ICS Lab Report #2

StuID: **Name:**

Problem Setting

Here we are using the LC3 assembly language to solve a problem: do 2 strings contain the same characters(namely anagrams), with no consideration of spaces. For example, "dormitory" and "dirty room" are anagrams. All upper case characters should be considered the same as lower case.

Algorithm Specification

Here we present a pseudocode form of our algorithm.

```
for i = 0 to len_str1
  if(str1[i] == space) continue
  if(str1[i] is upper-case) str1[i] <- str1[i] + 32
  arr[str1[i]] <- arr[str1[i]] + 1
for i = 0 to len_str2
  if(str2[i] == space) continue
  if(str2[i] is upper-case) str2[i] <- str2[i] + 32
  arr[str2[i]] <- arr[str2[i]] - 1
for i = 'a' to 'z'
  if(arr[i] != 0) halt with NO
halt with YES
```

This main idea of the program:

Count all occurences of characters, compare them if they are completely the same. If not the same, result is NO, otherwise YES.

LC3 Implementation

Using R7 + R1(start + offset) to represent the array pointer starting from x3200, and from R0 get the current character. By the way tests blank and upper-cases.

Here is for str1. for str2, replace ADD R6, R6, #1 with ADD R6, R6, -1.

```
LDI R0, lstr1
LOOP1
LDR R1, R0, #0 ;load the character
ADD R1, R1, #0
BRZ OUTLOOP1 ;if zero, means the string finished
LD R7, N_BLANK
ADD R7, R1, R7
BRZ RELOOP1 ; if blank, continue
LD R7, N_GRT
ADD R7, R1, R7
BRP NONTRANS1
ADD R1, R1, #8
ADD R1, R1, #8
ADD R1, R1, #8
ADD R1, R1, #8 ;if upper-case, add 32
NONTRANS1
LD R7, SPACE
ADD R7, R7, R1
LDR R6, R7, #0
ADD R6, R6, #1
STR R6, R7, #0 ;load arr[R1], add 1, and store
RELOOP1
ADD R0, R0, #1
BRNZP LOOP1 ;add the string offset, and continue next loop
OUTLOOP1
```

Check Problem

Q: How to find a upper-case character and translate it into lower-case?

A: Compare it with x0060 by add and branch. If lower than x0060 means the character is upper-case, then add it with 4 8's.

Q: If given a character [(in ASCII z+1), how do the count loop perform?

A: Add arr['['] with 1. However this memory won't be used in comparation.