Solutions to Tutorial 3

3.1 Stack Manipulation Operations

(1) `SP = 0xFFB`
(2) `PSH R1`
(3) `ADD SP,#4`
(4) `MOV R0,[SP+2]`.

3.2 Modular Programming – Subroutine Call and Parameter Passing

(1) `NumX` and `NumY` are passed by value. Values of memory variables are pushed to the stack. `Ans` is passed using reference. Address of the memory variable pushed onto the stack.

(2) After (b1) `PC = 0x004` `SP = 0xFFE`

   After (c1) `PC = 0x020` `SP = 0xFFB`

   After (s6) `PC = 0x00A` `SP = 0xFFC`

(3) `R0 = 0x00A`

(4) `ADD SP,#3`

(5) Replace `CALL MySub` with `CALL [PC+0x016]`.

(6) Suggested solutions:

```
MySub   PSHM  15            ; (s1) Save registers R0,R1,R2,R3
        MOV   R1,[SP+7]     ; (s2) Retrieve NumX from stack
        MOV   R2,[SP+6]     ; (s3) Retrieve NumY from stack
        MOVS  R3,#0         ; Complete the segment of code to compute the
Loop    ADD   R3,R1         ; value of NumX*NumY using successive addition
        DEC   R2            ; decrement NumY till reach zero
        JNE   Loop          ; loop back till NumX added NumY times
        MOV   R0,[SP+5]     ; (s4a) Move the result directly to …
        MOV   [R0],R3       ; (s4b) … the memory variable Ans
        POPM  15            ; (s5) Restore saved registers
        RET                 ; (s6) Return to calling program
```