CS2208 Assignment 5
Pengwei Zhang
April 2, 2018

Power.s

```
                AREA power, CODE, READONLY
x               EQU 2
n               EQU 3
                ENTRY

MAIN    ADR sp,stack                        ;define stack

                MOV r0,#n                   ;prepare n parameter
                STR r0,[sp,#-4]!        ;push r0 onto stack
                MOV r1,#x                   ;prepare x parameter
                STR r1,[sp,#-4]!        ;push r1 onto stack
                SUB sp,sp,#4                ;reserve space on stack for return value

                BL Power                    ;call power subroutine

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

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

DONE    B DONE                              ;infinite loop

;-------------------------------------------------------
                AREA power, CODE, READONLY
Power   STMFD sp!,{r0,r1,r2,fp,lr}          ;push registers onto stack
                MOV fp,sp                   ;set the fp register for the call of subroutine Power

                LDR r0,[fp,#0x1c]           ;get parameter n from stack
                LDR r2,[fp,#0x18]           ;get parameter x from stack

                CMP r0,#0x00                ;compare if n==0
                MOVEQ r0,#1                 ;prepare value 1 to be returned
                STREQ r0,[fp,#0x14]         ;store the returned value in stack
                BEQ return                  ;branch to return
```

```
                ANDS r1,r0,#1                   ;if n is odd
                BEQ EVEN                        ;branch to even
ODD             SUB r1,r0,#1                    ;prepare new parameter value
                STR r1,[sp,#-4]!        ;push r1 onto stack
                STR r2,[sp,#-4]!        ;push r2 onto stack
                SUB sp,sp,#4                    ;reserve place in stack for return value
                BL Power                        ;call subrountine power with new parameter
                LDR r1,[sp],#4                  ;pop result from stack and load into r1
                ADD sp,sp,#4                    ;remove parameter from stack
                MUL r2,r1,r2                    ;prepare value to be returned
                STR r2,[fp,#0x14]               ;store returned value in stack
                B return                        ;branch to the return section

EVEN    LSR r1,r0,#1                            ;prepare new parameter value
                STR r1,[sp,#-4]!        ;push r1 onto stack
                STR r2,[sp,#-4]!        ;push r2 onto stack
                SUB sp,sp,#4                    ;reserve a place in stack for return value
                BL Power                        ;call subrountine with new parameter
                LDR r1,[sp],#4                  ;pop result from stack and load into r1
                ADD sp,sp,#4                    ;remove paramter from the stack
                LSL r2,r1,#1                    ;prepare value to be returned
                STR r2,[fp,#0x14]               ;store returned value in stack

return  MOV sp,fp                               ;merge all working spaces for this function
                LDMFD sp!,{r0,r1,r2,fp,pc}      ;load all register and return to caller

;-----------------------------------------
                AREA prog2, DATA, READWRITE
result  DCD 0x00                                ;final result
                SPACE 0xB4                      ;space for stack
stack   DCD 0x00                                ;initial stack address
                END
```

Sketch of stack structure