```
# CS 2734
# loop.s: Simple loop example
# Print the numbers from 0 to 19
# Register mapping:
#   $t0    loop index = 0, initially
#   $t1    max value = 20
#
        .globl main
main:                           # main has to be a global label
        addu    $s7, $0, $ra    # save return address in global register
## Start of simple loop example
        addi    $t0, $zero, 0   # t0 = 0
        addi    $t1, $zero, 20  # t1 = 20

loop:   li      $v0, 1
        add     $a0, $0, $t0
        syscall

        li      $v0, 4          # print_str (system call 4)
        la      $a0, blank      # takes the address of string as an argument
        syscall

        addi    $t0, $t0, 1     # t0 = t0 + 1
        bne     $t0, $t1, loop  # if (t0 != t1) goto loop

        li      $v0, 4
        la      $a0, finish
        syscall
## End of simple loop example
        addu    $ra, $0, $s7    # restore the return address
        jr      $ra             # return to the main program

        .data
blank:  .asciiz " "     # string to print
finish: .asciiz "\nTh-th-th-that’s all folks\n"

### output
# four06% spim -file loop.s
# SPIM Version 6.0 of July 21, 1997
# Copyright 1990-1997 by James R. Larus (larus@cs.wisc.edu).
# All Rights Reserved.
# See the file README for a full copyright notice.
# Loaded: /usr/local/lib/trap.handler
# 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
# Th-th-th-that’s all folks
# four06%
```