Question **5**
Complete
Mark 1.00 out of 1.00
Flag question

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

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

Your answer is correct.

Information
Flag question

The next 3 questions are intermediate questions.

Read each question carefully.

Question **6**
Complete
Mark 1.00 out of 1.00
Flag question

Complete the parser below such that it parses a C\C++ array indexing expression (i.e. [1][2]):

Note: there must be at least one indexing expression.

```
cArrayIndex = some $ between (char '[') (char ']') number
```

Your answer is correct.

Question **3**
Complete
Mark 1.00 out of 1.00
Flag question

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

- a. Monad defines the function <$>
- b. Int is an example of Monad
- c. Maybe is an example of a Monad
- d. Monad is a type class

Your answer is correct.

Question **4**
Complete
Mark 1.00 out of 1.00
Flag question

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

```
satisfies (==c)
```

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

Your answer is correct.

Question **1**
Complete
Mark 0.33 out of 1.00
Flag question

Match the concepts:

Mappable types that can also unpack nested structures in results — Applicative
Mappable types — Functor
Generalized mappable types — Monad

Your answer is partially correct.
You have correctly selected 1.

Question **2**
Complete
Mark 1.00 out of 1.00
Flag question

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

- [x] a. To read data from a file we use the *read* function
- [ ] b. To obtain a line from the standard input, we can write

```
do
  name <- getLine
  putStrLn name
```

- [x] c. do notation can be only used with the IO monad
- [ ] d. Haskell's *main* function has the signature *main :: IO ()*

Your answer is correct.

Question **10**
Complete
Mark 1.00 out of 1.00
Flag question

Given the following code:

```
newtype Any = Any Bool
instance Semigroup Any where
   (Any a) <> (Any b) = Any (a || b)
instance Monoid Any where
   mempty = Any False
```

The result of the following expressions is:

`foldl (<>) mempty (map (\x -> Any (x `mod` 2 == 0)) [1, 2, 3])` — Any True

`foldl (<>) mempty (map (\x -> Any (x >= 2)) [2, 3, 4])` — Any True

Your answer is correct.

Question **9**

Complete

Mark 1.00 out of 1.00

Flag question

Given the following code that generates the hamming numbers:

```
merge3 x y z = merge (merge x y) z where
    merge (u:us) (v:vs)
        | u < v = u:merge us (v:vs)
        | u > v = v:merge (u:us) vs
        | otherwise = u:merge us vs

ham :: [Integer]
ham = 1:merge3 ham2 ham3 ham5

ham2 = [ 2*i | i <- ham ]

ham3 = [ 3*i | i <- ham ]

ham5 = [ 5*i | i <- ham ]

hammingGen :: Int -> [Integer]
hammingGen n = take n ham
```

Select what will be printed for each of the following commands after evaluating:

```
hammingGen 3
```

```
> :sprint ham3
ham3 =
```
3 : _

```
> :sprint ham2
ham2 =
```
2 : 4 : _

Question **7**

Complete

Mark 1.00 out of 1.00

Flag question

The following list comprehension:

```
[(x, y) |x <- [1, 2], y <- ['a', 'b']]
```

- a. Returns [(1, 'a'), (2, 'b')]
- b. Returns: [(1, 'a'), (1, 'b'), (2, 'a'), (2, 'b')]
- c. Fails to combile because the syntax is invalid
- d. Fails to combile because x and y have different types

Your answer is correct.

Question **8**

Complete

Mark 1.00 out of 1.00

Flag question

Select all the **false** statements about type classes

- a. Type classes are used to define classes, a special kind of data definition that includes methods and private fields
- b. Type classes are used to abstract common behavior for various types (like Java interfaces)
- c. All instance implementations for a type class must be in the module where the type class
- d. We can implement type classes defined by the standard library for our own types

Question **5**
Complete
Mark 1.00 out of 1.00
Flag question

Given the following function definition:

```
f :: [String] -> Int
f ["x", "y"] = 1
f ["x", "y", "z"] = 2
f ["x", _] = 3
f ("x":_) = 4
f _ = 5
```

the result of the following function call is:

```
f ["x", "z", "y"]
```

Answer: 4

Information
Flag question

The next 3 questions are intermediate questions.

Read each question carefully.

Question **6**
Complete
Mark 0.50 out of 1.00
Flag question

Select all the **false** statements about the bottom value:

- [x] a. In Haskell, None is the bottom value
- [ ] b. If an expression has the bottom type, any value can be assigned to it
- [ ] c. In Haskell, undefined is the bottom value
- [ ] d. It crashes the program if it's evaluated at runtime

Question **3**
Complete
Mark 1.00 out of 1.00
Flag question

Which of the following are examples of **not valid** ways to create local definitions in Haskell?

- [ ] a. y * 2 where y = 5
- [x] b. local y = 5 in y * 2
- [ ] c. let y = 5 in y * 1
- [x] d. y * 2 with y = 5

Your answer is correct.

Question **4**
Complete
Mark 1.00 out of 1.00
Flag question

Select the function that uses pattern guards correctly to implement the difference function:

- (•) a.
```
difference [] _ = []
difference (a:as) b
  | a `elem` b = difference as b
  | otherwise = a:difference as b
```
- ( ) b.
```
difference [] _ = []
difference (a:as) b =
  % a `elem` b -> difference as b
  % otherwise -> a:difference as b
```
- ( ) c.
```
difference [] _ = []
difference (a:as) b =
  if a `elem` b -> difference as b
  otherwise ->a:difference as b
```

Question **1**
Complete
Mark 0.67 out of 1.00
Flag question

Which function describes best the each of the following list comprehensions?

[x | x <- xs, length x > 2] filter

[drop 2 x | x <- xs] take

[not x | x <- xs] map

Your answer is partially correct.
You have correctly selected 2.

Question **2**
Complete
Mark 1.00 out of 1.00
Flag question

Select the snippets that are **valid** Haskell code

a. len [] = 0
len (x:xs) = 1 + len xs

b. allSame : a -> a -> a -> Bool
allSame a b c = (a == b) && (b == c)

c. allSame :: Eq a -> a -> a -> a -> Bool
allSame a b c = (a == b) && (b == c)

d. len [] = 0
len (x::xs) = 1 + len xs

Question **9**
Complete
Mark 0.00 out of 1.00
Flag question

Select all the **true** statements:

a. Elm has "built-in" definitions for all possible record accessors
b. Record accessors can be composed with function composition
c. The type of the accessor .a is {a : b} -> b
d. Records use nominal typing

Your answer is incorrect.

