

Question 9

Not yet  
answered

Marked out of  
1.00

🚩 Flag question

Select the **true** statements about monads in Haskell:

- ☐ a. `Int` is an example of a Monad
- ☒ b. Monad is an alias for Monoid
- ☐ c. `[a]` (List) is an example of a Monad
- ☒ d. Monad defines the function `>>=`

Question 8

Not yet

answered

Marked out of

1.00

Flag question

Select all the **false** statements about Input/Output in Haskell

- ☒ a. To obtain a line from the standard input, we can write  
do  
    name <- getLine  
    putStrLn name
- ☐ b. Haskell's *main* function has the signature *main :: IO ()*
- ☐ c. To read data from a file we use the *read* function    *readFile* *sau* *readLn*
- ☐ d. do notation can be only used with the IO monad

Previous page

Time left 0:04:06

Question 7

Not yet  
answered

Marked out of  
1.00

Flag question

Given the following combinator:

```
rep :: Int -> Parser a -> Parser [a]
rep 0 p = succeed []
rep n p = pMap (\(a, as) -> a:as) $ andThen p (rep (n-1) p)
```

That applies a given parser a fixed number of times and returns the results in a list.

Select the parser definition that would yield:

`Success ("ab", "3")`

for the following input:

`ab123`

i.e. `runParser p input == result`

**Hint: Try to find a pattern in the input and connect that with the output before considering the parser definitions below!**

- ☐ a. `p = pThen (pThen lower lower) (pThen digit digit)`
- ☐ b. `p = pMap \(a, b) -> a ++ b $ andThen (rep 2 lower) (rep 2 digit)`
- ☒ c. `p = pMap fst $ andThen (rep 2 lower) (rep 2 digit)`
- ☐ d. `p = pThen (rep 2 lower) (rep 2 digit)`

5

Which of the following names would best describe the following parser:

```
satisfies (^elem` ['0'..'9'])
```

- ☐ a. upper
- ☐ b. lower
- ☒ c. digit
- ☐ d. char

Question 4

Get

Answered

100% out of

100% question

Select the function signature that **best** represents a parser

- ☐ a. `String -> a`
- ☒ b. `String -> Result ParseError (a, String)`
- ☐ c. `[Int] -> Result Int a`
- ☐ d. `String -> Result ParseError a`

Question **3**

Not yet  
answered

Marked out of  
1.00

🚩 Flag question

Select the function that is the equivalent of the following function written in do notation

```
fn = do
  putStrLn "Line to reverse"
  line <- getLine
  putStrLn (reverse line)
```

- ☒ a. `putStrLn "Line to reverse" >> getLine >>= \line -> putStrLn (reverse line)`
- ☐ b. `putStrLn "Line to reverse" >>= getLine >>= \line -> putStrLn (reverse line)`
- ☐ c. `putStrLn "Line to reverse" >> getLine >> \line -> putStrLn (reverse line)`
- ☐ d. `putStrLn "Line to reverse" >>= getLine >>= putStrLn (reverse line)`

## Question 2

Not yet  
answeredMarked out of  
1.00

Flag question

Given the following combinator:

```
rep :: Int -> Parser a -> Parser [a]
rep 0 p = succeed []
rep n p = pMap (\(a, as) -> a:as) $ andThen p (rep (n-1) p)
```

That applies a given parser a fixed number of times and returns the results in a list.

Select the parser definition that would yield:

Success ("123", "")

for the following input:

AA1BB2CC3

i.e. runParser p input == result

**Hint: Try to find a pattern in the input and connect that with the output before considering the parser definitions below!**

- ☐ a.  $p = \text{rep } 3 \text{ (andThen (rep } 2 \text{ upper) digit)}$
- ☒ b.  $p = \text{rep } 3 \text{ (pThen (pThen upper upper) digit)}$
- ☐ c.  $p = \text{rep } 3 \text{ (andThen (pThen upper upper) digit)}$
- ☐ d.  $p = \text{andThen } s \text{ (andThen } s \text{ } s) \text{ where}$   
 $\quad \text{ld} = \text{(pThen (pThen upper upper) digit)}$

Time left 0:11:14

6 7

Question 1

Not yet  
answered

Marked out of  
1.00

Flag question

Given the following parser:

```
p = number `andThen` many (pThen (char ',') number)
```

Select the inputs that will successfully parse (i.e. will yield Success \_).

**Note: the parser doesn't have to consume all of the input in order to yield the Success variant!**

**Hint: Try to express in words (natural language) what the parser does before considering the inputs below.**

- ☒ a. 1,2,3
- ☐ b. 123,4,5,abc
- ☐ c. 123,4,5,
- ☐ d. 1

Next page