CSCI 3232 Systems Software Assignment 6

Upload all your files to the correct dropbox in Folio before the deadline --- **11:30PM Mar 27, Wednesday, 2019.**

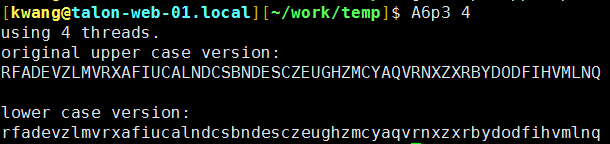
**Note: Make sure you have practiced all sample codes in slides and Folio’s example codes before you start this assignment. Make sure your scripts have Unix style line endings. See slide 32 of 6\_shell\_basics.pptx for details. About makefile submission issue in Folio, see last slide in 3\_Pointers\_Functions.pptx.**

1. (15 pts) Write a bash script **A6p1.sh** to output the number of files and subdirectories separately in the directory that is specified as the first command line argument to this script. **Do not** count recursively in subdirectories. **Do not** call any existing Linux utilities such as “ls”. You may refer to the *testSearch.sh* example and page 7 of the “*Basic Bash Scripting Lab.docx”* documentation in Folio to find out how to test if an item in a directory is a file or subdirectory. When grading, we will test your script using “./**A6p1.sh <dir>**” where <dir> can be any absolute (those starting with ‘/’) or relative directory (those not starting with ‘/’). A sample output can look like (You do NOT need to submit screen shots. Instead submit your source file.):

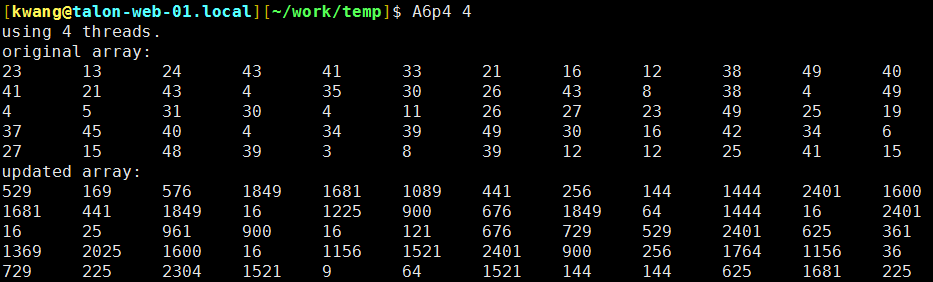
Number of files in <dir>: 12

Number of subdirectories in <dir>: 5

1. (15 pts) Write a bash script **A6p2.sh** that accepts one command line argument a which is an integer between 1 and 50 inclusive. Your script should output a list of integers starting from a and ending with 1 according to the iteration rule you are already familiar with (f(x)=x/2 if x is even; f(x)=3x+1 if x is odd). Don’t call any external programs. Implement all algorithms in your script. When grading, we will test your script using “./**A6p2.sh <integer:1~50>**”.
2. (35 pts) Write a C or C++ program **A6p3.c**(**pp**) that accepts one command line argument which is an integer *n* between 2 and 6 inclusive. Generate a string of 60 random upper case English characters and store them somewhere (e.g. in a char array). Use pthread to create *n* threads to convert the string into all lower case. You may call the *toupper* or *tolower* functions declared in the header *ctype.h*. You should divide this conversion task among the *n* threads as evenly as possible. Print out both the upper and lower case versions of the string on separate lines. That is, 60 upper case characters on one line and 60 **corresponding** lower case characters on another line. Hint: it is dangerous to have printing code in your thread function(s). You may refer to testThread.c, testThread2.c, pthread\_ex1.c, pthread\_ex2.c for examples. Note: if you do not use pthread to divide the conversion task among the threads, you may get zero points. A sample run of the program is shown below. You do NOT need to submit screen shots. Instead submit your source file.



1. (35 pts) Write a C or C++ program **A6p4.c**(**pp**) that accepts one command line argument which is an integer *n* between 2 and 4 inclusive. Generate 60 random integers between 1 and 49 inclusive and store them in a 5 by 12 two dimensional integer array (e.g.,int a[5][12];). Use pthread to create *n* threads to square all array elements. You should divide this update task among the *n* threads as evenly as possible. Print the array before the update and after the update separately as 5 by 12 matrices. Note: if you do not use pthread to divide the update task among the threads, you may get zero points. A sample run of the program is shown below. You do NOT need to submit screen shots. Instead submit your source file.



Include a single makefile to compile your programs in both 3 and 4. Up to 4 points will be deducted if you don’t provide a working makefile. When grading problems 3 & 4, we will type **make** (after changing the name of your submitted makefile if necessary) and then type **./A6p3** **<n>** and **./A6p4 <n>**.

Checklist of files to be submitted: A6p1.sh, A6p2.sh, A6p3.c(pp), A6p4.c(pp), makefile