Question **10**
Complete
Mark 0.00 out of 1.00
Flag question

The result of the following expression is:

```
type alias Point = {x: Int, y: Int}

points = [{x = 1, y = 2}, {x = 2, y = 4}, {x = 3, y = 5}]

mx : Int -> List Point -> List Point
mx d ps = ps |> List.map (\p -> { p | x = p.x + d })

my : Int -> List Point -> List Point
my d ps = ps |> List.map (\p -> { p | y = p.y + d })

points |> mx 1 |> my 2 |> List.map .x |> List.foldl (+) 0
```

Answer: 23

Question **7**
Complete
Mark 0.00 out of 1.00
Flag question

In the context of functions used for testing HTML, select the functions that belong to the Html.Test.Selectors module

- a. tag
- b. equal
- c. text
- d. class
- e. has

Your answer is incorrect.

Question **8**
Complete
Mark 0.00 out of 1.00
Flag question

Given the following definitions:

```
inc x = x + 1
dec x = x - 1
double x = x * 2
twice f x = f (f x)
```

=) (3-1)*2 + 1 = 5
=) (5-1)*2 + 1 = 9

What does the expression below evaluate to?

```
(twice (dec >> double >> inc)) 3
```

Answer: 9

Question **3**
Complete
Mark 1.00 out of 1.00
Flag question

The function countVowels can be rewritten using pipelines as:

countVowels s = List.length (List.filter isVowel (List.map Char.toLower s))

- a. countVowels s = s |> List.map Char.toLower |> List.filter isVowel |> List.length
- b. countVowels s = List.length <| List.filter isVowel <| List.map Char.toLower <| s
- c. countVowels s = s |> List.length |> List.filter isVowel |> List.map Char.toLower
- d. countVowels s = List.map Char.toLower <| List.filter isVowel <| List.length <| s

Your answer is correct.

Question **4**
Complete
Mark 0.00 out of 1.00
Flag question

In the context of Elm web apps, the Msg type represents:

- a. All possible actions that can cause the app to change its state
- b. The messages a users send in chat apps
- c. The difference between two states of the app
- d. The state of the app

Question **6**
Complete
Mark 0.00 out of 1.00
Flag question

Given the following definitions:

```
type CalculationError = FirstNaN | SecondNaN

divNums : Float -> Float -> Result CalculationError Float
divNums a b =
  if isNaN a then
    Err FirstNaN
  else if isNaN b then
    Err SecondNaN
  else
    Ok (a / b)
```

If we want to write a function that calls divNums and returns Result String Float, we have the following options:

- [x] a. Use a case expression to transform the error
- [x] b. Use Result.withDefault
- [ ] c. Use Result.mapError
- [ ] d. Use Result.map

Question **1**
Complete
Mark 1.00 out of 1.00
Flag question

To get the value that is inside the Just variant of Maybe or provide a default value, we can:

- [ ] a. Use the Maybe.unwrap function
- [ ] b. Use an if expression
- [x] c. Use the Maybe.withDefault function
- [x] d. Use a case expression

Your answer is correct.

Question **2**
Complete
Mark 0.00 out of 1.00
Flag question

The following code snippet:

```
view = div [] [style "color" "red", text "Some text"]
```

div [style "color" "red"] [text "Some text"]

- [ ] a. Will render the text "Some text" with red color
- [ ] b. Will generate invalid HTML that causes the browser to show an error
- [x] c. Will render the text "Some text" with black color, because the style is not applied correctly
- [ ] d. Will fail to compile

Your answer is incorrect.

Question **5**
Complete
Mark 1.00 out of 1.00
Flag question

Given the following function definition:

```
f x a b = x |> a |> b |> b
```

Se executa de 2 ori

The result of the following expression is:

```
f 3 (\x -> x + 5) (\x -> x * 2)
```

Answer: 32

Question **10**
Incorrect
Mark 0.00 out of 1.00
Flag question

Select the function(s) which could be rewritten in a point-free style:

- a. topN n l = List.take n (List.sort l)
- b. product l = List.foldl (*) 1 l
- c. append l1 l2 = List.foldr (::) l2 l1 ✖
- d. topN n l = (\li -> List.take n (List.sort li)) l

Question **9**
Incorrect
Mark 0.00 out of 1.00
Flag question

Which is a valid method of removing "Nothing" from a list of Maybe and also unbox whatever information is inside "Just"s from the input list?

a. ✖
```
removeNothingAndUnbox list =
 let
  aux l acc =
   case l of
    [] -> acc
    x::xs -> case x of
              Nothing -> aux xs acc
              Just a -> aux xs (x::acc)
 in
  aux list []
```

b.
```
removeNothingAndUnbox list =
 let
  aux l acc =
   case l of
    [] -> acc
    x::xs -> case x of
              Nothing -> aux xs acc
              Just a -> aux xs (a::acc)
 in
  aux list []
```

c.
```
removeNothingAndUnbox list = filter (\x -> x /= Nothing) list
```

Question **6**
Correct
Mark 1.00 out of 1.00
Flag question

For each example below, select the appropriate signature.

(\x -> \y -> \z -> z + y + z) 1	: number -> number -> number ✔
(\x -> \y -> \z -> z + y + z) 1 2 3	: number ✔
(\x -> \y -> \z -> z + y + z)	: number -> number -> number -> number ✔
(\x -> \y -> \z -> z + y + z) 1 2	: number -> number ✔

Question **7**
Correct
Mark 1.00 out of 1.00
Flag question

Given the following function definition:

```
mul a b = if b == 0 then 0 else a + mul a b - 1
```

What does `mul 3 2` evaluate to?

- a. RangeError: Maximum call stack size exceeded (Stack overflow) ✔
- b. 4
- c. 6

Question **8**
Correct
Mark 1.00 out of 1.00
Flag question

Which of the following is true for type variables?

- [x] a. they range over types
- [ ] b. their name can only contain only one letter
- [x] c. indicate same type between multiple arguments
- [x] d. they are only described in lowercase

Question **4**
Incorrect
Mark 0.00 out of 1.00
Flag question

Given the following function definition:

```
pow n i = if i == 0 then 1 else n * pow n (i-1)
```

The result of the following function call will be:

```
pow 5 2 - 1
```

25 − 1 = 24

Answer: 25

Question **5**
Correct
Mark 1.00 out of 1.00
Flag question

What is a predicate?

- (•) a. a function which returns a boolean
- ( ) b. a function which has a single argument
- ( ) c. a boolean field of a record

Question **1**
Incorrect
Mark 0.00 out of 1.00
Flag question

What will the following call evaluate to? howBig 2

howBig n = if n < 10 then "Small" else if n < 100 then "Medium"

- (•) a. "Small"
- ( ) b. "Medium"
- (•) c. Nothing, because of a compile time error
- ( ) d. Nothing, because of a runtime error

Question **2**
Correct
Mark 1.00 out of 1.00
Flag question

Which of these expressions will match a list containing exactly **one** element?

