Started on	Thursday, 3 November 2022, 10:51 AM
State	Finished
Completed on	Thursday, 3 November 2022, 11:00 AM
Time taken	9 mins 50 secs
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 6.00 out of 6.00

Match the following expressions in Haskell with their corresponding types.

```
data Maybe a = Nothing | Just a
```

`Just`	a -> Maybe a	✓
`Just Nothing`	Maybe (Maybe a)	✓
`Nothing`	Maybe a	✓
`return (Just Just)`	IO (Maybe (a -> Maybe a))	✓
`Just (return Just)`	Maybe (IO (a -> Maybe a))	✓
`return Just`	IO (a -> Maybe a)	✓
`return Nothing`	IO (Maybe a)	✓
`Just return`	Maybe (a -> IO a)	✓
`Just (return Nothing)`	Maybe (IO (Maybe a))	✓

Your answer is correct.

The correct answer is: `Just` → a -> Maybe a, `Just Nothing` → Maybe (Maybe a), `Nothing` → Maybe a, `return (Just Just)` → IO (Maybe (a -> Maybe a)), `Just (return Just)` → Maybe (IO (a -> Maybe a)), `return Just` → IO (a -> Maybe a), `return Nothing` → IO (Maybe a), `Just return` → Maybe (a -> IO a), `Just (return Nothing)` → Maybe (IO (Maybe a))

Question **2**

Correct

Mark 4.00 out of 4.00

Select all TRUE statements about Haskell programming language.

Select one or more:

- [✓] a. It is possible, but considered bad practice to define partial functions in Haskell. ✓
- [ ] b. Haskell has a weak dynamic typing system.
- [ ] c. Haskell facilitates object-oriented programming (provides features supporting OOP).
- [✓] d. Haskell facilitates imperative programming (provides features supporting imperative programming). ✓
- [✓] e. Haskell facilitates functional programming (provides features supporting functional programming). ✓
- [ ] f. Haskell has strict (eager) evaluation strategy.
- [ ] g. Haskell has a weak static typing system.
- [✓] h. Haskell is the best programming language. ✗
- [✓] i. Haskell allows working naturally with (potentially) infinite data structures. ✓
- [✓] j. Haskell has a strong static typing system. ✓
- [ ] k. Haskell has a strong dynamic typing system.
- [✓] l. Haskell has non-strict (lazy) evaluation strategy. ✓
- [ ] m. It is impossible to define partial functions in Haskell.
- [ ] n. It is considered a normal practice to define partial functions in Haskell.

Your answer is correct.

The correct answers are:
Haskell has a strong static typing system.,

Haskell has non-strict (lazy) evaluation strategy.,

Haskell facilitates imperative programming (provides features supporting imperative programming).,

Haskell facilitates functional programming (provides features supporting functional programming).,

Haskell allows working naturally with (potentially) infinite data structures.,

It is possible, but considered bad practice to define partial functions in Haskell.