Davy Nolan

Assignment #2

Date: 22nd of December 2017

DAVY NOLAN Cs1021 ASsignment #2

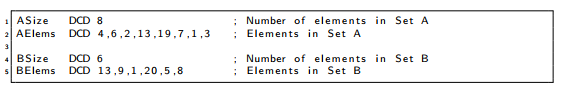
Memory

**Stage 1: Console Input**

**Aim:**

The aim of this stage of the assignment is to “design and write an ARM Assembly Language program that will create a third set, C, that is the symmetric difference of A and B.”

The following ARM Assembler directives illustrate how the sets are arranged:



**Solution:**

Set A = {4, 6, 2, 13, 19, 7, 1 ,3}

Set B = {13, 9, 1, 9, 5, 8}

**A** **B**

4. 6. 2.

1. 9. 9.

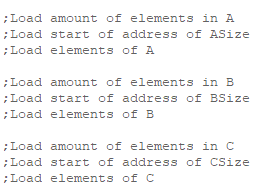
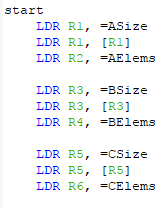
19. 7. 13. 5. 8.

3.

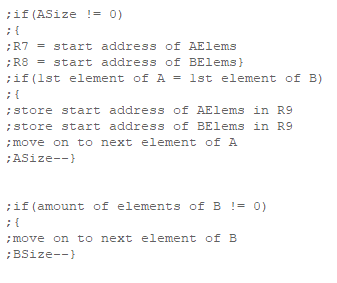
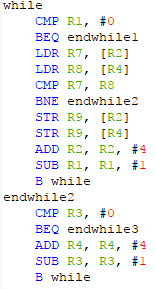
Set C = Symmetric Difference = {4, 6, 2, 19, 7, 3, 9, 9, 5, 8}

It was in imperative that the sets were loaded into the program. Pseudocode was prepared in order to do so:

Pseudocode:



Next, pseudocode was prepared to explain how to check each element of sets A and B and to compare them together. The “ASize” and “BSize” strings were used as counts and each time the program moves on to the next element, it decrements the corresponding count.



The program then continued to find the symmetric difference of sets A and B which is all of the elements that they don’t have in common.

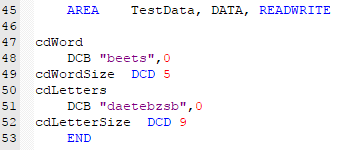
**Stage 2: Countdown Checker**

**Aim:**

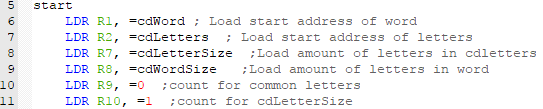
“Design and write an ARM Assembly Language program to determine if one string, A, can be formed from the nine letters contains in a second string, B.”

**Solution:**

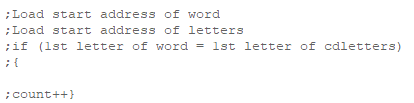
First of all, “cdWordSize” and “cdLetterSize” were added to the program to act as counts. These were set as values 5 and 9 due to the 5 letters being in the word “beets” and 9 in “daetebzsb”.

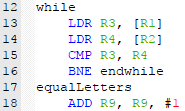


Just like in stage 1, the program begins with all the registers being loaded. Counts were also loaded into register R9 and R10.



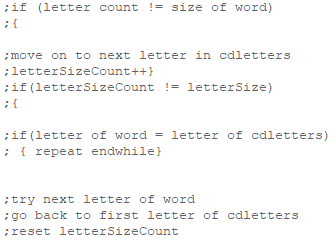
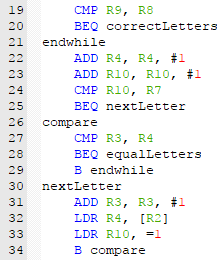
Pseudocode was prepared to explain how to go about beginning this program.





The program then had to go on and check for each letter in the 9-letter string and compare each letter to the first letter in the 5-letter word. If a letter matches, then the count increases by 1 until reaching 5. When the count reaches 5, we know that the word is legit and can be successfully created from the given 9 letters.

The program then must go on and check for the next letter in the word. A pseudocode was written up to do all of this:



The program also has to store a value of 1 in R0 if the word could be created from the given letters. I did this by stating in the program that if the letter count is equal to the word size, then branch to “correctLetters” where the program then loads the value 1 into R0.





**Methodology:**

The program appears to have an error in it as when it is run for the given example, 1 is not the value in R0.

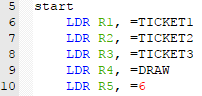
**Stage 3: Lottery**

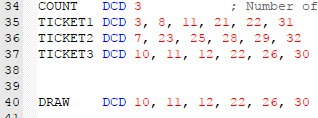
**Aim:**

“Design and write an ARM Assembly Language program that will determine the number of tickets that match four numbers, five numbers and six numbers. (i.e. your program should produce three result values for the number of “match four” tickets, “match five” tickets and “match six” tickets.)”

**Solution:**

To make this program more functional and easier to work with, the “TICKET” string was split into three strings “TICKET1”, “TICKET2”, and “TICKET3”. All these values were loaded into the registers at the beginning of the program. The “DRAW” string containing the winning numbers was also loaded at the beginning of the program. A number count of 6 was also loaded as each ticket consists of 6 numbers.





I experienced a lot of trouble with the rest of stage 3 of this assignment so therefore the rest is unfinished.