- [ ] a. No other answer is correct
- [ ] b. `[]`
- [x] c. `[x]`
- [x] d. `x::[]`

Question **3**
Correct
Mark 1.00 out of 1.00
Flag question

What will filteredList evaluate to?

filteredList = List.filter (\x -> x >= 10) [11, 10, 9, 5, 15, 6, 0]

Answer: [11, 10, 15]

Which of the following are examples of **not valid** ways to create local definitions in Haskell?

- [ ] a. local y = 5 in y * 2
- [ ] b. y * 2 where y = 5
- [ ] c. let y = 5 in y * 1
- [ ] d. y * 2 with y = 5

Which function describes best the each of the following list comprehensions?

[x | x <- xs, length x > 2] filter

[drop 2 x | x <- xs] take

[not x | x <- xs] filter

Choose...
filter
map
fold
take

Select all the **true** statements about type classes

- [ ] a. Type classes are used to abstract common behavior for various types (like Java interfaces)
- [ ] b. We can implement type classes defined by the standard library for our own types
- [ ] c. Type classes are used to define classes, types that also have methods and private fields
- [ ] d. All type class implementations for a data type must be in the module where the data is defined

Select all the **true** statements about type classes

- [ ] a. Type classes are used to organize related types in a file
- [ ] b. Any type class can be implemented for any type
- [ ] c. All type class implementations for a data type must be in the module where the data is defined
- [ ] d. Type classes are used to define a common interface for a set of operations that can be performed on various types

Select all the **true** statements about the bottom value:

- [ ] a. In Haskell, undefined is the bottom value
- [ ] b. In Haskell, Nothing is the bottom value
- [ ] c. The bottom value can be assigned to any type
- [ ] d. The compiler won't compile (i.e. will show an error) programs that contain the bottom value

Given the following code:

```
newtype Any = Any Bool
instance Semigroup Any where
   (Any a) <> (Any b) = Any (a || b)
instance Monoid Any where
   mempty = Any False
```

The result of the following expressions is:

```
foldl (<>) mempty (map (\x -> Any (x `mod` 2 == 0)) [1, 2, 3])
```

Choose...

```
foldl (<>) mempty (map (\x -> Any (x >= 2)) [2, 3, 4])
```

Choose...
Any True
True
False

Select the function that uses pattern guards correctly to implement the difference function:

- a.
```
difference [] _ = []
difference (a:as) b =
  % a `elem` b -> difference as b
  % otherwise -> a:difference as b
```
- b.
```
difference [] _ = []
difference (a:as) b =
  if a `elem` b -> difference as b
  otherwise ->a:difference as b
```
- c.
```
difference [] _ = []
difference (a:as) b
  | a `elem` b = difference as b
  | otherwise = a:difference as b
```

Clear my choice

Given the following code that generates the hamming numbers:

```
merge3 x y z = merge (merge x y) z where
    merge (u:us) (v:vs)
        | u < v = u:merge us (v:vs)
        | u > v = v:merge (u:us) vs
        | otherwise = u:merge us vs

ham :: [Integer]
ham = 1:merge3 ham2 ham3 ham5

ham2 = [ 2*i | i <- ham ]

ham3 = [ 3*i | i <- ham ]

ham5 = [ 5*i | i <- ham ]

hammingGen :: Int -> [Integer]
hammingGen n =
```

Select what will l ch of the following commands after evaluating:

```
hammingGen 4
```

Choose...
2 : 4 : 6 : _
5 : _
2 : 4 : _
5 : 10 : _
2 : _

```
> :sprint ham2
ham2 =
```

Choose...

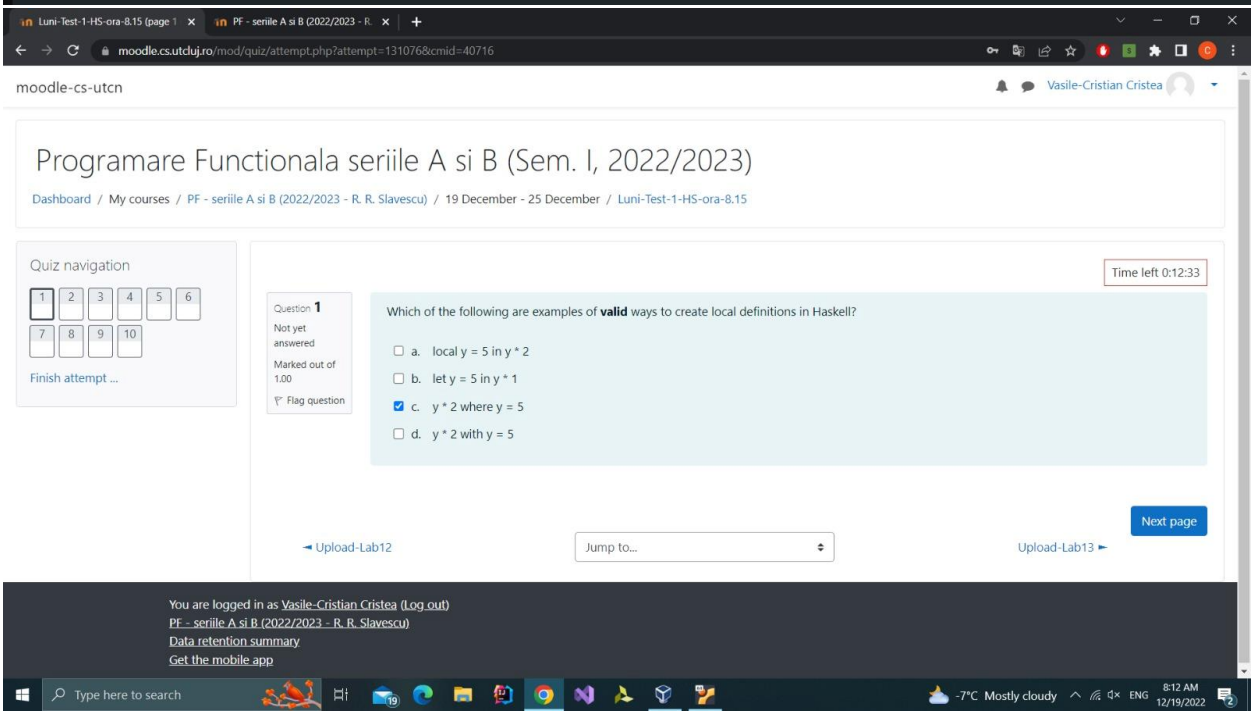
```
> :sprint ham5
ham5 =
```

Choose...

