Lab 4: C - More with Strings

Chapter 8 – C Characters and Strings

20 points total

All Programs Must Have:

Header Comments and Inline Comments
Consistent Indentation and Spacing

See the Documentation and Style Guidelines.

All programs must compile. Programs that do not compile will receive a grade of zero.

Lab 4: More with Strings

- Make a new folder for lab4.
- Write this in C. Create a header file functions4.h with the prototypes for the two functions shown below.

  Include a preprocessor wrapper.

1. void splitAlpha (const char * original, char * lower, char * upper);

   The variables original, lower, and upper are pointers to null-terminated strings (character arrays). The function splitAlpha should copy the lowercase letters from original to lower, and the uppercase letters from original to upper.

   The memory areas pointed to by lower and upper are assumed to be large enough to hold the number of characters that need to be moved.

   The string original may contain non-alphabetic characters.

   The contents of original should not be changed.

   For example, if original contains "The symbol for Intel is INTC."

   After the function completes, lower should contain "hesymbolforntelis" (remember to put in the null character) and upper should contain "TIINTC" (null character at the end).

2. void printSequences (const char * text);

   The function printSequences should print any sequences of letters or digits that occur in the string. At the end of each sequence, print a newline.

   For example, if text contains "abk123@XY", the function should print

   ab
   123
   XY

   Letters must be the same case in order to form a sequence. "rS" would not be a sequence.

   A sequence must be at least two characters. A single letter or a single digit doesn't form a sequence.

- Create functions4.c with the function definitions for splitAlpha and printSequences.

- Create lab4.c

  Write a main function that tests the functions splitAlpha and printSequences.

- Create a makefile that compiles functions4.c and lab4.c to produce an executable called lab4.

  Word of Advice: Type the makefile without looking at another makefile. You will need to do this for the exam.

- Test thoroughly.

Zip 4 files: functions4.h, functions4.c, lab4.c, and the makefile.

Note: the only acceptable format is .zip

Submit the zip file on Canvas - Lab Assignment 4.

Grading Criteria:

Documentation and Style	3 points
Makefile	3 points
splitAlpha	5 points
printSequences	5 points
lab4 main function	4 points