Raymond Wong
250996400
rwong328
compsci 2208
assignment5

2 stack frames for n=0
3 stack frames for n=1
4 stack frames for n=2
5 stack frames for n=3
5 stack frames for n=4
6 stack frames for n=5
6 stack frames for n=6
7 stack frames for n=7
6 stack frames for n=8
7 stack frames for n=9
7 stack frames for n=10
8 stack frames for n=11
7 stack frames for n=12

Source code with comments

```
                AREA question1, CODE, READWRITE
                ;Raymond Wong, 250996400, rwong328, compsci 2208 assignment 5
                ENTRY
                mov sp, #0x1000                         ;set up stack pointer
                mov fp, #0x1000                         ;set up frame pointer
                B main                                          ;go to main function
                ;power function
power   sub sp, sp, #4                          ;create stack frame
                str fp, [sp]                            ;push frame pointer on the stack
                mov fp, sp                                      ;frame pointer point at the base
                str lr, [sp, #-4]!                      ;push the lr into the stack
                sub sp, sp, #4                          ;move 4 byte up for int y
                ldr r0, [fp, #8]                        ;get x from the stack frame
                str r0, [sp, #-4]!                      ;store x in the stack frame and update stack pointer
                ldr r0, [fp, #4]                        ;get n from the stack frame
                cmp r0, #0                                      ;check if n equal to 0
                BNE check                                       ; go to check if not equal
                str r0, [sp, #-4]!                      ;store n in the stack frame
                mov r0, #1                                      ;store 1 in r0
                str r0, [fp, #12]                       ;put the 1 in y or result variable of the last stack frame, which is the return value
                B return                                        ;go to return
check   and r1, r0, #0x00000001         ;check if the n value is odd
                cmp r1, #0                                      ;compare the value with 0
                BEQ even                                        ; if zero, then it is zero
                SUB r0, r0, #1                          ;get n-1
                STR r0, [sp, #-4]!                      ;store n-1 into the stack in the variable n position and update the pointer
                BL power                                        ;branch to power and save the return address
                ldr r1, [fp, #-8]                       ;get result from the stack frame
                ldr r2, [fp, #-12]                      ;get x from the stack frame
                mul r0, r1, r2                          ;multiple x by the return result
                str r0, [fp, #12]                       ;store multiplied result in y or result variable of the last stack frame, which is the return value
                B return
even    mov r0, r0, LSR #1                      ;divide n by 2
                str r0, [sp, #-4]!                      ;store n/2 into n variable position
                BL power                                        ;branch to power and save the return address
                ldr r0, [fp, #-8]                       ;get result from the stack frame
                mul r1, r0, r0                          ;mutiply the result, y*y
                str r1, [fp, #12]                       ;store y*y in the stack frame
return  ldr lr, [fp, #-4]                       ;get the past lr from the stack frame
                mov sp, fp                                      ;restore the stack pointer
                ldr fp, [sp]                            ;restore old the frame pointer from stack
                add sp,sp, #4                           ;move the stack pointer down 4 bytes
                mov pc, lr                                      ;rturn by loading LR into PC
                ;main function to pass by value via stack, and create stack frame for x and
main    sub sp, sp, #4                          ;move the stack pointer up
                str fp, [sp]                            ;push the frame pointer on the stack
                mov fp, sp                                      ;the frame pointer point at the base
                ;put parameter x, n and result variable in the stack frame
                mov r0, #0              ;set the result initially equal to 0
                str r0, [sp, #-4]!                      ;put the result variable in stack frame
                mov r0, #2                                      ;let int x=2
                str r0, [sp, #-4]!                      ;put x variable in stack frame
                mov r0, #5                                      ;let int n=3
                str r0, [sp, #-4]!                      ;put n in stack frame
                BL power
                ;clean the stack from the parameter
                mov sp, fp                                      ;restore stack pointer
                ldr fp,[sp]                                     ;retore old frame pointer from stack
                add sp, sp, #4                          ;move 4 byte down 4 bytes
Loop    B Loop                                          ;end the program
                END
```

2^5

↓

2 × power(2,4)

↓

y × y = power(2,2)

↓

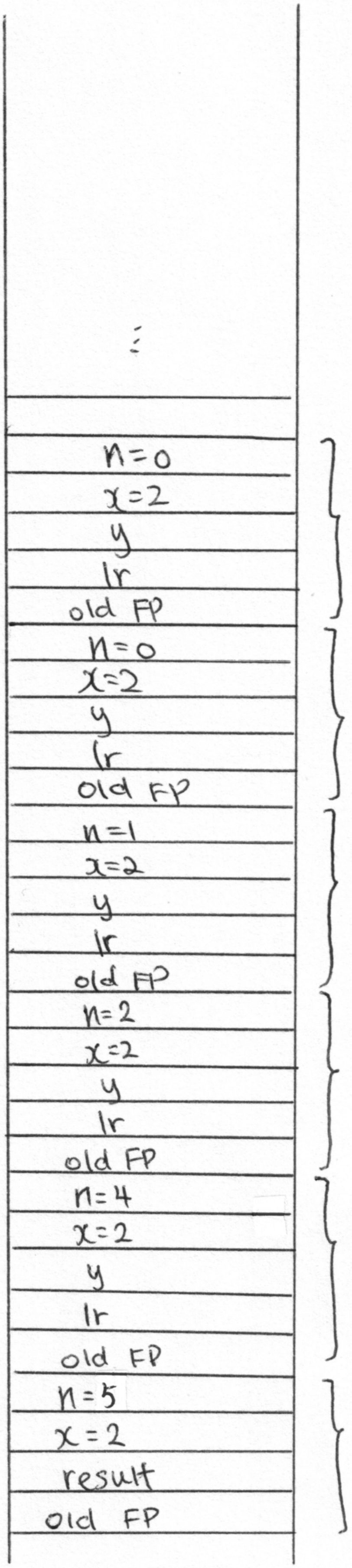
y×y = power(2,1)

↓

2 × power(2,0)

↓

1



stack frame for power(2,0) that store the result=1 to y from last stack frame

stack frame for power (2,1) that pass x=2, n=0 to next stack frame → power(2,0)

stack frame for power (2,2) that pass x=2, n=1 to next stack frame → power(2,1)

stack frame for power (2,4) that pass x=2, n=2 to next stack frame → power(2,2)

stack frame for power(2,5) that pass x=2, n=4 to next stack frame → power(2,4)

stack frame for main function