

1. (a) int, 7
- (b) bool, false
- (c) int, 1
- (d) Ill-typed - ^ is the string concatenation operator.
- (e) Ill-typed - need an else statement
- (f) string, "positive"
- (g) Ill-typed - two distinct types
- (h) bool, false

Locate each binding in the code below. Then determine the scope of each identifier. (Give the range of lines.) If the binding is shadowed by a later one, indicate that as well. The first two are done as an example.

1	let profit_500 : int =	<i>Bound: profit_500, lines 7-23</i>
2	let price = 500 in	<i>Bound: price, lines 3-6</i>
3	let attendees = 120 in	<i>Bound: attendees, lines 4-6</i>
4	let revenue = price * attendees in	<i>Bound: revenue, lines 5-6</i>
5	let cost = 18000 + 4 * attendees in	<i>Bound: cost, line 6</i>
6	revenue - cost	
7		
8	let attendees (ticket_price : int) : int =	<i>Bound: attendees, lines 10-30</i>
9	(-15 * ticket_price)/10 + 870	
10		
11	let test () : bool =	<i>Bound: test, lines 13-30</i>
12	(attendees 500) = 120	<i>(shadowed after line 15)</i>
13	;; run_test "attendees at \$5.00" test	
14		
15	let test () : bool =	<i>Bound: test, lines 18-30</i>
16	(attendees 480) = 150	<i>(shadowed after line 28)</i>
17	;; run_test "attendees at \$4.80" test	
18		
19	let cost (ticket_price : int) : int =	<i>Bound: cost, lines 21-30</i>
20	18000 + (attendees ticket_price) * 4	
21		
22	let revenue (ticket_price : int) =	<i>Bound: revenue, lines 24-30</i>
23	(attendees ticket_price) * ticket_price	
24		
25	let profit (ticket_price : int) : int =	<i>Bound: profit, lines 27-30</i>
26	(revenue ticket_price) - (cost ticket_price)	
27		
28	let test () : bool =	<i>Bound: test, line 30</i>
29	(profit 500) = profit_500	
30	;; run_test "profit at \$5.00" test	