```
# Program to calculate number moves necessary to complete Towers of Hanoi with n discs.
# # discs "n" is specified by user input (integer).
# prompt user input, call recursive hanoi function, print out resulting value

# Written by Kollen G


       .data
       .align  2
prompt: .asciiz "\nEnter number of discs: "
result: .asciiz "\nIt will require "
result2:.asciiz " moves to complete Towers of Hanoi."

#--------------------------------
        .text
        .globl  main

main:
    move    $s0, $0     # s0 : computed # moves

    la  $a0, prompt #load prompt string
    li  $v0, 4      #code to print string
    syscall         #print

    li  $v0, 5      #take int input "N"
    syscall
    move    $s1, $v0    # s1 = user input int "N"

    move    $a0, $s1
    jal hanoi
    move    $s0, $v0    # s0 = result from hanoi
    j   print

#------ Display results and exit ---------------------------------

print:
    la  $a0, result #load display string
    li  $v0, 4      #code to print string
    syscall         #print

    li  $v0, 1      #code to print int
    move    $a0, $s0    #load computed # moves
    syscall         #print

    la  $a0, result2    #load display string
    li  $v0, 4      #code to print string
    syscall         #print

#----------------- Exit --------------------
    li  $v0, 10
    syscall
```

```
#******************************************************************
    # hanoi function
    #
    # a0 - user input "n"
    #
        #
        # v0 - computed # moves required
hanoi:
#--------------- Usual stuff at function beginning  ---------------
        addi    $sp, $sp, -24
        sw  $ra, 20($sp)
        sw  $s0, 16($sp)
        sw  $s1, 12($sp)
        sw  $s2, 8($sp)
        sw  $s3, 4($sp)
        sw  $s4, 0($sp)
#-------------------------- function body -------------------------
    move    $s0, $a0        # s0 : set to n
        move    $s2, $0     # s2 : computed # moves

        # base case if n <= 0
        bgt     $s0, 0, cont    # if (n<=0)
        addi    $s2, $0, 0  # s2 = 0
        # base case else if n = 1
cont:   bne $s0, 1, cont2   # if (n==1)
        addi    $s2, $0, 1  # s2 = 1

        #recursive call
cont2:  ble $s0, 1, done    # if (n>1)
        addi    $a0, $s0, -1    # a0 = (n-1)
        jal hanoi       # compute
        sll $s1, $v0, 1 # s1 = 2 * hanoi(n-1)

        addi    $s2, $s1, 1 # s2 = 2*hanoi(n-1) + 1
done:   move    $v0, $s2

#-------------------- Usual stuff at function end -----------------
        lw      $ra, 20($sp)
        lw  $s0, 16($sp)
        lw  $s1, 12($sp)
        lw  $s2, 8($sp)
        lw  $s3, 4($sp)
        lw  $s4, 0($sp)
        addi    $sp, $sp, 24
        jr      $ra
```