Computer Organization Lab 1

SPIM MIPS Simulator

1. Introduction: SPIM is a MIPS processor simulator that can simulate the execution of MIPS assembly program. The goal of this project is to let you get familiar with the environment of SPIM and programming with MIPS assembly language.

2. Problem: Input two strings, where the first string is the main string while the second string is a pattern string. First reverse the main string and replace the main string as the reversed one. Secondly compare the pattern string with the new main string and check how many substrings in the new main string match the pattern string. Two substrings can overlap. For instance, two substrings main_string[1..4] and main_string[3..6] both match the pattern string and have two characters, main_string[3] and main_string[4], in common. Then we have to count in these two substrings as matched substrings.

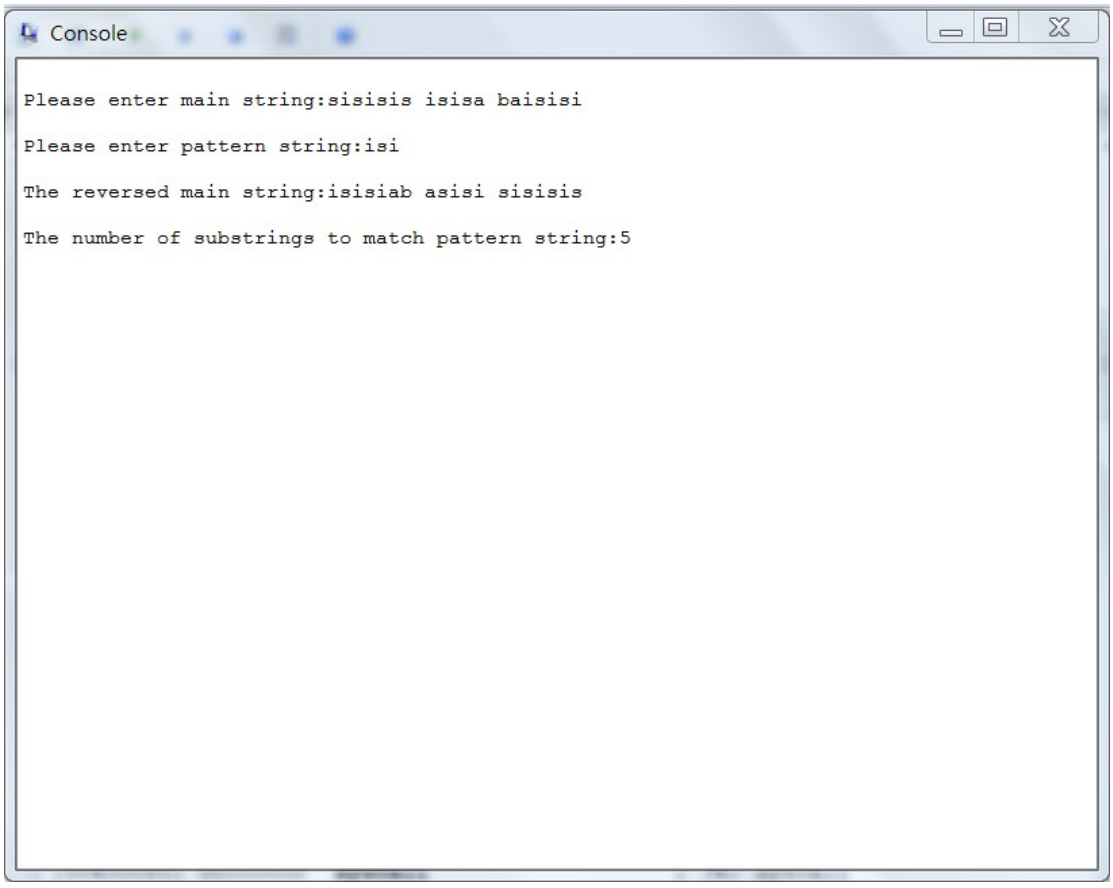
3. Tips: some utilities for MIPS string operations

```
.data
    str: .space 32 #offer a 32-byte space to store string
.text
.global main
main:
String_Input:
    li $v0,8 #invoke read string syscall. Pls refer to the uploaded slides on E3
    la $a0,str #set the starting address of space
    la $a1,5 #limit the string length of pattern string as 4
    syscall

Get_Partial_Sstring:
    la $t0,str1 #string start address
    lb $t1,1($t0) #t1=str1[1]
```

```
Compare_Char:
    la $t0,str1 #string start address
    lb $t1,0($t0) #t1=str1[0]
    lb $t2,1($t0) #t2=str1[1]
    beq $t1,$t2,same #if(str1[0]==str1[1]) go to same
Swap_Char:
    la $t0,str1 #string start address
    lb $t1,0($t0) #t1=str1[0]
    lb $t2,1($t0) #t1=str1[1]
    sb $t1,1($t0) #str[1]=str[0]
    sb $t2,0($t0) #str[0]=str1[1]
```

4. input: two strings. The first is the main string and the second is the pattern string.
5. output: <1> the reversed main string
   <2> the number of substrings in the main string to match pattern string.
6. example:



7. Team constraint: One person/team
8. Score ranking: One set of two strings will be offered on E3 and TA will prepare for two sets of strings for private testing.
    i. Incomplete program – at most 50 pts. (explain the program structure and the parts that have been completed in a word file)
    ii. Complete program but fail in all test cases – 65 (explain the program structure and potential bugs in your program based on the test result of public test cases)
    iii. Pass the test of one case – 85, pass the test of two cases – 91, pass the test of three cases – 98
9. Due date: 3/31 – on-line submission on E3.