

PF - serie A si B (2022/2023 - R. R. Slavescu) / 16 January - 22 January / Lun-Test-2-145-cra-8.15

6 7

Question 1
Not yet answered
Marked out of 1.00
1 Flag question

Time left 0:11:14

Given the following parser:
 $p = \text{number} \text{ andThen } \text{many } (p \text{Then } (\text{char } ',') \text{ number})$

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

Lab 14: Test (Labs -11-12 10%) and Make-up

Given the following parser

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. $p = \text{andThen s (andThen s s) where}$
 $\text{ld} = (\text{pThen (pThen upper upper) digit})$

Given the following combinator

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)`

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

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
- ☒ d. `do` notation can be only used with the IO monad

Select all the false statements about Input/Output in Haskell

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 `>>=`

Select the true statements about monads in Haskell:

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

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

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

Which of the following names would best describe the following parser

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`

Select the function signature that best represents a parser