CSCI 3232 Systems Software Assignment 5

Upload all your files to the correct dropbox folder in Folio before the deadline --- **11:30PM, Mar 12, Tuesday, 2019.**

**Note: Make sure you have practiced all relevant sample codes in slides and Folio’s example codes before you start this assignment. Make sure your scripts have Unix style line endings. See slide 32 of 6\_shell\_basics.pptx for details. About makefile submission issue in Folio, see last slide in 3\_Pointers\_Functions.pptx.**

1. (20 pts) Write a bash script **A5p1.sh** to (1) print out all command line arguments, one per line (each argument is assumed to be an integer); (2) then print out the same command line arguments in numerical order, one per line. Do not call any existing Linux utilities such as *sort*. (When grading, no more than three command line arguments will be provided for your script.) For example, if your script is invoked as “**A5p1.sh 33 222 11**”, then it can print the following:

33

222

11

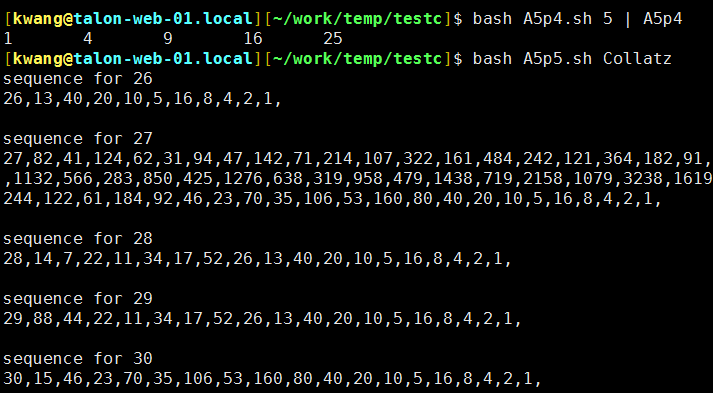
After sorting in numerical order:

11

33

222

1. (15 pts) Write a bash script **A5p2.sh** to print out 15 equations on 15 lines, one per line. Each of these equations should be of the form i\*j=k, where i is an integer that runs from 1 through 3, j runs from 1 through 5 and k is the product of i and j. (For example, if i is 2 and j is 3, k should be 6.) You will print 15 *different* equations because the number of possible combinations of i and j is 3\*5=15. Your script should use loop to achieve the functionality. Do not define or use any functions in this script.
2. (15 pts) Write a bash script **A5p3.sh** to redo problem 2. This time you need to define a function that takes two integer parameters a and b and prints out an equation of the form a\*b=c, where c is the product of the two parameters a and b. The main part of your script should use loop and call the function you defined. That is, you should call the function 15 times to display the 15 equations.
3. (20 pts) Write a bash script **A5p4.sh** that accepts one command line argument which is supposed to be a positive integer *n*. The script should print all integers from 1 through *n*. Write a C or C++ program **A5p4.c(pp)** to read from stdin as many integers as there are available and then print the squares of all these integers. You should test your code by **“./A5p4.sh <n> | ./A5p4**” assuming the compiled C/C++ program is A5p4. Submit both files.
4. (30 pts) Recall that in previous homework, you have written a C/C++ program that outputs a sequence of integers starting from an input integer supplied by the user on the command line and ending with 1 based on the iteration rule f(x)=3x+1 if x is odd and f(x)=x/2 if x is even. Write a bash script **A5p5.sh** to call this C/C++ program with **all** integers from 26 through 30 on the command line. That is, you call your C/C++ program 5 times in your script **A5p5.sh** using a loop, each time supplying a unique integer between 26 and 30 (inclusive) on the command line for your C/C++ program. Make sure your 5 output lists of integers are clearly separated. Please also submit your C/C++ program source file (which need not be the version you previously submitted but should be named **A5p5.c(pp)**). You can choose whether to hardcode your C/C++ program name in your **A5p5.sh** script or not. When grading, we will first type **make** and then invoke your **A5p5.sh** in a suitable way based on whether your C/C++ program name has been hardcoded in your script or not. Partial outputs of testing the programs in problems 4 & 5 are shown below:



Submit a working makefile to compile C/C++ programs in problem 4 & 5. Make sure your makefile works by testing it. Without a working makefile, up to 4 pts could be deducted.

Checklist for files to be submitted: A5p1.sh, A5p2.sh, A5p3.sh, A5p4.sh, A5p4.c(pp), A5p5.sh, A5p5.c(pp) and makefile for problem 4 & 5.