Select the function that uses pattern guards correctly to implement the dropWhile function:

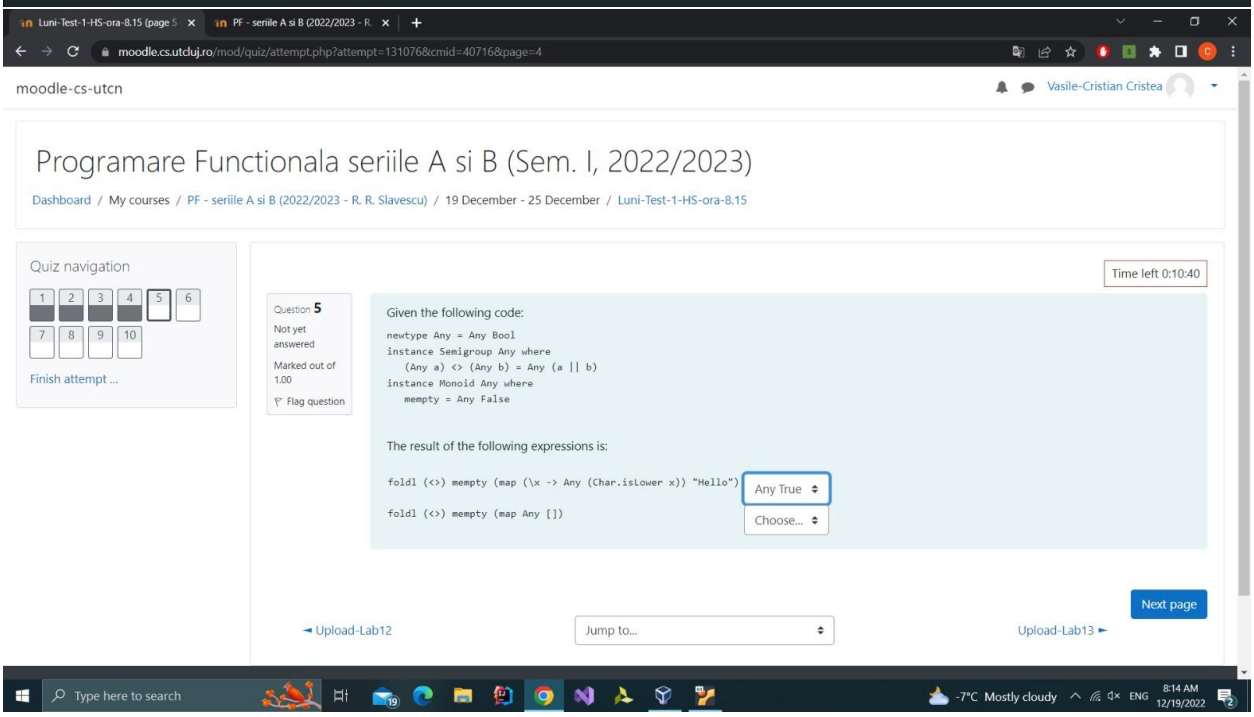
- a.
```
dropWhile _ [] = []
dropWhile p (x:xs)
  | p x = dropWhile p xs
  | otherwise = x:xs
```
- b.
```
dropWhile _ [] = []
dropWhile p (x:xs)
  | p x -> dropWhile p xs
  | else -> x:xs
```
- c.
```
dropWhile _ [] = []
dropWhile p (x:xs) =
  when p x -> dropWhile p xs
  else x:xs
```

