1 Purpose

The purpose of this program is to compare two strings and return the difference between those two strings:

1. If a **NULL** character is found, the function returns zero as both strings will be the same.
2. If the ASCII value of the character of the rst string is greater than that of the second string, then the positive difference (> 0) between their ASCII values is returned.
3. If the ASCII value of the character of the rst string is less than that of the second string, then the negative difference (< 0) between their ASCII values is returned.

There are some examples:

1.1 Example 1

String 1

Address	x3100	x3101	x3102	x3103	x3104	x3105
Value(Hex)	x0044	x0073	x0054	x0041	x0073	x0000
Character	D	s	T	A	s	NULL

String 2

Address	x3200	x3201	x3202	x3203	x3204	x3205
Value(Hex)	x0044	x0073	x0074	x0041	x0000	
Character	D	s	t	A	NULL	

Result

Address	x3300
Value(Hex)	xFFE0

2 Principles

In order to compare those characters, we need to calculate the difference of there ASCII code, if two of them are same, the result will be 0. And we need to **tranverse** those two strings, so I did follow steps:

Compare Get the negative version of the second character's ASCII code, then subtract the first character by the negative number, if the result is 0, then they are same characters.

Tranverse In order to tranverse those strings, I need do some optraions about the address pointers(Some registers storing some addresses). So I load the address of the first character of a string into a register (R0 as an example). ADD the register by 1 after one loop, it makes the program to fetch the second character of that string at the beginning of the next loop.

There are some steps that I have done in my program to achieve the target:

Step 1 Load the address of the first characters of two strings. (LD R1, S1_ADDR ; LD R2, S2_ADDR)

Step 2 **Compare** the two characters, if they are **not** same, jump to Step 4.

Step 3 Add the address register bt 1, jump to Step 2.

Step 4 Store the difference of the characters.

3 Procedure

Once I finished my program, I found that the result is not as expected, so I used the "Step in" function to check my instructions. Finally, I found that I used the wrong register at the end of my program. After I changed that, the result begame right.

4 Results

The result of my program are as follows:

4.1 Result 1

Address	x3100	x3101	x3102	x3103	x3104	x3105
Value(Hex)	x0044	x0073	x0054	x0041	x0073	x0000
Character	D	s	T	A	s	NULL

Address	x3200	x3201	x3202	x3203	x3204	x3205
Value(Hex)	x0044	x0073	x0074	x0041	x0000	
Character	D	s	t	A	NULL	

Address	x3300
Value(Hex)	xFFE0

4.2 Result 2

Address	x3100	x3101	x3102	x3103	x3104	x3105
Value(Hex)	x0044	x0073	x0054	x0041	x0073	x0000
Character	D	s	T	A	s	NULL

Address	x3200	x3201	x3202	x3203	x3204	x3205
Value(Hex)	x0044	x0073	x0054	x0041	x0000	
Character	D	s	T	A	NULL	

Address	x3300
Value(Hex)	x0073

Obviusly the results of my program are as expected.