Assignment #4 Key

```
# Stephen P. Leach -- 10/03/07
# Polish.asm - functions that produce the fist 16 terms of the
#       Polish sequence starting with a given positive integer,
#       along with a simple driver program to test the code.


# Procedure terms --- written by Stephen P. Leach -- 10/03/07
#       Computes and prints the first 16 numbers in the Polish
#       sequence for a given starting number.  The actual
#       generation of a term is done through calling another
#       function, Polish.
# Register use:
#       $a0     integer parameter from calling routine, for
#               function Polish and for syscall
#       $v0     integer parameter for syscall and return
#               value from function Polish
#       $s0     saved value of current term
#       $s1     number of terms produced

terms:  addi    $sp, $sp, -12  # save $ra, $s0 and $s1 on stack
        sw      $ra, 8($sp)
        sw      $s1, 4($sp)
        sw      $s0, 0($sp)

        move    $s0, $a0       # save current term in $s0
        li      $s1, 16        # number of terms to be produced

        la      $a0, text      # print beginning text for sequence
        li      $v0, 4
        syscall

loop1:  move    $a0, $s0       # print current term
        li      $v0, 1
        syscall

        addi    $s1, $s1, -1   # decrement term counter
        beq     $s1, $0, exit  # and exit, if done

        la      $a0, blank     # otherwise, print blank after term,
        li      $v0, 4
        syscall

        move    $a0, $s0       # compute next term
        jal     Polish         # by calling the Polish function

        move    $s0, $v0       # return value becomes current term

        j       loop1          # and continue loop

exit:   la      $a0, cr        # go to next line of output

        lw      $s0, 0($sp)    # and restore values from stack
        lw      $s1, 4($sp)
        lw      $ra, 8($sp)
        addi    $sp, $sp, 12

        jr      $ra            # return to calling routine


# function Polish -- written by Stephen P. Leach -- 10/03/07
#       Given the current term (in $a0), this function computes
#       and returns (in $v0) the next term in the Polish
#       sequence (sum of the squares of the digits).
# Register use:
#       $a0     value of the current term
#       $v0     return value (next term)
#       $t0     the value 10
```

```
#       $t1     temporary calculations

Polish: move    $v0, $zero      # initialize next term to 0
        li      $t0, 10         # and $t0 to 10

loop2:  rem     $t1, $a0, $t0   # $t1 is units digit of current term
        mul     $t1, $t1, $t1   # square that value
        add     $v0, $v0, $t1   # add this value to next term

        div     $a0, $a0, $t0   # discard units digit of current term

        bne     $a0, $0, loop2 # continue if non-zero digits remain

        jr      $ra             # return to calling routine


# Driver program provided by Stephen P. Leach -- written 10/03/07

main:   la      $a0, intro      # print intro
        li      $v0, 4
        syscall

loop:   la      $a0, req        # request value of n
        li      $v0, 4
        syscall

        li      $v0, 5          # read value of n
        syscall

        ble     $v0, $0, out    # if n is not positive, exit

        move    $a0, $v0        # set parameter for terms procedure

        jal     terms           # call terms procedure

        j       loop            # branch back for next value of n

out:    la      $a0, adios      # display closing
        li      $v0, 4
        syscall

        li      $v0, 10         # exit from the program
        syscall


        .data
intro:  .asciiz "Welcome to the Polish sequence tester!"
req:    .asciiz "\nEnter an integer (zero or negative to exit): "
adios:  .asciiz "Come back soon!\n"

text:   .asciiz "First 16 terms: "
blank:  .asciiz " "
cr:     .asciiz "\n"
```