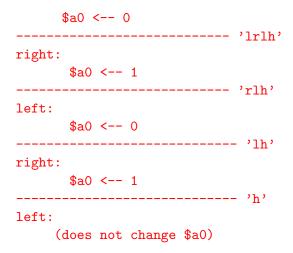
Question 1 (25 points): The program in the next page was written in an attempt to simulate a Turing machine. The program violates the MIPS calling conventions, by design, for increased efficiency: register \$s0 is used to store a global variable and thus is not be saved/restored according to the conventions. Please answer the following questions:

1. (5 points) If the line that contains the comment # print \$a0 were to be replaced with a system call to print the value of \$a0, which value would be printed?

The value printed would be one:



2. (5 points) There are two jump instructions that are commented as end of recursion. Only one of these two instructions is executed. What is the address of the instruction that will be executed?

The instruction at address 0x 8000 0030 will be executed, the other will not.

3. (15 points) Assume that when execution starts \$sp = 0x 1000 0000. In the figure below indicate the address to which \$sp will be pointing when the end-of-recursion instruction that you identified above is executed. Write the entire content of the stack when that instruction is executed.

|                         | Addr   | ess  | Memory Word  |
|-------------------------|--------|------|--------------|
| 0                       | x 1000 | 0014 |              |
| 0                       | x 1000 | 0010 |              |
| 0                       | x 1000 | 000C |              |
| 0                       | x 1000 | 8000 |              |
| 0                       | x 1000 | 0004 |              |
| 0                       | x 1000 | 0000 |              |
| 0                       | x OFFF | FFFC | 0x 8000 0018 |
| 0                       | x OFFF | FFF8 | 0x 8000 0078 |
| 0                       | x OFFF | FFF4 | 0x 8000 0040 |
| $\$$ SP $\rightarrow$ 0 | x OFFF | FFF0 | 0x 8000 0078 |
| 0                       | x OFFF | FFEC |              |

```
.data
0x4000 0000
             tape:
                     .string 'lrlh'
              .text
0x8000 0000
                   la
                             $s0, tape
                             $a1, '1'
0x8000 0004
                   li
0x8000 0008
                   ٦i
                             $a2, 'r'
                             $a3, 'h'
0x8000 000C
                   li
                             $a0, $zero, $zero
0x8000 0010
                   add
0x8000 0014
                   jal
                             right
              # print $a0
0x8000 0018
                   lw
                             $ra, 0($sp)
              left:
0x8000 001C
                   addi
                             $sp, $sp, -4
0x8000 0020
                             $ra, 0($sp)
                   SW
                             $t0, 0($s0)
0x8000 0024
                   lb
                             $s0, $s0, 1
0x8000 0028
                   addi
0x8000 002C
                   bne
                             $t0, $a3, test_direction_1
                                                             # if $t0 != 'h' do not halt
0x8000 0030
                             left_return
                                                             # halt (end of recursion)
                   j
              test_direction_l:
                             $a0, $a0, -1
0x8000 0034
                   addi
0x8000 0038
                             $t0, $a1, goto_left
                                                             # if $t0 == 'l' then go to left
                   beq
0x8000 003C
                             right
                                                             # else go right
                   jal
0x8000 0040
                             left_return
                   j
              goto_left:
0x8000 0044
                             left
                   jal
              left_return:
0x8000 0048
                   lw
                             $ra, 0($sp)
0x8000 004C
                   addi
                             $sp, $sp, 4
0x8000 0050
                   jr
                             $ra
              right:
                             $sp, $sp, -4
0x8000 0054
                   addi
0x8000 0058
                             $ra, 0($sp)
                   sw
0x8000 005C
                             $t0, 0($s0)
                   1b
0x8000 0060
                   addi
                             $s0, $s0, 1
0x8000 0064
                   bne
                             $t0, $a3, test_direction_r
                                                             # if $t0 != 'h' do not halt
0x8000 0068
                                                             # halt (end of recursion)
                   j
                             right_return
              test_direction_r:
                             $a0, $a0, 1
0x8000 006C
                   addi
0x8000 0070
                             $t0, $a2, goto_right
                                                             # if $t0 == 'r' then go to right
                   beq
0x8000 0074
                   jal
                             left
                                                             # else go left
0x8000 0078
                   j
                             right_return
              goto_right:
0x8000 007C
                   jal
                             right
              right_return:
                             $ra, 0($sp)
0x8000 0080
                   lw
0x8000 0084
                   addi
                             $sp, $sp, 4
0x8000 0088
                   jr
                             $ra
```