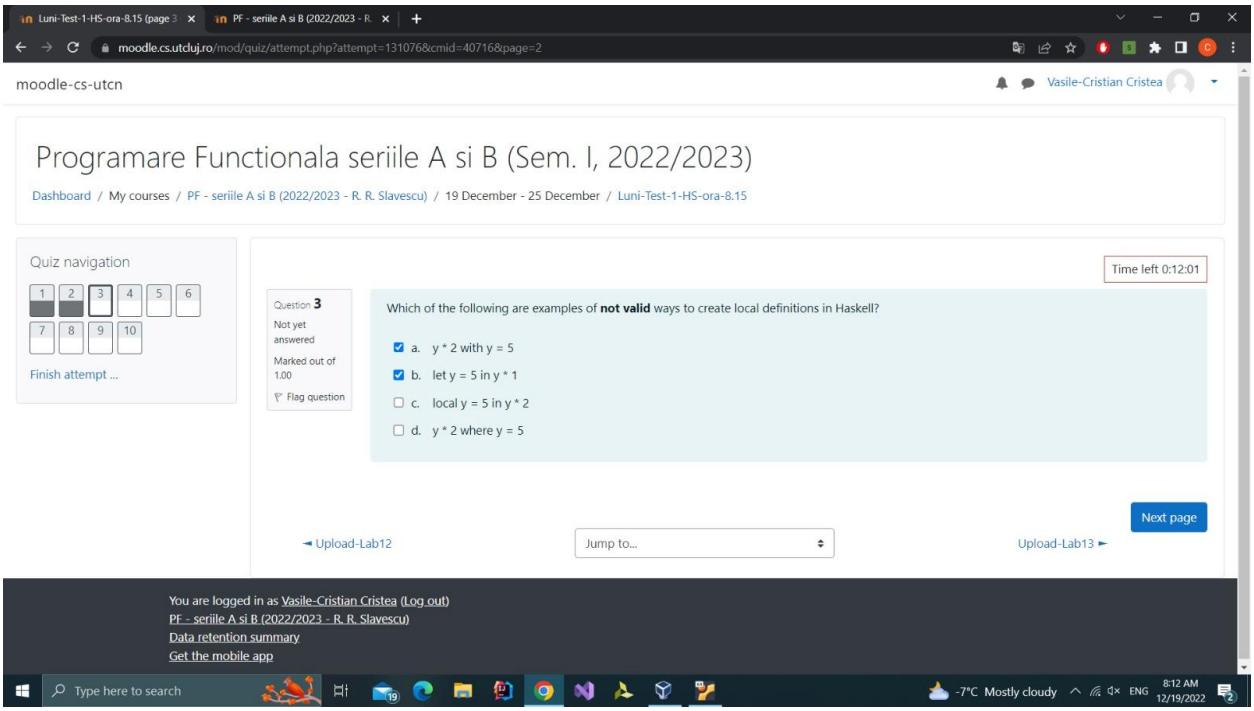
Clear my choice

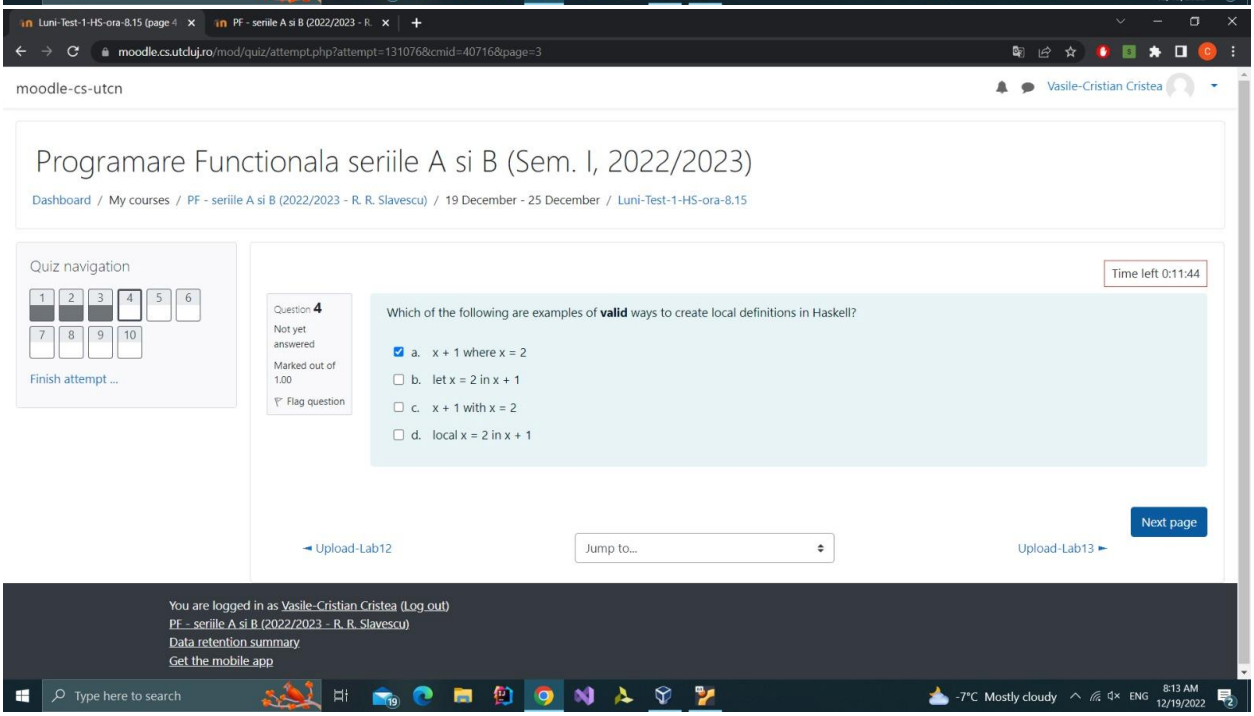


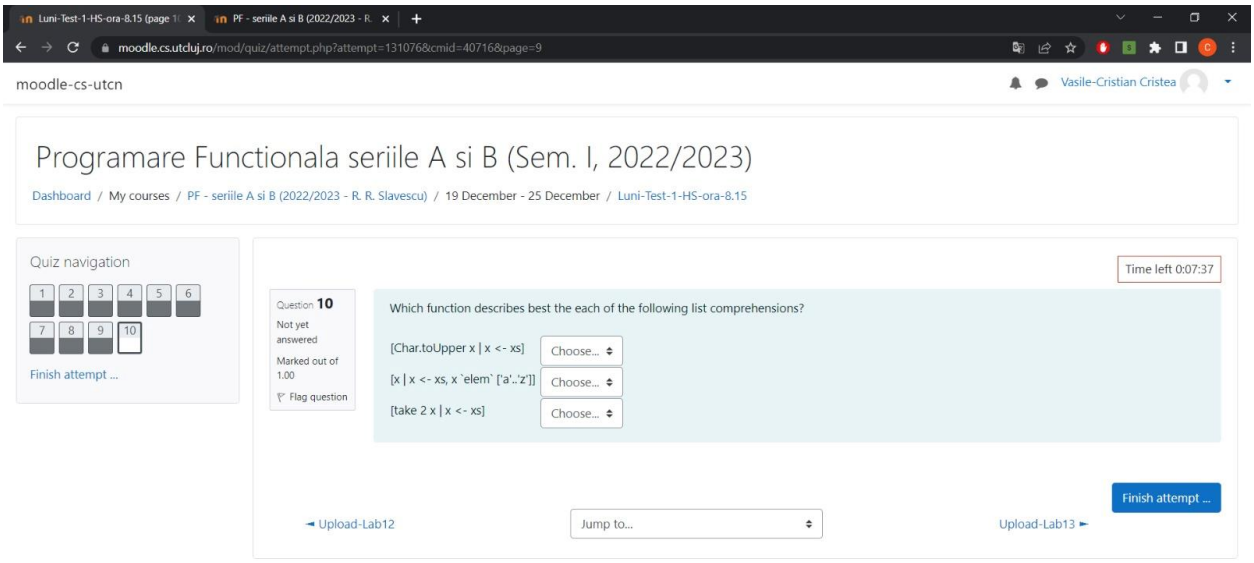
Select the function that uses pattern guards correctly to implement the filter function:

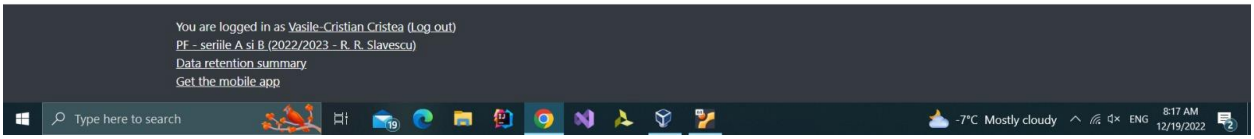
- a.
```
filter _ [] = []
filter p (x:xs) =
  if p x then x:filter p xs
  otherwise filter p xs
```
- b.
```
filter _ [] = []
filter p (x:xs)
  | p x -> x:filter p xs
  | else -> filter p xs
```
- c.
```
filter _ [] = []
filter p (x:xs)
  | p x = x:filter p xs
  | otherwise = filter p xs
```

