

COPY OF SOURCE CODE

```

                                AREA power, CODE, READONLY
x                               EQU 5
n                               EQU 2
                                ENTRY
main        ADR sp,stack        ;define stack
                                MOV r0, #x          ;prepare x
                                STR r0,[sp,#-4]!    ;push x to the stack
                                MOV r0, #n          ;prepare n
                                STR r0,[sp,#-4]!    ;push n to the stack

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

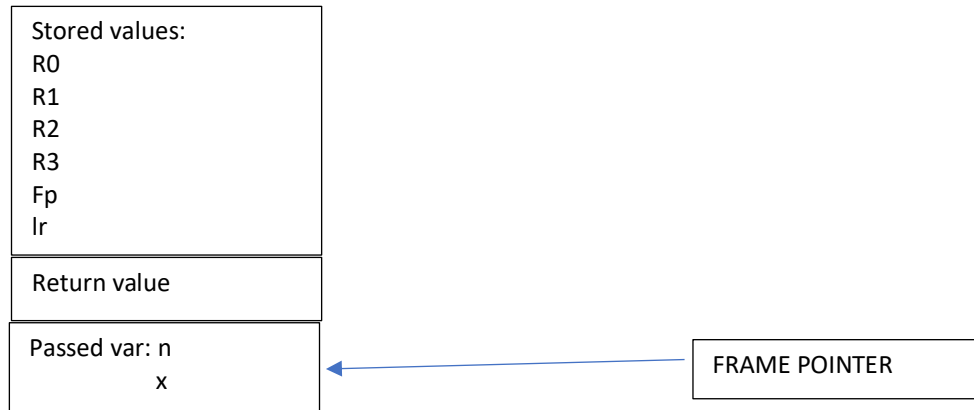
                                BL pow
                                LDR r0,[sp],#4      ;load result into r0
                                ADD sp,sp,#4        ;remove param from the stack
                                ADR r1, result      ;get address of result
                                STR r0,[r1]         ;store the result
loop        b loop

-----
                                AREA power, CODE, READONLY
pow
stack        STMFD sp!, {r0,r1,r2,r3,fp,lr}      ;push registers to
                                MOV fp,sp          ;set fp
                                LDR r0,[fp,#0x20]   ;r0 is x
                                LDR r1,[fp,#0x1C]   ;r1 is n
                                ; if (n==0){
                                CMP r1,#0
                                MOVEQ r1,#1         ;prepare value to
return        STREQ r1, [fp,#0x18]                ;push value to be
returned      BEQ return                          ;}
                                ; if (n & 1){
                                AND r2,r1,#1        ;bitwise and (to
find if n is odd) CMP r2,#1                       ;if 'true'
                                SUBEQ r1,#1         ;new n value (n-
1)
                                STREQ r0,[sp,#-4]!    ;store x on stack
                                STREQ r1,[sp,#-4]!    ;store n on stack
                                SUBEQ sp,sp,#4        ;reserve spot on
stack for return BLEQ pow                          ;call power
subroutine    LDREQ r2,[sp],#4                     ;load result into r2
```

	ADDEQ sp,sp,#4	;remove param
from stack		
	MULEQ r3,r2,r0	;x * power(x, n -
1)		
	STREQ r3, [fp,#0x18]	;push value to be
returned		
	BEQ return	;}
	;else	
	LSR r1,#1	;new n value
(n>>1)		
	STR r0,[sp,#-4]!	;store x on stack
	STR r1,[sp,#-4]!	;store n on stack
	SUB sp,sp,#4	;reserve spot on
stack for return		
	BL pow	;call power
subroutine		
	LDR r2,[sp],#4	;load result into
r2		
	ADD sp,sp,#4	;remove param
from stack		
	MUL r3,r2,r2	;y*y
	STR r3, [fp,#0x18]	;push value to be
returned		
	B return	;}
return	MOV sp,fp	;collapse
and return	LDMFD sp!,{r0,r1,r2,r3,fp,pc}	;load all registers
<hr/>		
	AREA power, DATA, READWRITE	
result	DCD 0x00	;final answer
	SPACE 0x255	;space for the stack
stack	DCD 0x00	;initial stack position (FD)
<hr/>		
	END	

STACK

Used FD stack structure where the stack frame was moved for every entry into the function.
There would be $n/2$ total stack frames.



To Run

To run, create a .ini with the line
MAP 0x20, 0x30C EXEC READ WRITE