

Type inference





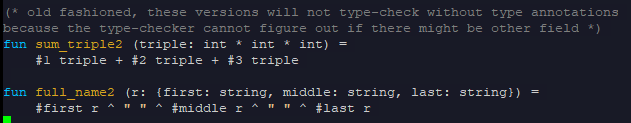
* Inferred type: int
  + Because you use addition in the function body





* Inferred type: string
  + Because you use concatenation in the function body

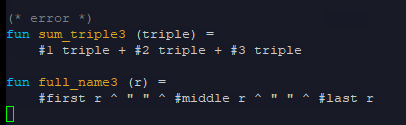
Old fashioned way

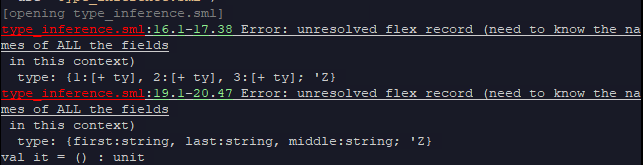


This will run smoothly because you define what type is the argument triple and r

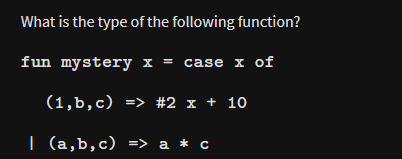


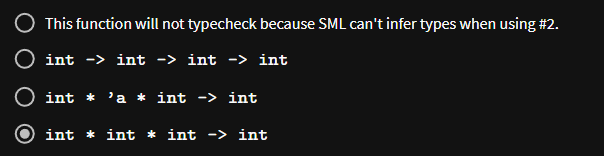
But when you did not define any type for both triple and r:

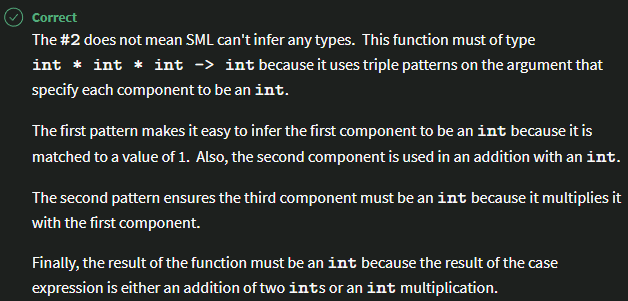


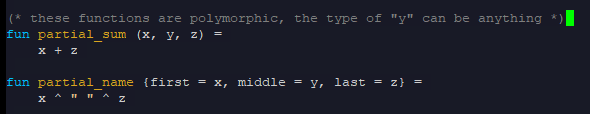


* This is why in homework 1, we also write the types for the arguments whenever we will use tuples as a parameter











Because SML cannot infer the type of y because it is not used in the function body

This is okay if the type returned is like this when the homework is asking for a (int \* int \* int) -> int type



