CSCI 3232 Systems Software Assignment 3

Upload all your files to the correct dropbox in Folio before the deadline --- **11:30PM Jun 27th, Sunday, 2021. Note: Put all your codes into separate TEXT files with appropriate file extensions (.h, .c, etc) as you would compile them. If Folio complains, you can add .txt extension to the files. For example, myfile.c.txt, myfile.sh.txt, makefile.txt.**

1. (40 pts) Follow the example C program in Folio consisting of three files (a.c, a.h, main.c) and write a C program that generates a user specified number of random lower case English characters and prints these characters, offsets them in place, and then prints the offset characters. Here “offset” means changing ‘a’ to ‘b’, ‘b’ to ‘c’, etc, and ‘z’ to ‘a’. The program should consist of three files ***mysource.h***, ***mysource.c*** and ***myMain.c.*** The file myMain.c should contain only the definition of main, which must call three additional functions to accomplish the task. The three functions to generate the characters, offset the characters and print the characters should be declared in mysource.h with the following prototype and defined in mysource.c.

void generateChars(char \*myarr, int len);

void offsetChars(char \*myarr, int len);

void printChars(char \*myarr, int len);

The user provided number of characters to be generated must be obtained from stdin (when your Eagle ID is odd) or from the first command line argument (when your Eagle ID is even). Review slides or example codes in Folio or previous homework assignments to learn about how to do this. You need to submit the above mentioned three files with the specified names. A sample run can look like the following:

[kwang@computer][~/temp]$ ./A3p1 15

Original random string:

uetyepdmsgbedyb

Offset string:

vfuzfqenthcfezc

2. (25 pts) What is the output of the following C program? You should be able to tell the answer without compiling or running it.

int main(int argc, char \*argv[]) {

char a[26], t, \*p, \*q;

int i;

for(i=0;i<26;i++) a[i]='A'+i;

p=a; q=a+25;

while(p<q) {

t=\*p; \*p=\*q; \*q=t;

p++; q--;

}//while

for(i=0;i<26;i++) printf("%c",a[i]); printf("\n");

return 0;

}//end of main

3. (33 pts) Write a C program A3p3.c that accepts two command line arguments *m* and *n* (assumed to be positive integers between 3 and 6 inclusive). The program should allocate a dynamic 2D int array to store the elements of an *m* by *n* matrix, whose elements are randomly generated integers between -50 and 50 inclusive. The program should create a second dynamic 2D int array to store the same matrix elements as in the first 2D array but in reverse row order. After that, the program should print the contents of the two dynamic 2D arrays in matrix format. A sample run can look like the following:

[kwang@computer][~/temp]$ ./A3p3 4 6

first matrix:

-43 30 -46 -5 22 41

22 -37 35 23 -5 -40

-28 -32 -27 14 -39 -29

-2 -33 33 -18 43 -24

second Matrix:

-2 -33 33 -18 43 -24

-28 -32 -27 14 -39 -29

22 -37 35 23 -5 -40

-43 30 -46 -5 22 41

Write a single makefile to compile all programs for problem 1,3 (up to two points will be deducted if you don’t supply a working makefile). Use A3p1 and A3p3 as the output program file names in the makefile.

Checklist of the six files to be submitted: one solution file A3p2.txt, mysource.h, mysource.c, myMain.c, A3p3.c, makefile