A simple example

```
        .ORIG    x3000
MAIN    LD       R6,STADDR
        ADD      R5,R6,#-1
;; Call SLSQR with N
        LD       R0,N           ;; push the argument
        ADD      R6,R6,#-1
        STR      R0,R6,#0
        JSR      SLSQR
        LDR      R0,R6,#0       ;; pop the result
        ADD      R6,R6,#1
        ADD      R6,R6,#1       ;; pop and discard the argument
        ST       R0,NSQ
        HALT

;; SillySquare routine
SLSQR   ADD      R6,R6,#-1      ;; "push" space for return
        ADD      R6,R6,#-1      ;; push R7
        STR      R7,R6,#0
        ADD      R6,R6,#-1      ;; push R5
        STR      R5,R6,#0
        ADD      R5,R6,#-1      ;; update R5 for new frame

        ADD      R6,R6,#-2      ;; save space for two registers
        STR      R0,R5,#0
        STR      R1,R5,#-1

        AND      R0,R0,#0       ;; R0 <= 0
        LDR      R1,R5,#4       ;; Load argument
        BRz      RETSQ
        ADD      R0,R0,#1
RETSQ   STR      R0,R5,#3       ;; store return value

        LDR      R0,R5,#0       ;; restore saved registers
        LDR      R1,R5,#-1
        ADD      R6,R5,#1       ;; "pop" local variables

        LDR      R5,R6,#0       ;; pop R5
        ADD      R6,R6,#1
        LDR      R7,R6,#0       ;; pop R7
        ADD      R6,R6,#1
        RET

STADD   .FILL    x4000
N       .FILL    x7
NSQ     .BLKW    1
        .END
```

A non-simple example

```
          .ORIG    x3000
MAIN    LD      R6,STADDR
        ADD    R5,R6,#-1
;; Call SLSQR with N
        LD      R0,N            ;; push the argument
        ADD     R6,R6,#-1
        STR     R0,R6,#0
        JSR     SLSQR
        LDR     R0,R6,#0        ;; pop the result
        ADD     R6,R6,#1
        ADD     R6,R6,#1        ;; pop and discard the argument
        ST      R0,NSQ
        HALT

;; SillySquare routine
SLSQR   ADD     R6,R6,#-1       ;; "push" space for return
        ADD     R6,R6,#-1       ;; push R7
        STR     R7,R6,#0
        ADD     R6,R6,#-1       ;; push R5
        STR     R5,R6,#0
        ADD     R5,R6,#-1       ;; update R5 for new frame

        ADD     R6,R6,#-2       ;; save space for two registers
        STR     R0,R5,#0
        STR     R1,R5,#-1

        AND     R0,R0,#0        ;; R0 <= 0
        LDR     R1,R5,#4        ;; Load argument
        BRz     RETSQ

;; SillySqure(n) = SillySquare(n-1) + n + n - 1

          ADD     R0,R1,#-1     ;; Push N-1
          ADD     R6,R6,#-1
          STR     R0,R6,#0

          JSR     SLSQR         ;; Compute SillySquare(N-1)
          LDR     R0,R6,#0      ;; pop return value to R0
          ADD     R6,R6,#1      ;;
          ADD     R6,R6,#1      ;; pop and discard argument

          ADD     R0,R0,R1      ;; R0 <- (N-1)*(N-1) + N
          ADD     R0,R0,R1      ;; R0 <- (N-1)*(N-1) + 2*N == N*N + 1
          ADD     R0,R0,#-1     ;; R0 <- N*N

RETSQ   STR     R0,R5,#3        ;; store return value

        LDR     R0,R5,#0        ;; restore saved registers
        LDR     R1,R5,#-1
        ADD     R6,R5,#1        ;; "pop" local variables

        LDR     R5,R6,#0        ;; pop R5
        ADD     R6,R6,#1
        LDR     R7,R6,#0        ;; pop R7
        ADD     R6,R6,#1
        RET

STADD   .FILL   x4000
N       .FILL   x7
NSQ       .BLKW   1
          .END
```