```
/*--- recursion.asm ---*/
/*

# Starter code AND demonstration of how to call a function as part of
# a complete SPIM program.

        .data
str_Welcome:  .asciiz   "Welcome to MIDN Hanling's Password Generator!\n "
str_GetSeed:  .asciiz   "Enter seed value (in range 0..10): "
str_GetPIN:   .asciiz   "Enter PIN value (in range 0..10): "
str_Out:      .asciiz   "Your password: "
str_Error:    .asciiz   "Seed or pin error!\n"
str_CR:       .asciiz   "\n"

        .text
        .globl main
main:
        # Print the Welcome prompt
        la      $a0, str_Welcome        # 'load address' of string to print
        li      $v0, 4                  # syscall #4 = print string
        syscall

        # Print prompt for seed value
        la      $a0, str_GetSeed        # 'lad address' of string to print
        li      $v0, 4                  # syscall #4 = print string
        syscall
        # Read seed from the user
        li      $v0, 5                  # syscall #5 = read integer
        syscall                         # read int, result in $v0
        # Check for valid seed
        slti    $t0, $v0, 11            # if seed < 11, t0 = 1
        beq     $t0, $zero, Error       # if t0 = 0, Exit
        # Save
        move    $s0, $v0                # s0 = seed

        # Print prompt for PIN
        la      $a0, str_GetPIN         # 'lad address' of string to print
        li      $v0, 4                  # syscall #4 = print string
        syscall
        # Read seed from the user
        li      $v0, 5                  # syscall #5 = read integer
        syscall                         # read int, result in $v0
        # Check for valid seed
        slti    $t0, $v0, 11            # if PIN < 11, t0 = 1
        beq     $t0, $zero, Error       # if t0 = 0, Exit
        # Save
        add     $s1, $v0, $zero         # s1 = PIN

        # Call a function to add that number together with 13
        add $a0, $zero, $s0             # Set up first argument (seed)
        add $a1, $zero, $s1             # Set up second argument (PIN)
        jal turkey                      # do add, result now in $v0
        add $s2, $v0, $zero             # s2 = answer

        # Print string announcing the result
        la      $a0, str_Out            # 'load address' of string to print
        li      $v0, 4                  # syscall #4 = print string
        syscall

        # Print actual result
        add     $a0, $s2, $zero         # want to print answer
        li      $v0, 1                  # syscall #1 = print integer
        syscall

        # terminate the program
        j Exit

Error:
        la      $a0, str_Error          # load error message
        li      $v0, 4                  # syscall #4 = print string
        syscall
Exit:
        li $v0, 10
        syscall


# Define seahorse
seahorse:
                slti $t0, $a0, 5
                bne  $t0, $zero, elseifS  #if dd <= 4, branch

                addi $sp, $sp, -8         #add 2 words to sp
                sw $a0, 4($sp)            #dd: 4 off sp
                sw $ra, 0($sp)            #ra: 0 off sp

                addi $a0, $a0, -1         #dd-1 ready for seahorse
                jal seahorse              #call seahorse
                mul $v0, $v0, 2           #double the result
                lw $t0, 4($sp)            #t0 = dd
                add $v0, $v0, $t0         #v0 has the answer (2*result+dd)

                lw $ra, 0($sp)            #get the ra back
                addi $sp, $sp, 8          #fix sp
                jr $ra                    #return for if

        elseifS:
                slti $t0, $a0, 3
                bne  $t0, $zero, elseS    #if dd <= 2, branch

                addi $sp, $sp, -8         #add 2 words to sp
                sw $a0, 4($sp)            #dd: 4 off sp
                sw $ra, 0($sp)            #ra: 0 off sp

                addi $a0, $a0, -2         #dd-2 ready for seahorse
                jal seahorse              #call seahorse
                lw $t0, 4($sp)            #t0 = dd
                mul $t0, $t0, 3           #to = 3*dd
                add $v0, $v0, $t0         #v0 has the answer (result+3*dd)

                lw $ra, 0($sp)            #get the ra back
                addi $sp, $sp, 8          #fix sp
                jr $ra                    #return for else if

        elseS:
                mul $v0, $a0, 3           #v0 = 3*dd
                addi $v0, $v0, 8          #v0 has the answer (8+dd*3
                jr $ra                    #return for else

# Define turkey
turkey:
                slti $t0, $a0, 3
                beq  $t0, $zero, elseifT  #if dd <= 4, branch

                addi $sp, $sp, -4         #add 1 word to sp
                sw $ra, 0($sp)            #ra 0 off sp

                add $a0, $a1, $zero       #cc ready for seahorse
                jal seahorse              #call seahorse
                addi $v0, $v0, 5          #v0 has the answer (5+result)

                lw $ra, 0($sp)            #get the ra back
                addi $sp, $sp, 4          #fix sp
                jr $ra                    #return for if

        elseifT:
                slti $t0, $a0, 5
                beq  $t0, $zero, elseT    #if dd <= 4, branch
```

```
                addi $sp, $sp, -8         #add 2 words to sp
                sw $a0, 4($sp)            #dd: 4 off sp
                sw $ra, 0($sp)            #ra: 0 off sp

                addi $a0, $a0, -1         #dd-1 ready for turkey
                addi $a1, $a1, 3          #cc+3 ready for turkey
                jal turkey                #call turkey
                lw $t0, 4($sp)            #t0 = dd
                add $v0, $v0, $t0         #v0 has the answer (dd+result)

                lw $ra, 0($sp)            #get the ra back
                addi $sp, $sp, 8          #fix sp
                jr $ra                    #return for else if

        elseT:
                addi $sp, $sp, -16        #add 4 words to sp
                sw $a0, 8($sp)            #dd: 8 off sp
                sw $a1, 4($sp)            #cc: 4 off sp
                sw $ra, 0($sp)            #ra: 0 off sp

                addi $a0, $a0, -1         #dd-1 ready for turkey (cc is fine)
                jal turkey                #call turkey
                sw $v0, 12($sp)           #result1 12 off sp

                lw $t0, 8($sp)            #t0 = dd
                lw $t1, 4($sp)            #t1 = cc
                add $a0, $t0, -2          #dd-2 ready for turkey
                add $a1, $t1, -1          #cc-1 ready for turkey
                jal turkey                #call turkey

                mul $v0, $v0, 2           #v0 = 2*result2
                lw $t0, 12($sp)           #t0 = result1
                add $v0, $v0, $t0         #v0 = result1 + 2*result2
                addi $v0, $v0, 3          #v0 = 3 + result1 + 2*result2 (ans)

                lw $ra, 0($sp)            #get the ra back
                addi $sp, $sp, 16         #fix sp
                jr $ra                    #return for else if
\n*/
```