```
        AREA Assignment_5, CODE, READONLY
        ENTRY
                                                ; Assignment 5
                                                ; Program to do x^n
                                                ; D Truong


        BL      power                           ; call power function

power   STMFD  sp!, {r0, lr}
        ; (push) Full decending stack, 6-11 Load int x and   ; unsigned int n into stack memory
        ADD     r0, sp, #4
        SUB     sp, sp, #16
        STR     r1, [r0, #-16]
        STR     r2, [r0, #-20]
        STR     r4, [r0, #-20]

        LDR     r4, [r0, #-20]          ; 13-14 checking if n is 0
        CMP     r4, #0
        BNE     if                      ; branch to if (n&1)
        MOV     r2, #1                  ; 16-17 return 1 if it is 0
        b       ending

if      LDR     r4, [r0, #-20]          ; 19-22 check if (n&1)
        AND     r4, r4, #1
        CMP     r4, #0
        BEQ     return1                 ; jump to return1 if not

        LDR     r4, [r0, #-20]          ; 24-31 load values and return x*power(x, n-1)
```

```
                SUB             r4, r4, #1
                MOV             r1, r3
                LDR             r1, [r0, #-16]
                BL              power                   ; recursively call own function on value
                MOV             r3, r1
                LDR             r2, [r0, #-16]
                MUL             r2, r3, r4
                b               ending                  ; jump to ending branch

return1         LDR     r4, [r0, #-20]                  ; 33-42 Load values and return y=power(x,n>>1)
                LSR             r4, r4, #1
                MOV             r2, r4
                LDR             r1, [r0, #-16]
                BL              power                   ; recursively call own function on value
                MOV             r4, r1
                STR             r4, [r0, #-8]
                LDR             r4, [r0, #-8]
                LDR             r3, [r0, #-8]
                MUL             r2, r3, r4

ending          MOV             r4, r2          ; 44-47 ending off function and getting values off stack
                MOV             r1, r4
                SUB             sp, r0, #4
                LDMIA           sp!, {r0, lr}           ; (pop) incremented after
                BX              lr
                END
```