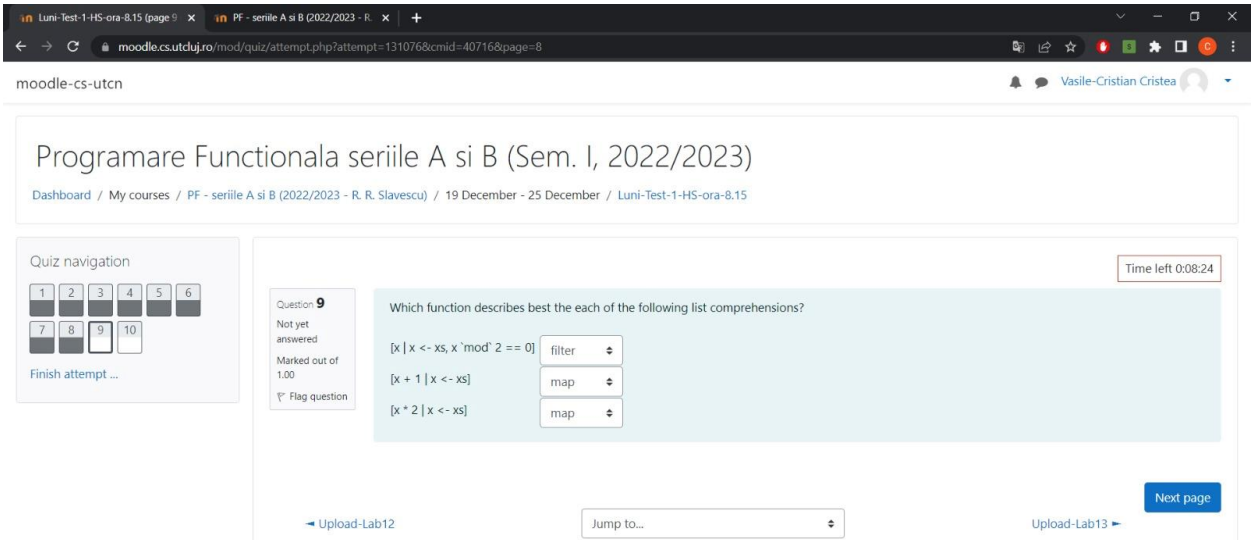


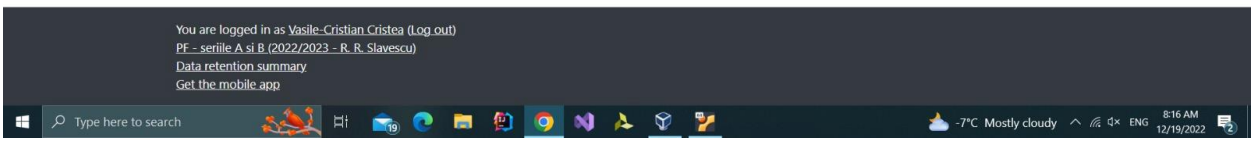

Luni-Test-1-HS-ora-8.15 (page 3
PF - seriile A si B (2022/2023 - R.
moodle.cs.utcluj.ro/mod/quiz/attempt.php?attempt=131076&cmid=40716&page=2
moodle-cs-utcn
Vasile-Cristian Cristea
Programare Functionala seriile A si B (Sem. I, 2022/2023)
Dashboard / My courses / PF - seriile A si B (2022/2023 - R. R. Slavescu) / 19 December - 25 December / Luni-Test-1-HS-ora-8.15
Quiz navigation
1 2 3 4 5 6
7 8 9 10
Finish attempt ...
Time left 0:12:01
Question 3
Not yet answered
Marked out of 1.00
Flag question
Which of the following are examples of not valid ways to create local definitions in Haskell?
a. y * 2 with y = 5
b. let y = 5 in y * 1
c. local y = 5 in y * 2
d. y * 2 where y = 5
Next page
Upload-Lab12
Jump to...
Upload-Lab13
You are logged in as Vasile-Cristian Cristea (Log out)
PF - seriile A si B (2022/2023 - R. R. Slavescu)
Data retention summary
Get the mobile app
Type here to search
-7°C Mostly cloudy
ENG
8:12 AM
12/19/2022

