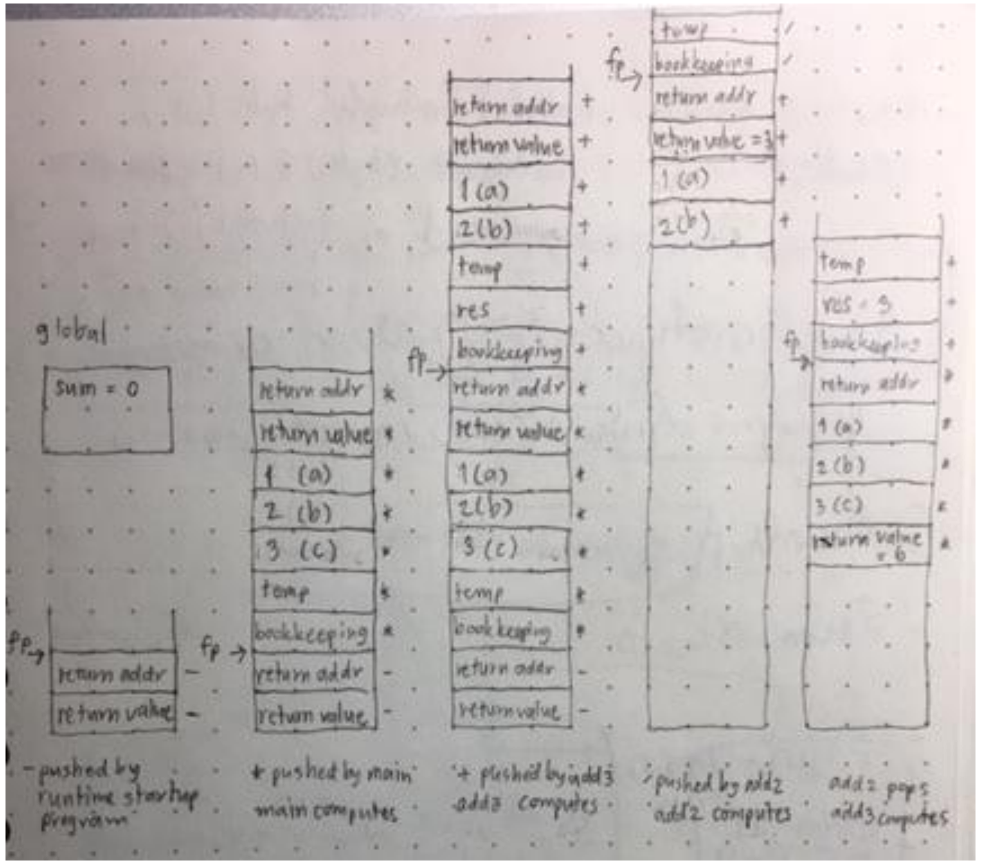
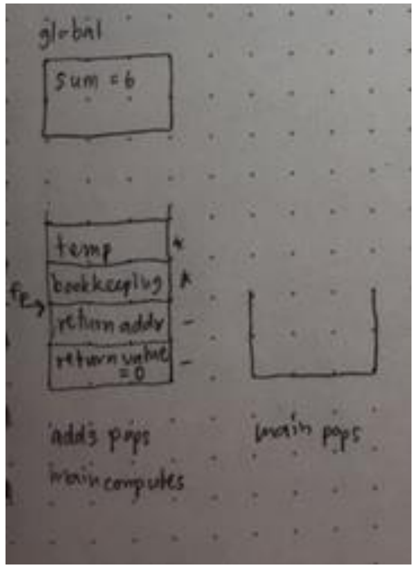
Exercise: Write sequence of stack allocation for calling add3()

```
int add2 (int a, int b) {
  return a + b;
}

int add3 (int a, int b, int c) {
   int res;
   res = add2(a, b);
   return res + c;
}

int sum = 0;
int main() {
  sum += add3 (1, 2, 3);
  return 0;
}
```





Exercise: Heap-Based Objects and Binding

Since lifetime means the time between creation and destruction,

Binding lifetime is <u>time between creation and destruction of a name-to-object binding.</u>

Object lifetime is <u>time between creation and destruction of an object.</u>

If object lifetime is longer than binding lifetime, we have <u>memory leak (or garbage).</u>

If binding lifetime is longer than object lifetime, we have <u>dangling reference.</u>

Exercise: Scope and Referencing Environment

```
//C
float op1(int x, float y) {
    int z;
    …
}
float op2(int z) { … }
```

op1 is considered one scope and op2 another scope.

