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.

```ocaml
let profit_500 : int = 
  let price     = 500 in 
  let attendees = 120 in 
  let revenue   = price * attendees in 
  let cost      = 18000 + 4 * attendees in 
  revenue - cost

let attendees (ticket_price : int) : int = 
  (-15 * ticket_price)/10 + 870

let test () : bool = 
  (attendees 500) = 120
  ;; run_test "attendees at $5.00" test

let cost (ticket_price : int) : int = 
  18000 + (attendees ticket_price) * 4

let revenue (ticket_price : int) = 
  (attendees ticket_price) * ticket_price

let profit (ticket_price : int) : int = 
  (revenue ticket_price) - (cost ticket_price)

let test () : bool = 
  (profit 500) = profit_500
  ;; run_test "profit at $5.00" test
```

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