CSCI 3232 Systems Software Assignment 6

Upload all your files to the correct dropbox in Folio before the deadline --- **11:30PM Mar 12, Sunday, 2017.**

1. Modify the *testSearch.sh* script in Folio Bash scripting examples into **A6p1.sh** to output the number of files and subdirectories separately in the directory that is specified as the first command line argument to the script. Do not count recursively in subdirectories. You may refer to 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. (15 points) (When grading, the TA can test your script using “**A6p1.sh <dir>**” where <dir> can be any absolute or relative directory path.) Your output can look like:

Number of files in <dir>: 12

Number of subdirectories in <dir>: 5

1. Write a bash script **A6p2.sh** that accepts one command line argument a which is an integer between 1 and 50. Output a list of integers starting from a and ending with 1 according to the 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. (15 points) (When grading, the TA can test your script using “**A6p2.sh <integer>**”.)
2. Write a C or C++ program **A6p3.c** (or **A6p3.cpp**) that accepts one command line argument which is an integer *n* between 2 and 4 inclusive. Generate a string of length 60 which contains random lower case English characters and store them in a variable of suitable type of your choice. Use pthread to create *n* threads to convert the string into all upper case. (You may call the toupper and tolower functions declared in ctype.h). You should divide this conversion task among the *n* threads as evenly as possible. Print out both the lower case and upper case version of the string in separate lines. You may refer to testThread.c, testThread2.c, pthread\_ex1.c, pthread\_ex2.c for example codes. (35 points) (Note: if you do not use pthread to do the conversion, you will get zero points.)
3. Write a C or C++ program **A6p4.c** (or **A6p4.cpp**) that accepts one command line argument which is an integer *n* between 2 and 4 inclusive. Generate 60 random integers in the range of [1,99] and store them in a 5 by 12 two dimensional integer array (say, a[5][12]). Use pthread to create *n* threads to update these numbers according the following rule: if a number is even increment it by one; if a number is odd decrement it by one. 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 two dimensional arrays. You may refer to testThread.c, testThread2.c, pthread\_ex1.c, pthread\_ex2.c for example codes. (35 points) (Note: if you do not use pthread to do the update, you will get zero points.)
4. Include a makefile to compile your programs in 3 and 4. (5 points will be deducted if you don’t provide a working makefile.) When grading, the TA will type **make** (after changing the name of your submitted makefile if necessary) and then type **./A6p3** **<n>** and **./A6p4 <n>**.