```
1:  # Program to display nth term of fibonacci sequence, based on recursive calculation
2:  # term "n" is specified by user input (integer).
3:  # prompt user input, call fibonacci function, print out resulting value
4:
5:  # Written by Kollen G
6:
7:
8:         .data
9:         .align  2
10: prompt: .asciiz "\nEnter n'th value of Fib sequence to be computed: "
11: result: .asciiz "\nThe n'th value of Fib sequence is: "
12:
13: #-------------------------------
14:         .text
15:         .globl  main
16:
17: main:
18:     move    $s0, $0     # s0 : computed Fibonacci value
19:
20:     la  $a0, prompt #load prompt string
21:     li  $v0, 4      #code to print string
22:     syscall         #print
23:
24:     li  $v0, 5      #take int input
25:     syscall
26:     move    $s1, $v0    # s1 = user input
27:
28:     move    $a0, $s1
29:     jal fib
30:     move    $s0, $v0    # s0 = result from fib
31:     j   print
32:
33: #------ Display results and exit ---------------------------------
34:
35: print:
36:     la  $a0, result #load display string
37:     li  $v0, 4      #code to print string
38:     syscall         #print
39:
40:     li  $v0, 1      #code to print int
41:     move    $a0, $s0    #load computed fibonacci
42:     syscall         #print
43:
44: #----------------- Exit ---------------------
45:     li  $v0, 10
46:     syscall
47:
48:
49:
50: #*****************************************************************
51:     # fib function
52:     #
53:     # a0 - user input "n"
54:     #
55:         #
```

```
56:         # v0 - computed fib
57: fib:
58: #--------------- Usual stuff at function beginning  ---------------
59:         addi    $sp, $sp, -24
60:         sw  $ra, 20($sp)
61:         sw  $s0, 16($sp)
62:         sw  $s1, 12($sp)
63:         sw  $s2, 8($sp)
64:         sw  $s3, 4($sp)
65:         sw  $s4, 0($sp)
66: #-------------------------- function body --------------------------
67:     move    $s0, $a0        # s0: set to n
68:         li      $t1, 1          # t1: set to 1 for base case test
69:
70:         # base cases
71:         ble     $s0, 0, done    # if (n==0)
72:         ble     $s0, 1, done    # if (n==1)
73:
74:         #recursive calls
75:         addi    $a0, $s0, -1    # a0 = (n-1)
76:         jal fib     # compute
77:         move    $s1, $v0    # s1 = fib(n-1)
78:
79:         addi    $a0, $s0, -1    # a0 = (n-2)
80:         jal fib     # compute
81:         move    $s2, $a0    # s2 = fib(n-2)
82:         sll $s2, $s2, 2 # s2 = 4 * fib(n-2)
83:
84:         add $t1, $s1, $s2   # t1 = fib(n-1) + 4*fib(n-2)
85: done:   move    $v0, $t1
86:
87: #-------------------- Usual stuff at function end -----------------
88:         lw      $ra, 20($sp)
89:         lw  $s0, 16($sp)
90:         lw  $s1, 12($sp)
91:         lw  $s2, 8($sp)
92:         lw  $s3, 4($sp)
93:         lw  $s4, 0($sp)
94:         addi    $sp, $sp, 24
95:         jr      $ra
96:
97:
```