Homework Assignment 4 (100 points)
COP4342 Unix Tools
Due: 11/12/2019

Part I. Fibonacci number F(N) for an integer value N is defined as follows. F(N) = 1, if N = 0 or N = 1, and F(N) = F(N-1) + F(N-2), for N > 1. For a given value N, your program should compute and output the Fibonacci number F(N). When the input value of N is a negative value, your program should terminate. **You must implement your program using recursive subroutines**. An example run of the program is given below. (50 points)

```
duan@linprog3 (~...homework/solutions) % perl fibonacci_recursive.pl
Please enter value of N: 0
f(0) = 1
Please enter value of N: 1
f(1) = 1
Please enter value of N: 2
f(2) = 2
Please enter value of N: 3
f(3) = 3
Please enter value of N: 4
f(4) = 5
Please enter value of N: 5
f(5) = 8
Please enter value of N: 10
f(10) = 89
Please enter value of N: 20
f(20) = 10946
Please enter value of N: 30
f(30) = 1346269
Please enter value of N: -1
duan@linprog3 (~...homework/solutions) %
```

Part II. In this part of the assignment, you need to implement a simple "wc" program. (Please read the manual page of the word count wc Unix command to get more information on command wc.) Given a file as a command-line argument, you need to report the size of the file in three aspects: the number of characters in the file, the number of words in the file, and the number of line of the file. A word is defined as a non-empty sequence of consecutive characters in the file that are not whitespaces. In addition to these three statistics of the file, you should also report the number of times that each unique word appears in the file. In the output, words should be sorted in ascending ASCII code order. An example run of the program is given below.

```
duan@linprog3 (~...homework/solutions) % perl wc.pl test.txt
Number of characters: 71
Number of words: 12
Number of lines: 5
```

```
Frequency of words in the file:
-------------------------------
#test: 1
(test: 1
123test: 1
This: 1
a: 1
again#: 1
is: 1
second: 1
test: 2
test123: 1
test2): 1
duan@linprog3 (~...homework/solutions) %
```

Hint: Given a string, you can use the function split() to obtain all the words separated by whitespaces in the string. For example

my $str = “This is a test”;
my @words = split /\s+/, $str;

Now @words contains four elements, “This”, “is”, “a”, and “test”.

Submission: Tar your programs into a single file and submit online on Canvas. Make sure you tar your programs correctly. You are responsible for incorrect submissions (for example, empty tar file). You can untar the file **under a different directory** to make sure that you do include both programs in the tar file. Same late policy applies if the submission is incorrect and you need to submit a new version.