CSE2312 (Fall 2022)
Homework #4

Notes:
With this homework, we write assembly string functions for the RPi 3b/3b+/4b.

All numbers are in base-10 unless otherwise noted.

If part of a problem is not solvable, explain why in the answer area.

The target date to complete this homework set is October 18, 2022.

This homework set will not be graded, but please solve all of the problems to prepare for the quizzes and exams.

1. Explain why the R4 register is pushed and popped in the strfindn function coded in class. In general, understand how you handle the case where more than R0-3 are needed. Also understand why you have to push the LR register before calling a function from a function.

2. Write assembly functions that implement the following C functions:

a. bool isStrEqual(const char str1[], const char str2[])
   // returns true (1) if the strings match, false (0) otherwise

b. void strConcatenate(char strTo[], const char strFrom[])
   // appends the contents of string strFrom to strTo
   // note: strTo must have enough space to hold the contents of strFrom and strTo

c. void leftString(char * strOut, const char * strIn, uint32_t length)
   // input: array (strIn) containing the input string, and the number of characters to extract (length)
   // output: array (strOut) containing up to, but not exceeding length number of strIn characters from the start of the array
   // strIn = ‘abcdef’, length = 5 → returns strOut = ‘abcde’
   // strIn = ‘abcdef’, length = 7 → returns strOut = ‘abcdef’

d. int16_t decimalStringToInt16(const char * str)
   // convert the null-terminated string (str) to a signed 16-bit integer
   // treat the string as representing a decimal number
   // if a character other than 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, or – is present or the value is out of range, return 0
   // the -, if present after the first character should cause the function to return 0

e. uint8_t hexStringToUint8(const char * str)
   // convert the null-terminated string (str) to an unsigned 8-bit integer
   // treat the string as representing a hexadecimal number
   // if a character other than 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, or F is present or the value is too large, return 0.