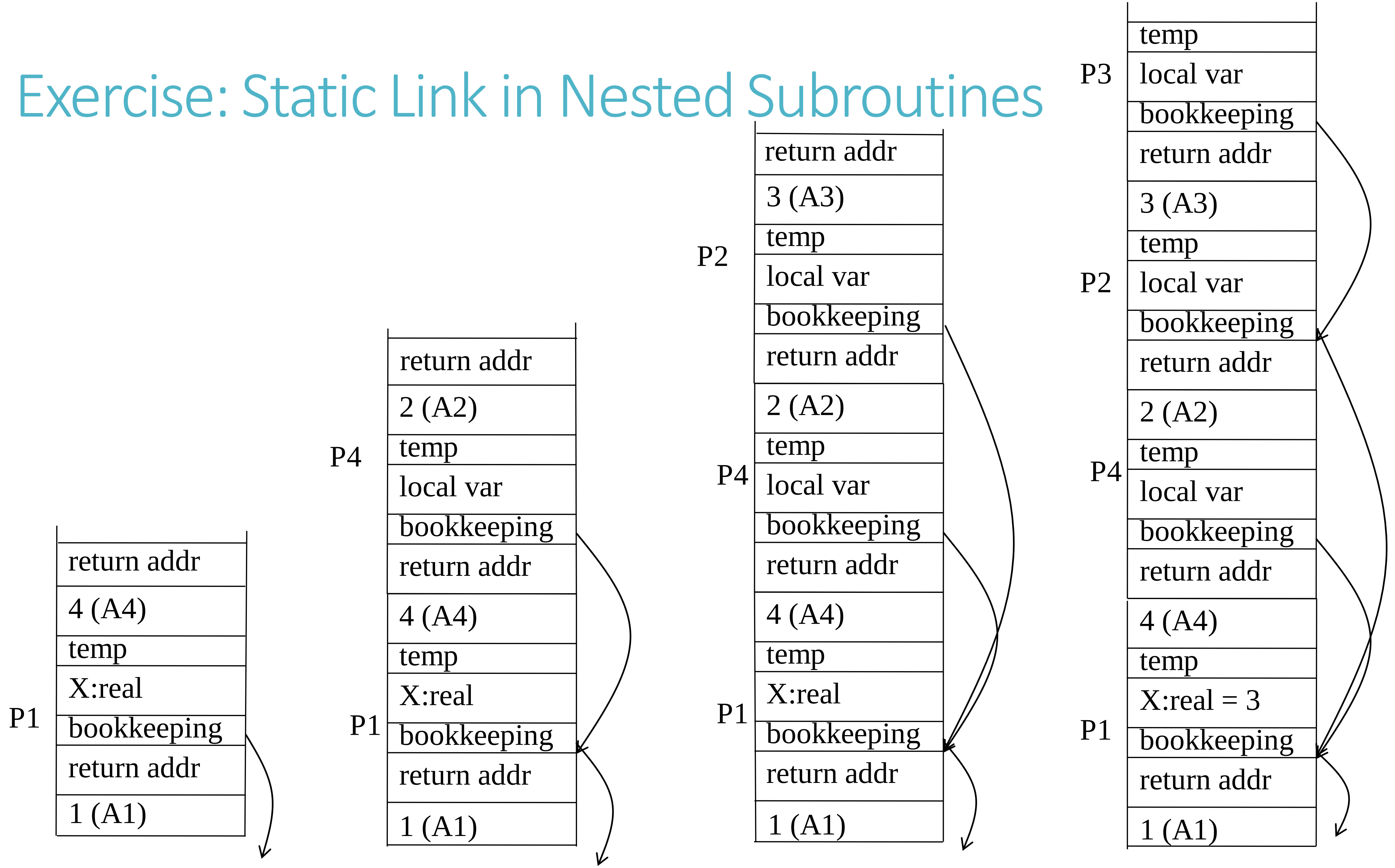
Scope of x and y is op1.

Scope of z is op1 (first z) and op2 (second z). There are two bindings of z.

Referencing environment of op1 consists of x, y, (first) z, op1, op2.

Referencing environment of op2 consists of (second) z, op2, op1.

Exercise: Static Link in Nested Subroutines



Exercise: Bindings in Dynamic Scoping

With dynamic scope rule, if first is entered from main

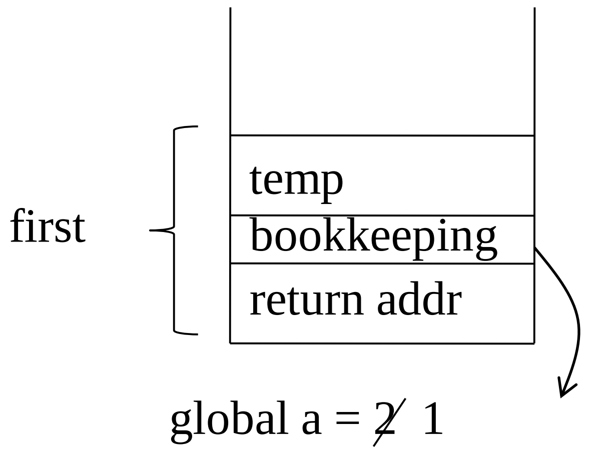
- What does write_integer refer to, global a or local a? <u>Global a</u>
- What does write_integer write? <u>Global a = 1</u>

```
a : integer          -- global declaration

procedure first
        a := 1

procedure second
        a : integer          -- local declaration
        first ()

a := 2
if read_integer () > 0
        second ()
else
        first ()
write_integer (a)
```


first
temp
bookkeeping
return addr
global a = ~~2~~ 1