```
module Task where
data JValue = JString String
    | JNumber Double
    | JBool Bool
    | JNull
    | JObject [(String, JValue)]
    | JArray   [JValue]


renderJValue :: JValue → String
renderJValue (JString s)   = show s
renderJValue (JNumber n)   = show n
renderJValue (JBool True)  = "true"
renderJValue (JBool False) = "false"
renderJValue JNull         = "null"


renderJValue (JObject o) = "{" ++ renderPairs o ++ "}"
renderJValue (JArray a) = "[" ++ renderPairs a ++ "]"

renderPair :: (String, JValue) → String
renderPair (k,v)   = show k ++ ": " ++ renderJValue v


renderPairs [] = ""
renderPairs [p] = renderPair p
renderPairs (p:ps) = renderPair p ++ "," ++ renderPairs ps


renderArrayValues [] = ""
renderArrayValues [v] = renderJValue v
renderArrayValues (v:vs) = renderJValue v ++ "," ++ renderArrayValues vs
```

🟪 is annotated to have the type 🟦
at the 1st argument

Chameleon cannot assign types to the following expressions:

TYPE A→B	EXPRESSION f(x)	TYPE A→B
[(String, JValue)] → JValue	JArray	[JValue] → JValue
[(String, JValue)]	a	[JValue]
[(String, JValue)] → a	renderPairs	[JValue] → a
(String, JValue)	p	JValue
(String, JValue) → a	renderPair	JValue → a