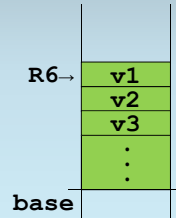## Let's Start by Saving R1 and Reading v1

```
SUM_OF_3
ST R1,SAVE_R1   ; save R1
LDR R0,R6,#0    ; R0 ← v1
```

So far, so good?

But we're not going to pop v1...
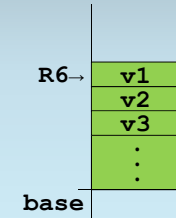
```
SAVE_R1 .BLKW #1
```

Make a space down here.

R6→ v1, v2, v3, ·, ·, base

Load v2 Using LDR from M[R6 + 1]

```
SUM_OF_3
ST R1,SAVE_R1   ; save R1
LDR R0,R6,#0    ; R0 ← v1
LDR R1,R6,#1    ; R1 ← v2
ADD R0,R0,R1    ; R0 ← v1 + v2
```

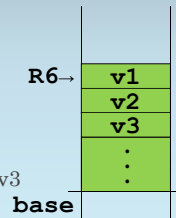Read v2 before popping v1.

And find the sum...

```
SAVE_R1 .BLKW #1
```

R6→ v1, v2, v3, ·, ·, base

Do the Same for v3 (with Offset 2)

```
SUM_OF_3
ST R1,SAVE_R1   ; save R1
LDR R0,R6,#0    ; R0 ← v1
LDR R1,R6,#1    ; R1 ← v2
ADD R0,R0,R1    ; R0 ← v1 + v2
LDR R1,R6,#2    ; R1 ← v3
ADD R0,R0,R1    ; R0 ← v1 + v2 + v3
```
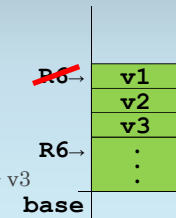
Now read v3.   And find the sum...

```
SAVE_R1 .BLKW #1
```

R6→ v1, v2, v3, ·, ·, base

Pop All Three Values at Once

```
SUM_OF_3
ST R1,SAVE_R1   ; save R1
LDR R0,R6,#0    ; R0 ← v1
LDR R1,R6,#1    ; R1 ← v2
ADD R0,R0,R1    ; R0 ← v1 + v2
LDR R1,R6,#2    ; R1 ← v3
ADD R0,R0,R1    ; R0 ← v1 + v2 + v3
ADD R6,R6,#3    ; pop all three
```

Done with the values: pop all three!

```
SAVE_R1 .BLKW #1
```

R̶6̶→ v1, v2, v3, R6→ ·, ·, base

8