CS2208 Assignment 5

Jane Wang

250901299

```
        AREA    assign4, CODE, READWRITE

        ENTRY
        ADR     SP, STCK        ;load address of stack into stack pointer
        MOV     r0, #XVAR       ;loads value of x into r0
        MOV     r1, #NVAR       ;loads value of n into r1
        BL      BEGIN           ;starts function

FINAL   ADR     r1, result      ;loads address of result variable
        STR     r0, [r1]        ;stores final result into result variable

LOOP    B       LOOP            ;end of program

BEGIN   STMFD   SP!, {FP, LR}   ;store frame pointer and link register onto
                                ;stack
        ADD     FP, SP, #4      ;sets frame pointer
        STMFD   SP!, {r0, r1}   ;pushes r0 (x) and r1 (n) onto the stack

BASE    LDR     r1, [FP, #-8]   ;loads n into working register r1
        CMP     r1, #0          ;checks if n equals 0
        BNE     ODD             ;not base case - branch to check odd/even

        MOV     r1, #1          ;base case is reached - set r1 = 1
        B       RETURN          ;branch to RETURN since base case is reached

ODD     STR     r1, [FP, #-8]   ;store current n into working register r1
        TST     r1, #1          ;check if current n is odd
        BEQ     EVEN            ;current n is even - branch to EVEN
        SUB     r1, r1, #1      ;current n is even - subtract 1 from it
        BL      BEGIN           ;recursive call on current n

ODDREC  LDR     r1, [FP, #-8]   ;loads value from working register for
                                ;recursive multiplication
        CMP     r1, #1          ;compares value to 1
        BEQ     ONE             ;value is 1 - branch to ONE

        LDR     r1, [FP, #-12]  ;loads x from the stack
ONE     MUL     r0, r1, r0      ;multiplies x by the current n
        B       RETURN          ;branches to RETURN

EVEN    STR     r1, [FP, #-8]   ;store current n into working register r1
        MOV     r1, r1, LSR #1  ;n is even - divide current n by 2
        BL      BEGIN           ;recursive call on current n

EVENREC LDR     r1, [FP, #-8]   ;loads value from working register for
                                ;recursive multiplication
        MOV     r1, r0          ;moves the current value to r1
        MUL     r0, r1, r0      ;squares the value, store in r0
```

```
RETURN  SUB     SP, FP, #4      ;re-adjust FP before return
        LDMFD   SP!, {FP, PC}   ;restore FP and return

        AREA    assign4, DATA, READWRITE
XVAR    EQU     2               ;symbolic name for x constant
NVAR    EQU     4               ;symbolic name for n constant
        SPACE   0xFF            ;space for the stack
                ALIGN
STCK    DCD     0x00            ;beginning of the stack
                END
```

Stack Frame

- Full descending stack