Name:

Deposit your quiz solution in this directory:

/data/courses/ece\_1111/current/quizzes/qu\_02/lastname\_firstname

Make sure the permissions are set properly on this directory, which we refer to as the parent directory, *so only you (and a superuser) can view the contents*. Create two subdirectories, *p01* and *p02*, for your solutions to each of the problems below. There should be no files in the parent directory.

1. (50 pts) In this problem, you are going to write a bash shell script, *p01/p01.sh*, that processes this file:

/data/courses/ece\_1111/resources/data/text/data\_v01.txt

Your script should do the following:

* split the file into words (a word is defined as text separated by a space or a newline character)
* convert the words to uppercase
* sort the words in alphabetical order
* count the number of occurrences of each word
* print the sorted list to the terminal showing the word and the number of times it occurred

Verify that your answer matches what you get when you manually do this task. Your answer should look something like this:

4 ALEX

2 BOB

2 JOE

2 JORDAN

2 MARY

Once you have completed this script, make sure it runs on this file:

/data/courses/ece\_1111/resources/data/text/data\_v00.txt

This is the file I’ll use to evaluate your solution assuming you get the test file correct. Your script must take a filename as an argument:

p01/p01.sh any\_file\_i\_want.dkdkdk

should work assuming the second argument, which is the filename, exists. I should be able to run your script from any directory, such as */home/picone*, and provide any filename as input. The script must have the proper permissions set so that it is executable.

Finally, you can use AI tools or Google search to solve parts of this problem, but you must be able to explain how your code works.

1. (50 pts) In a subdirectory *p02*, write a Python program, not a script, in a file named *p02*, that does the following:

- prints the current data and time to the terminal (stdout)

- prints the number of lines in your file (hint: look at len() and readlines())

For example:

ece-000\_[1]: p02/p02.py

date: Fri Sep 5 08:07:53 EDT 2025

number of lines: 27

Your output should look exactly like the above though the format of the time and data can vary depending on what tool you use. Nothing should be hardcoded. The number of lines must be the actual length of the file. The length of the file can be checked using the Unix command “wc -l p02/p02.py”.