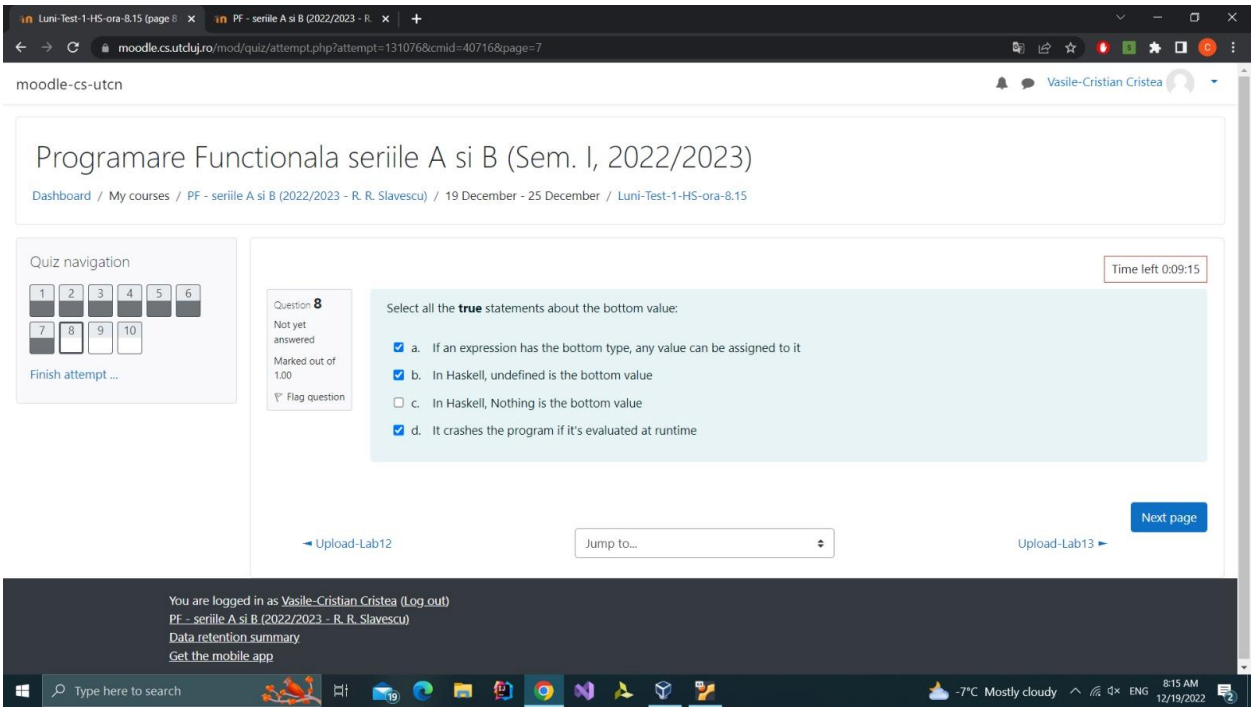

Luni-Test-1-HS-ora-8.15 (page 4
PF - seriile A si B (2022/2023 - R.
moodle.cs.utcluj.ro/mod/quiz/attempt.php?attempt=131076&cmid=40716&page=3
moodle-cs-utcn
Vasile-Cristian Cristea
Programare Functionala seriile A si B (Sem. I, 2022/2023)
Dashboard / My courses / PF - seriile A si B (2022/2023 - R. R. Slavescu) / 19 December - 25 December / Luni-Test-1-HS-ora-8.15
Quiz navigation
1 2 3 4 5 6
7 8 9 10
Finish attempt ...
Time left 0:11:44
Question 4
Not yet answered
Marked out of 1.00
Flag question
Which of the following are examples of valid ways to create local definitions in Haskell?
a. x + 1 where x = 2
b. let x = 2 in x + 1
c. x + 1 with x = 2
d. local x = 2 in x + 1
Next page
Upload-Lab12
Jump to...
Upload-Lab13
You are logged in as Vasile-Cristian Cristea (Log out)
PF - seriile A si B (2022/2023 - R. R. Slavescu)
Data retention summary
Get the mobile app
Type here to search
-7°C Mostly cloudy
ENG
8:13 AM
12/19/2022

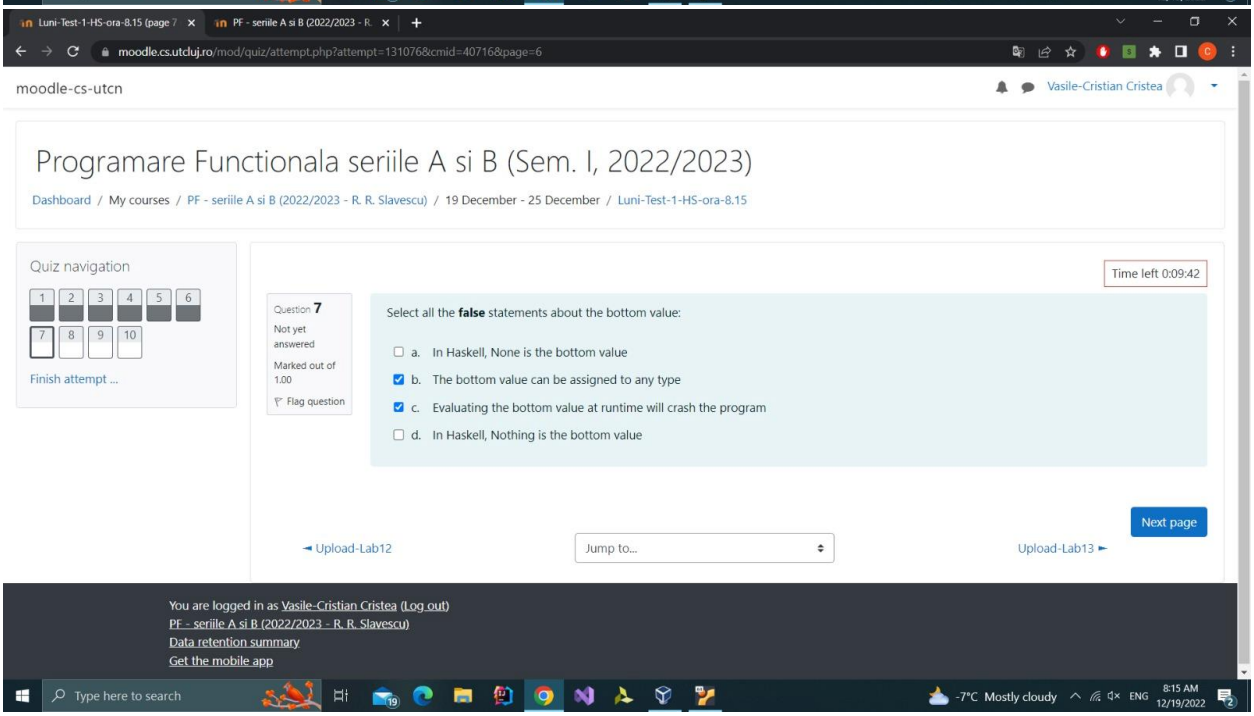

Luni-Test-1-HS-ora-8.15 (page 10
PF - seriile A si B (2022/2023 - R.
moodle.cs.utcluj.ro/mod/quiz/attempt.php?attempt=131076&cmid=40716&page=9
moodle-cs-utcn
Vasile-Cristian Cristea
Programare Functionala seriile A si B (Sem. I, 2022/2023)
Dashboard / My courses / PF - seriile A si B (2022/2023 - R. R. Slavescu) / 19 December - 25 December / Luni-Test-1-HS-ora-8.15
Quiz navigation
1 2 3 4 5 6
7 8 9 10
Finish attempt ...
Time left 0:07:37
Question 10
Not yet answered
Marked out of 1.00
Flag question
Which function describes best the each of the following list comprehensions?
[Char.toUpper x | x <- xs]
[x | x <- xs, x `elem` ['a'..'z']]
[take 2 x | x <- xs]
Choose...
Finish attempt ...
Upload-Lab12
Jump to...
Upload-Lab13

