Submit a zipfile to the Homework 9 assignment within the Course Documents section on Blackboard. The zipfile should contain four C source files, each with the same name as the program specified (e.g. nprimes.c).

All of your programs should validate their input and generate an error if the input is invalid. Your programs should never crash if the input is missing or malformed. If your program generates an error, it should print a message to stdout and return a non-zero exit code. Check the website for example input and output.

I expect to see nicely indented, commented code. Comments should include at the very least, your name, the purpose of the program, and an explanation of the algorithm. Your program should compile without warnings or errors when compiled with “gcc -Wall”.

1) Write a program called nprimes which takes a single integer input n on the command line. It should produce as output the first n prime numbers, one on each line. For example, running “nprimes 4” should produce the output 2 3 5 7, with each number on its own line.

2) Write a program called stats which takes its input from either a file named on the command line, or from standard input. The input should consist of real numbers, both positive and negative, separated by spaces and/or newlines. The program’s output should consist of three lines: the first line should print the minimum number input, the second should print the maximum, and the third should print the average of all the numbers input. Use the %g or format specifier to print the min, max, and average. The program should generate an error if the input contains no numbers.

3) Write a program called bigsum which can sum a set of numbers larger than the maximum long integer size (about 19 decimal digits). Your program should take its input either from a file named on the command line, or from standard input. The input should consist of non-negative integers of up to 32 digits, separated by spaces and/or newlines. The output is a single number equal to the sum of all of the inputs. Your program should generate an error if any input is longer than 32 digits, or if the output cannot be stored in a 32 digit number.

4) Write a program called wordsearch which takes its input either from a file named on the command line, or from standard input. The input should consist of two parts: first, a block of m lines of n uppercase letters, with m and n both less than 256; then a single blank line, followed by a set of words to search for, one on each line, in upper or lower case. Each word should be hidden within the original block of text, horizontally, vertically, or diagonally, forwards or backwards. Your program should output for each word, the starting and ending row and column where the word can be found, with (0,0) being the top-left position, or indicate whether the word was not found.