

Matt Moore  
Gordon Gu

## Lab 1 Write-up

### Question 2

- A. The use of `pi` at line 4 is bound at which line? The use of `pi` at line 7 is bound at which line?
- a. `pi` at line 4 is bound to the `pi` at line 3 (`val pi`)
  - b. `pi` at line 7 is bound to `pi` at line 1 (because `pi` at line 3 is within the scope of the `pi` being used on line 4, whereas `pi` at line 1 is in the outer scope used by line 7)
- B. The use of `x` at line 3 is bound at which line? The use of `x` at line 6 is bound at which line? The use of `x` at line 10 is bound at which line? The use of `x` at line 13 is bound at which line?
- a. `x` at line 3 is bound to line 2 (because it is within the scope and is the parameter being used)
  - b. `x` at line 6 is bound to line 5 (because it is within that scope of the case at line 5)
  - c. `x` at line 10 is bound to line 5 (because it also is within the scope from the case at line 5)
  - d. `x` at line 13 is bound to line 1 (because it is outside all other scopes therefore is bound to the line of the same “level”)

### Question 3

- A. Is the body of `g` well-typed?
- a. Yes, the body of `g` is well typed, even without the return value. This is because regardless of input, the function will return some tuple value with a nested tuple value of two integers as its contained value, with another integer as the second contained value. Ex. `((Int, Int), Int)`.
- B. a: `Int` because  
    `1: Int`
- b: `(Int, Int)` because  
        `x: Int`  
        `3: Int`
- (b,1): `((Int, Int), Int)` because  
        b: `(Int, Int)`  
        `1: Int`
- (b, a + 2): `((Int, Int), Int)` because  
        b: `(Int, Int)`  
        a: `Int`  
        `2: Int`