Example of a MULT Subroutine

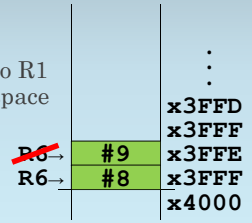
```
MULT
AND R2,R2,#0
ADD R1,R1,#0
BRz MULTDONE
MULTLOOP
ADD R2,R2,R0
ADD R1,R1,#-1
BRnp MULTLOOP
MULTDONE
ADD R0,R2,#0
RET
```

What is the call interface for this subroutine?

Inputs: R0, R1
Output: R0 ← R0 × R1
Caller-saved:
R1, R2, R7
Callee-saved:
R3, R4, R5, R6

To Multiply: Pop Twice, Multiply, Push Product

```
STACKMULT

LDR R1,R6,#0    ; pop 9 into R1
ADD R6,R6,#1    ; remove space
```

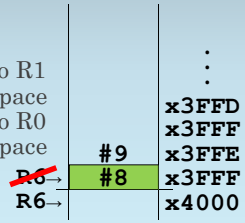


Is the "9" still in memory?

Probably, but it's NOT on the stack.

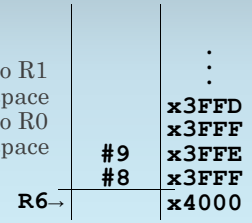
To Multiply: Pop Twice, Multiply, Push Product

```
STACKMULT

LDR R1,R6,#0    ; pop 9 into R1
ADD R6,R6,#1    ; remove space
LDR R0,R6,#0    ; pop 8 into R0
ADD R6,R6,#1    ; remove space
```



To Multiply: Pop Twice, Multiply, Push Product

```
STACKMULT

LDR R1,R6,#0    ; pop 9 into R1
ADD R6,R6,#1    ; remove space
LDR R0,R6,#0    ; pop 8 into R0
ADD R6,R6,#1    ; remove space
JSR MULT        ; R0 is 72
```



We're ready to call MULT!

Note that the stack is empty.

