

Question 1: Worksheet: Tracing Recursive Stack Frames in RISC-V `sqr:`

```
addi sp, sp, -8
sw   ra, 4(sp)
sw   a0, 0(sp)

slti t0, a0, 2
bne  t0, x0, base

addi a0, a0, -1
jal  ra, sqr

lw   a0, 0(sp)
add  a0, a0, a0
add  a0, a0, a0
addi a0, a0, -1
j    done
```

```
base:
    # return n
```

```
done:
    lw   ra, 4(sp)
    addi sp, sp, 8
    jr   ra
```

Assume the function is initially called with `a0 = 10`.

Part A: Understanding the Stack Frame

1.1. What quantities get stored in each stack frame? (Hint: look for the `sw` instructions)

1.2. How many bytes does each stack frame use?

1.3. Which register holds the return value?

Part B: Trace the First Few Calls

Call	Value of a0 at start
sqr(10)	10
sqr(____)	
sqr(____)	
sqr(____)	

Part C: Stack Snapshot

Draw the stack after **three recursive calls** (i.e., when computing `sqr(8)`).

Label:

- saved ra
- saved a0
- current sp

Part D: Conceptual Checks

1.1. Why must ra be saved?

1.2. Why is a0 saved before the recursive call?

1.3. True or False: The stack grows toward higher memory addresses.