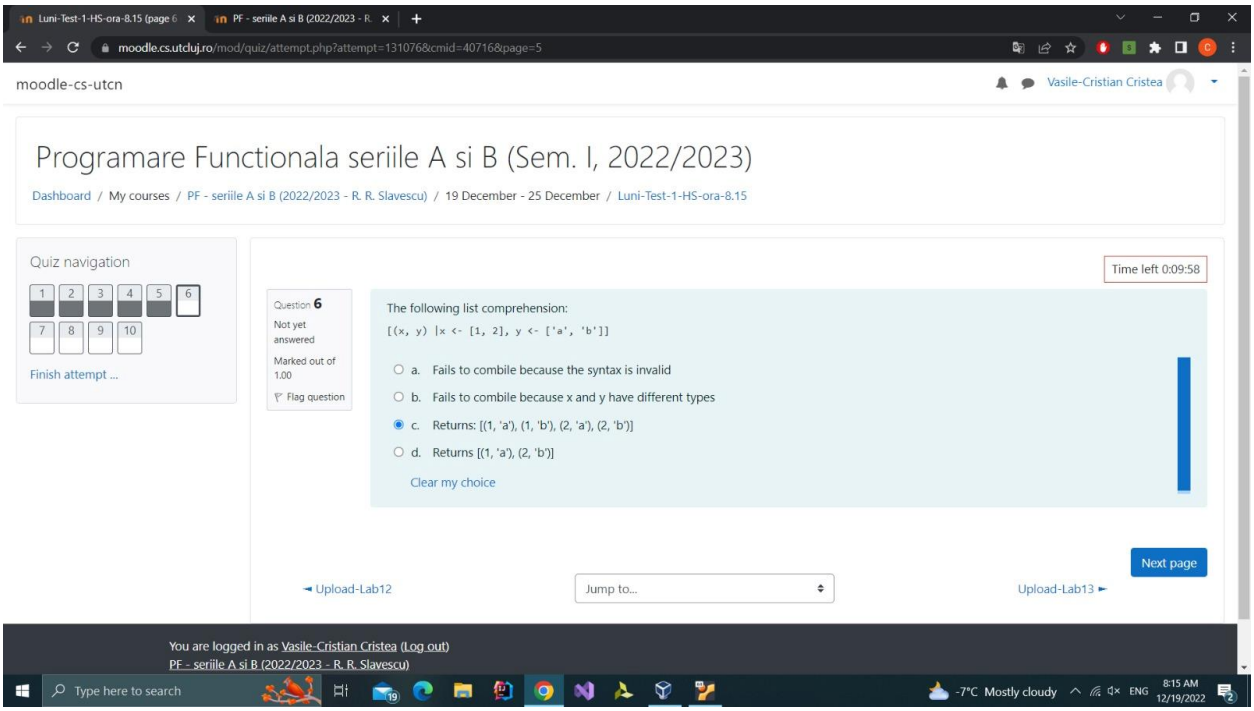

You are logged in as Vasile-Cristian Cristea (Log out)
PF - seriile A si B (2022/2023 - R. R. Slavescu)
Data retention summary
Get the mobile app
Type here to search
-7°C Mostly cloudy
ENG
8:17 AM
12/19/2022


Luni-Test-1-HS-ora-8.15 (page 9
PF - seriile A si B (2022/2023 - R.
moodle.cs.utcluj.ro/mod/quiz/attempt.php?attempt=131076&cmid=40716&page=8
moodle-cs-utcn
Vasile-Cristian Cristea
Programare Functionala seriile A si B (Sem. I, 2022/2023)
Dashboard / My courses / PF - seriile A si B (2022/2023 - R. R. Slavescu) / 19 December - 25 December / Luni-Test-1-HS-ora-8.15
Quiz navigation
1 2 3 4 5 6
7 8 9 10
Finish attempt ...
Time left 0:08:24
Question 9
Not yet answered
Marked out of 1.00
Flag question
Which function describes best the each of the following list comprehensions?
[x | x <- xs, x `mod` 2 == 0]
[x + 1 | x <- xs]
[x * 2 | x <- xs]
filter
map
map
Next page
Upload-Lab12
Jump to...
Upload-Lab13


You are logged in as Vasile-Cristian Cristea (Log out)
PF - seriile A si B (2022/2023 - R. R. Slavescu)
Data retention summary
Get the mobile app
Type here to search
-7°C Mostly cloudy
ENG
8:16 AM
12/19/2022


Luni-Test-1-HS-ora-8.15 (page 8
PF - seriile A si B (2022/2023 - R.
moodle.cs.utcluj.ro/mod/quiz/attempt.php?attempt=131076&cmid=40716&page=7
moodle-cs-utcn
Vasile-Cristian Cristea
Programare Functionala seriile A si B (Sem. I, 2022/2023)
Dashboard / My courses / PF - seriile A si B (2022/2023 - R. R. Slavescu) / 19 December - 25 December / Luni-Test-1-HS-ora-8.15
Quiz navigation
1 2 3 4 5 6 7 8 9 10
Finish attempt ...
Time left 0:09:15
Question 8
Not yet answered
Marked out of 1.00
Flag question
Select all the true statements about the bottom value:
a. If an expression has the bottom type, any value can be assigned to it
b. In Haskell, undefined is the bottom value
c. In Haskell, Nothing is the bottom value
d. It crashes the program if it's evaluated at runtime
Next page
Upload-Lab12
Jump to...
Upload-Lab13
You are logged in as Vasile-Cristian Cristea (Log out)
PF - seriile A si B (2022/2023 - R. R. Slavescu)
Data retention summary
Get the mobile app
Type here to search
-7°C Mostly cloudy
ENG
8:15 AM
12/19/2022


Luni-Test-1-HS-ora-8.15 (page 7
PF - seriile A si B (2022/2023 - R.
moodle.cs.utcluj.ro/mod/quiz/attempt.php?attempt=131076&cmid=40716&page=6
moodle-cs-utcn
Vasile-Cristian Cristea
Programare Functionala seriile A si B (Sem. I, 2022/2023)
Dashboard / My courses / PF - seriile A si B (2022/2023 - R. R. Slavescu) / 19 December - 25 December / Luni-Test-1-HS-ora-8.15
Quiz navigation
1 2 3 4 5 6 7 8 9 10
Finish attempt ...
Time left 0:09:42
Question 7
Not yet answered
Marked out of 1.00
Flag question
Select all the false statements about the bottom value:
a. In Haskell, None is the bottom value
b. The bottom value can be assigned to any type
c. Evaluating the bottom value at runtime will crash the program
d. In Haskell, Nothing is the bottom value
Next page
Upload-Lab12
Jump to...
Upload-Lab13
You are logged in as Vasile-Cristian Cristea (Log out)
PF - seriile A si B (2022/2023 - R. R. Slavescu)
Data retention summary
Get the mobile app
Type here to search
-7°C Mostly cloudy
ENG
8:15 AM
12/19/2022


Luni-Test-1-HS-ora-8.15 (page 6
PF - seriile A si B (2022/2023 - R.
moodle.cs.utcluj.ro/mod/quiz/attempt.php?attempt=131076&cmid=40716&page=5
moodle-cs-utcn
Vasile-Cristian Cristea
Programare Functionala seriile A si B (Sem. I, 2022/2023)
Dashboard / My courses / PF - seriile A si B (2022/2023 - R. R. Slavescu) / 19 December - 25 December / Luni-Test-1-HS-ora-8.15
Quiz navigation
1 2 3 4 5 6
7 8 9 10
Finish attempt ...
Time left 0:09:58
Question 6
Not yet answered
Marked out of 1.00
Flag question
The following list comprehension:
[(x, y) |x <- [1, 2], y <- ['a', 'b']]
a. Fails to combile because the syntax is invalid
b. Fails to combile because x and y have different types
c. Returns: [(1, 'a'), (1, 'b'), (2, 'a'), (2, 'b')]
d. Returns [(1, 'a'), (2, 'b')]
Clear my choice
Next page
Upload-Lab12
Jump to...
Upload-Lab13
You are logged in as Vasile-Cristian Cristea (Log out)
PF - seriile A si B (2022/2023 - R. R. Slavescu)
Type here to search
-7°C Mostly cloudy
ENG
8:15 AM
12/19/2022