

[Dashboard](#) / [My courses](#) / [03-ACS-L-A2-S2-PP-CA-CB-CC](#) / [24 April - 30 April](#) / [Test - Laborator 8](#)**Started on** Tuesday, 2 May 2023, 2:15 PM**State** Finished**Completed on** Tuesday, 2 May 2023, 2:22 PM**Time taken** 7 mins 41 secs**Marks** 1.00/4.00**Grade** 2.50 out of 10.00 (25%)Question **1**

Correct

Mark 1.00 out of 1.00

Fie codul:

```
data Subject = PA | PP | PC | ED | CN
↳ deriving (Show, Eq, Ord)

codeCracker :: Subject -> Subject ->
↳ Subject -> String
codeCracker s1 s2 s3 = if (s1 < s2) && (s3
↳ == s2)
                        then "Success"
                        else "Failure"
```

Care apel al funcției codeCracker va întoarce "Success" ?

Select one:

- ☐ a. `codeCracker PC CN PC`
- ☒ b. `codeCracker PA PP PP` ✓
- ☐ c. Funcția nu va întoarce niciodată "Success"
- ☐ d. `codeCracker (Subject PP) (Subject ED)`
↳ `(Subject ED)`

Răspunsul dumneavoastră este corect.

The correct answer is: `codeCracker PA PP PP`Question **2**

Incorrect

Mark 0.00 out of 1.00

Vrem să definim în Haskell o structură de date similară unui Map din Java. Varianta de a face acest lucru este:

Select one:

- ☒ a. `newtype Map k v = [(k, v)]` ✗

- ☐ b. Nu se poate defini o astfel de structură de date, întrucât nu este permis să avem două variabile de tip.
- ☐ c. `newtype Map k v = Map [(k, v)]`
- ☐ d. `data Map k v = Map (k, v)`

Răspunsul dumneavoastră este incorect.

The correct answer is:

`newtype Map k v = Map [(k, v)]`

Question **3**

Incorrect

Mark 0.00 out of 1.00

Se dau următoarele 2 programe, care vor fi rulate separat.

```
-- program 1
newtype DataSet1 = Zero | One deriving
  ↳ (Show, Eq)
func1 x
  | x == Zero = "zero"
  | x == One  = "one"

-- program 2
data DataSet2 = Two | Three deriving (Show,
  ↳ Eq)
func2 x
  | x == Two  = "two"
  | x == Three = "three"
```

Ce se va afișa la apelurile următoare?

```
func1 Zero
func2 Two
```

Select one:

- ☒ a. `"zero"`
`"two"`
- ☐ b. `"zero"`
`eroare`
- ☐ c. `eroare`
`eroare`
- ☐ d. `eroare`
`"two"`

✗

Răspunsul dumneavoastră este incorect.

eroare
"two"

The correct answer is:

Question **4**

Incorrect

Mark 0.00 out
of 1.00

Sunt arborii obținuți prin următoarele apeluri echivalenți:

```
foldl insertElem BSTNil [7, 4, 12, 2, 3, 1, 10, 15, 8
```

și

```
foldr insertElem2 BSTNil [7, 4, 12, 2, 3, 1, 10, 15,
```

unde:

```
data BST a = BSTNod {
    vl :: a
    , lt :: (BST a)
    , rt :: (BST a)
} | BSTNil deriving Show
```

```
insertElem :: (Ord a, Eq a) => BST a -> a -> BST a
insertElem BSTNil elem = BSTNod elem BSTNil BSTNil
insertElem root@(BSTNod value left right) elem
    | value == elem = root
    | value < elem = BSTNod value left (insertElem right
    ↪ elem)
    | value > elem = BSTNod value (insertElem left elem
    ↪ right)
```

```
-- la fel ca insertElem, dar cu parametrii în ordine
    ↪ inversă
insertElem2 :: (Ord a, Eq a) => a -> BST a -> BST a
insertElem2 elem BSTNil = BSTNod elem BSTNil BSTNil
insertElem2 elem root@(BSTNod value left right)
    | value == elem = root
    | value < elem = BSTNod value left (insertElem2 ele
    ↪ right )
```

```
| value > elem = BSTNod value (insertElem2 elem left  
→ right
```

Select one:

- ☒ a. Da, arborele fiind de căutare, el nu poate avea decât o singură structură. ❌
- ☐ b. Nu, se construiesc doi arbori diferiți.
- ☐ c. Nu, pentru că signatura lui insertElem nu este compatibilă cu foldl și generează o eroare de tip.
- ☐ d. Nu, pentru că signatura lui insertElem2 nu este compatibilă cu foldr și generează o eroare de tip.

Răspunsul dumneavoastră este incorect.

The correct answer is: Nu, se construiesc doi arbori diferiți.

[◀ Test - Laborator 7](#)

Jump to...

[Test - Laborator 9 ▶](#)