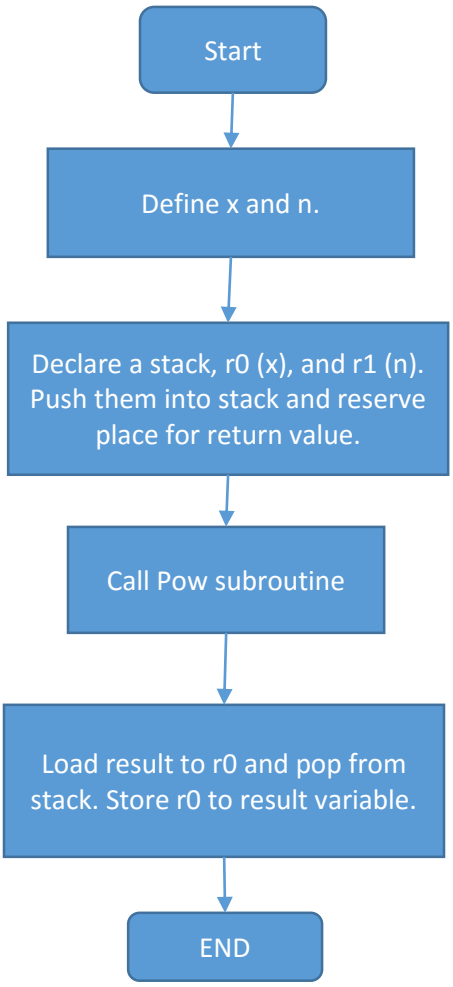
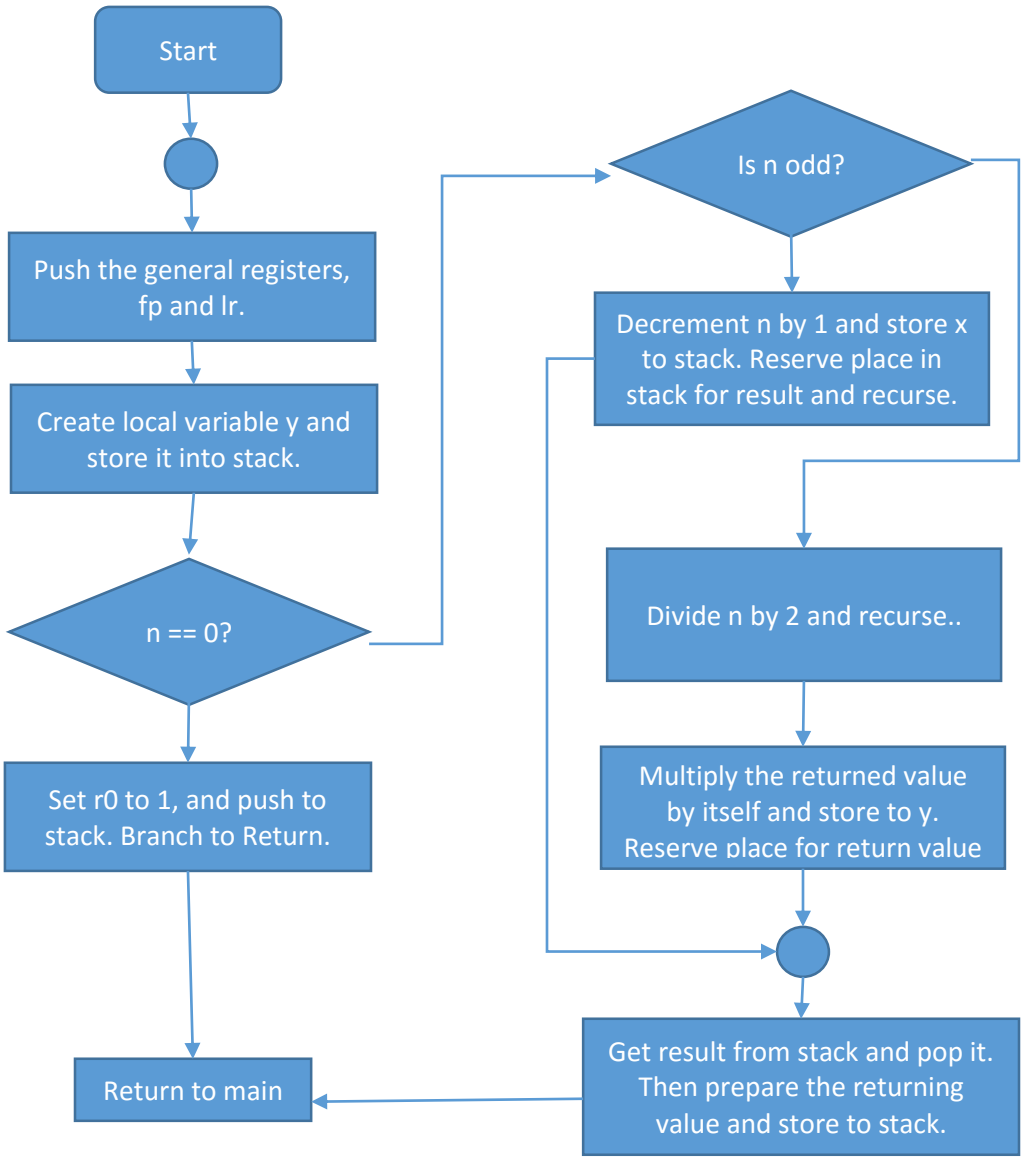
Assignment 5 Flow Chart

