\_[1]: p03.exe p03.txt*

*1: 214*

*2: 437*

To avoid counting the linefeed at the end of the line, use this loop:

*while (buf[i] != '\n') {*

 *…*

*}*

Or the equivalent. This loop iterates over the line until it reaches a linefeed.

You must use a make file and follow our standard protocol for developing code like this (e.g.. use our makefile template, comment code, space code properly, etc.