// Init

1. LDI #1 // R7 = 1
2. MOV R5 R7 // Now R5 = 1
3. LDI #2
4. MOV R6 R7 // Now R6 = 2
5. MOV R0 R7 // R0 = n
6. LDI H
7. MOV R1 R7 // R1 = H

// Start of function call

1. A: MOV R3 R0 // R3 = n
2. SUB R3 R6 // R3 = n – 2. If n <= 1, then R3 now < 0
3. LDI G
4. JMI R7 R3 // If R3 < 0, Jump to G

// Prepare to do fib(n-2)

1. ADD R3 R5 // R3 is now n-1
2. LDI E
3. MOV R4 R7 // R4 = E
4. LDI A // R7 = A
5. PUSH 3 // Push down register window by 3
6. JMP R7 // Jump to A. Now the ‘new’ R0 = n-1, and ‘new’ R1 = E

// Prepare to do fib(n-1)

1. E: MOV R2 R3 // R2 = R3 = y
2. MOV R3 R0 // R3 = n
3. SUB R3 R6 // R3 = n-2
4. LDI F
5. MOV R4 R7 // R4 = F
6. LDI A
7. PUSH 3 // Push down register window by 3
8. JMP R7 // Jump to A

// Return y after doing y = y + fib(n-1)

1. F: ADD R2 R3 // do y = y + fib(n-1)
2. MOV R0 R2 // Now R0 contains return value
3. POP 3 // Move window up by 3
4. JMP R4

// Return y=1 if n <= 1

1. G: MOV R0 R5 // R0 = 1
2. POP 3 // Move window up by 3
3. JMP R4
4. H: STP // Final value is in R0