```
; Class: CSE 313 Machine Organization Lab
; Section: 02
; Instructor: Taline Georgiou
; Term: Fall 2018
; Name: Eugene Kim
; Lab #4: Fibonacci Numbers
; Description:  The program takes an integer value, n, stores it
;       at x3100, and determines if it is at least 2 in magnitude.
;       If it is, then the program stores a value of integer 1
;       at x3101. If it isn't, with a series of loops, it calculates
;       the Fibonacci number for n.
;               The program executes further by calculating the last
;       n, denoted by N, and corresponding Fibonacci number, F_N, before
;       running out of sufficient bits in memory causing the sign bit to
;       change due to a carry of 1 into its bit location. The program
;       concludes by storing N into x3102 and F_N into x3103.
;               To experiment, change the integer value in Data.asm.

        .ORIG   x3000
        LDI     R0, X                   ; Load input integer, n, into R0.
; Check if n is at least 2.
        ADD     R0, R0, #-2             ; Subtract 2 from input integer, n, because
                                        ; we skip the first 2 Fibonacci sequences.
        BRnz    NZ                      ; If the result is positive it means that
                                        ; the input integer, n, is greater than 2,
                                        ; meaning its Fibonacci number is determine
                                        ; by applying F_n = F_(n-1) + F_(n-2).
; R4 being used to decrement the number of iterations to be carried out in the proc
        AND     R4, R4, #0
        ADD     R4, R4, #-1
; Preparing registers R1, R2, and R3.
        AND     R1, R1, #0              ; Clearing R1 and
        ADD     R1, R1, #1              ; loading R1 with integer 1.
        AND     R2, R2, #0              ; Clearing R2 and
        ADD     R2, R2, #1              ; loading R2 with integer 1.
        AND     R3, R3, #0              ; Clearing R3 and
LOOP    ADD     R0, R0, #0              ; loading R0 with integer 0 and bringing it
                                        ; into scope to be evaluated by the proceec
        BRz     ZERO                    ; Branch used to check if input integer, n,
                                        ; If so, then the program will execute to l
                                        ; where it'll store the Fibonacci number fc
        ADD     R3, R1, R2              ; If not, then apply F_n = F_(n-1) + F_(n-2
        AND     R1, R1, #0              ; n - 2 times or until the loop has
        ADD     R1, R2, #0              ; iterated n - 2 times at which point n - 2
        AND     R2, R2, #0              ; where it breaks out the loop
        ADD     R2, R3, #0              ; and stores the corresponding Fibonacci nu
        ADD     R0, R0, R4
        BR      LOOP
ZERO    STI     R3, F_n
        BRzp    SKIP
NZ      AND     R1, R0, #0
        ADD     R1, R0, #1
        AND     R2, R2, #0
        ADD     R2, R2, #1
        STI     R2, F_n
SKIP    AND     R4, R4, #0
        ADD     R4, R4, #2
        AND     R1, R1, #0
        ADD     R1, R1, #1
        AND     R2, R2, #0
        ADD     R2, R2, #1
        AND     R3, R3, #0
```