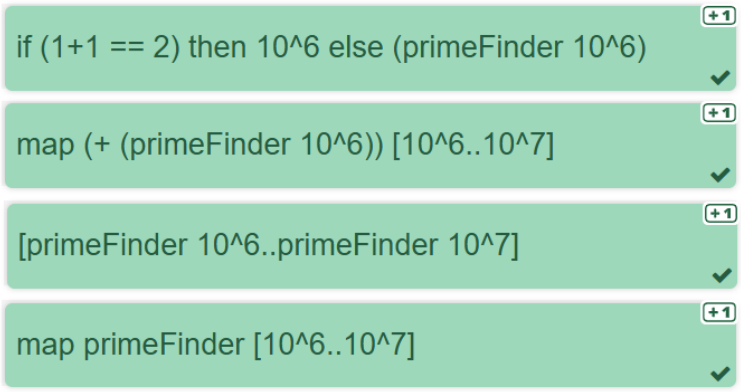
Lab Sheet 8.1

a)



b) The *currentTime* function returns a value, but this value changes due to time changes, therefore this function will not be referentially transparent. For the code to be referentially transparent, the function could be passed as an argument to *secsSinceMidnight* by using the IO type. IO actions must be explicitly executed to get a result. For referential transparency to be respected pure values must always evaluate to the same result while impure operations like fetching the time are isolated in the IO monad. The referential transparency is preserved because the IO type separates side effects from pure computations.

```
import Data.Time.Clock
import Data.Time.LocalTime

--I have created a new data type for Time that
--stores hours, minutes and seconds as integers
data Time = Time Int Int Int

currentTime :: IO Time
currentTime = do
    now <- getZonedTime
    let (TimeOfDay h m s) = localTimeOfDay $ zonedTimeToLocalTime now
    return (Time h m (floor s))

--This function should return the number of seconds since midnight
secsSinceMidnight :: IO Time -> IO Int
secsSinceMidnight timeAction = do
    (Time h m s) <- timeAction
    return (3600 * h + 60 * m + s)
```