

Recall: Function that finds $\sum_{i=1}^N i$:

```
int FINDSUM (int N) {  
    int sum = 0;  
    while (N != 0) {  
        sum += N;  
        -- N;  
    }  
    return (sum);  
}
```

Redo this code as a recursive function:

```
int FINDSUM (int N) {  
    if (N == 0) {  
        return (0); (*)  
    }  
    else  
        return (N + FINDSUM (N-1)); (*)  
}
```

Example: main program calls. FINDSUM(5)

main: FINDSUM(5) : (*) return (5 + FINDSUM(4));

FINDSUM(4) : (*) return (4 + FINDSUM(3));

FINDSUM(3) : (*) return (3 + FINDSUM(2));

FINDSUM(2) : (*) return (2 + FINDSUM(1));

FINDSUM(1) : (*) return (1 + FINDSUM(0));

FINDSUM(0) : (*) return (0);

.text

.global - start

```
- start:  mov    SP, #0x20000    // initial stack pointer.
          mov    R4, #N          // R4 → N
          LDR     R0, [R4]        // R0 = N
          BL      FINDSUM
          STR     R1, [R4], #4
END:      B       END
```

N: .word 11

Sum: .space 4

/* Recursive sum

N is in R0, result returned in R1 */

FINDSUM: movs R1, R0 // N == 0?

moveq PC, LR // if yes, return 0;

RECURSE: push {R0, LR} // save state
 (N, return addr.)

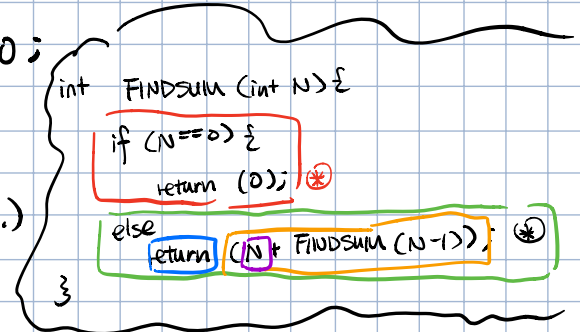
SUB R0, #1 // N-1

BL findsum // recurse

POP {R0, LR} // restore state

ADD R1, R0 // FINDSUM (N-1) + N

MOV PC, LR



e.g. N=3

R0	R1	LR
3	3	END
2	2	EEE
1	1	EEE
0	0	EEE
1	1	
2	3	
3	6	

6 - returned to main program