Main



Pow Subroutine



For amount of stack frames…

N = 0: 8 stack frames

N = 1: 9 stack frames

N = 2: 10 stack frames

N = 3: 11 stack frames

N = 4: 11 stack frames

N = 5: 12 stack frames

N = 6: 12 stack frames

N = 7: 13 stack frames

N = 8: 13 stack frames

N = 9: 14 stack frames

N = 10: 14 stack frames

N = 11: 15 stack frames

N = 12: 15 stack frames

Assignment 5 Code

```
;-------------------------------------------------------------------------------------------------------------------------

                AREA power, CODE, READONLY

x               EQU     5               ;x value for x^n
n               EQU     3               ;n value for x^n

                ENTRY

Main            ADR     sp, stack       ;define the stack

                MOV     r0, #x          ;prepare the parameter x
                MOV     r1, #n          ;prepare the parameter n
                STR     r1, [sp, #-4]!  ;push n on the stack
                `STR    r0, [sp, #-4]!  ;push x on the stack

                SUB     sp, sp, #4      ;reserve a place in the stack for return value

                BL      Pow             ;call the Pow subroutine

                LDR     r0, [sp], #4    ;load the result in r0 and pop it from the stack
                ADD     sp, sp, #4      ;remove the parameter from the stack

                ADR     r1, result      ;get the address of result
                STR     r0, [r1]        ;store the final result of calculation into result
                                        ;variable

Loop            B       Loop            ;infinite loop
```

```
;-------------------------------------------------------------------------------------------------------------------------

                AREA power, CODE, READONLY
Pow             STMFD       sp!,{r0, r1,r2 , fp, lr}    ;push general registers, as well as fp and lr
                MOV         fp,sp                       ;set fp for this call
                SUB         sp, sp, #4                  ;create space for y local variable

                LDR         r0, [fp, #0x18]             ;get parameter from stack

                STR         r0, [fp,#-0x4]              ;update the value of local y variable

                CMP         r1, #0                      ;IF n == 0, THEN
                MOVEQ       r0, #1                      ;       prepare the value to be returned
                STREQ       r0, [fp, #0x14]             ;       store the returned value in stack
                BEQ         Return                      ;       branch to return section

                TST         r1, #1                      ;Performs test to see if n is odd or not
                                                        ;IF n is odd, THEN
                SUBNE       r1, #1                      ;       decrement n by 1
                STRNE       r0, [sp, #-4]!              ;       store x in to stack
                SUBNE       sp, sp, #4                  ;       reserve a place in stack for return
                                                        ;       value

                BLNE        Pow                         ;       call Pow Function

                                                        ;ELSE IF n is even, THEN
                LSREQ       r1, #1                      ;       divide n by 2

                BLEQ        Pow                         ;       call Pow Function
```

```
                MULEQ       r2, r0, r0          ;       multiply returned value by itself and
                                                ;       store to y
                STREQ       r2,[sp,#-4]!        ;       store y to stack

                SUBEQ       sp, sp, #4          ;       reserve place in stack for return value

                LDR         r0, [sp], #4        ;load the result in r0 and pop it from the stack
                ADD         sp, sp, #4          ;remove the parameter from the stack

                MUL         r2, r0, r2          ;prepare the value to be returned
                STR         r2, [fp, #0x14]     ;store the returned value in stack

Return          MOV         sp, fp              ;collapse all working space for this function call
                LDMFD       sp!,{r0,r1, r2, fp, pc}  ;load all registers and return to caller

;-------------------------------------------------------------------------------------------------------------------------

                AREA power, DATA, READWRITE
result          DCD         0x00                ;the final result
                SPACE   0xB4                    ;declare the space for stack
stack           DCD         0x00                ;initial stack position (FD model)

;-------------------------------------------------------------------------------------------------------------------------

                END
```