

Tipul b trebuie să adere la clasa Show





Răspunsul dumneavoastră este corect.

The correct answer is:

Tipul b trebuie să adere la clasa Show.

Question 4

Correct Mark 1.00 out of 1.00

Dacă dorim să exprimăm faptul că un arbore t1 e mai mic decât un arbore t2 atunci când t1 este mai scund decât t2, ce clasă trebuie instanțiată și ce operator al clasei este suficient să fie implementat?

Select one:



b. Ord
$$si (==)$$



Show
$$\sin (i=)$$

0





Răspunsul dumneavoastră este corect.

The correct answer is:

111

Ord și compare





[a]
fără a garanta că
a
este instanță a clasei
Eq

Question 3

Correct Mark 1.00 out of 1.00

Codul de mai jos dă eroare pentru că:

data Mytype a = Build a a
instance Show (Mytype b) where
 show (Build x y) = show x ++ show y

Select one:

- a. Codul nu dă eroare.
- b. Tipul a nu este același cu tipul b.
- Build nu întoarce un String.
- d. Tipul b trebuie să adere la clasa Show.





Răspunsul dumneavoastră este corect.

The correct answer is:

Tipul b trebuie să adere la clasa Show.

Question 4

Correct Mark 1.00 out of 1.00

SHOW ONE PAGE AT A TIME

Review



Fie următoarea clasă ce definește un concept generalizat de paritate. Definim o listă ca fiind pară sau impară în funcție de lungimea acesteia.

```
class MyParity a where
 myOdd :: a -> Bool
 myEven :: a -> Bool
 myOdd x = not (myEven x)
 myEven x = not (myOdd x)
instance MyParity [a] where
  myEven list = if list == [] then True else not $\$ myEven $\$ tail
  → list
```

Avem erori la compilare? Dacă da, de ce?

Select one:

Da. Nu putem folosi a. (==) pe liste de tip [a] fără a garanta că este instanță a clasei Eq

Nu. b.

Da. Linia C. instance MyParity [a] where ar trebui să fie înlocuită de instance MyParity [] where

Da. Nu este implementat myOdd. 🗙 d.





Răspunsul dumneavoastră este incorect.

The correct answer is:

Da. Nu putem folosi (==) pe liste de tip [a] fără a garanta că









Question 1

Mark 0.00 out of 1.00

Se dă următorul tip:

```
data MyList a = Empty | Node a (MyList a)
```

Se dorește să se adauge acest tip la clasa Eq, astfel încât două elemente să fie considerate egale fie dacă ambele sunt Empty, fie dacă informația dintrun nod este dublul informației din celălalt. De exemplu, (Node 1 _) și (Node 2 _) sunt considerate egale).

Care este varianta corectă?

Select one:

niciunul dintre răspunsuri nu este corect a.

b. instance Eq (MyList a) where

Empty == Empty = True

(Node el1 _) == (Node el2 _) = (el1

+ 2 == el2) || (el2 * 2 == el1)

== _ = False

11 /= 12 = not (11 == 12)

d. instance Num a => Eq (MyList a) where

Empty == Empty = True

(Node el1 _) == (Node el2 _) = (el1

+ 2 == el2) || (el2 * 2 == el1)

== _ = False

X

Răspunsul dumneavoastră este incorect.

The correct answer is:

111





Question 4

Correct Mark 1.00 out of 1.00

Se dă programul:

```
data PaymentStatus = Pending | Active | Cancelled deriving (Eq,
→ Show)
data Payment = PaymentC
   { userId :: String
   , transactionId :: Int
   , status :: PaymentStatus
   } deriving (Eq, Show)
```

Care dintre următoarele variante implementează corect funcția care returnează statusul tranzacției?

Select one:

a. getStatus1 (PaymentC _ _ s) = s

Doar getStatus1:

Toate variantele sunt corecte. b.





Doar getStatus3:

C.

getStatus3 payment@(PaymentC _ _ s) = s



Doar getStatus2: d.

getStatus2 = status



Răspunsul dumneavoastră este corect.

The correct answer is:

Toate variantele sunt corecte.





C.

d. $a \rightarrow T a$





Răspunsul dumneavoastră este incorect.

The correct answer is:





Question 3

Incorrect
Mark 0.00 out of 1.00

Pentru definitia:

data Lst a = Nil | Cons a (Lst a)

, ce tip are expresia

(flip Cons) . Cons 'a'

? (obs: flip se aplică pe o funcție binară și schimbă ordinea în care aceasta își va primi argumentele)

Select one:

a. Lst Char -> Char -> Lst Char



b. Lst Char -> Lst Char



C. Char a => a -> Lst a -> a -> Lst a





d. Char $a \Rightarrow Lst a \Rightarrow a \Rightarrow Lst a$



Răspunsul dumneavoastră este incorect.

The correct answer is:

Lst Char -> Char -> Lst Char



Ouestion 4

Correct







Răspunsul dumneavoastră este incorect.

The correct answer is:

Doar pentru

pathBetween

, deoarece este posibil ca cele două noduri să nu fie conectate.

Question 2

Incorrect Mark 0.00 out of 1.00

Fie tipul de mai jos:

data T a = C { field :: a }

Care este tipul funcției (field . C)?

Select one:

- a. a -> a
- b. T a -> a
- C. T a -> T a
- d. $a \rightarrow T a$

X

Răspunsul dumneavoastră este incorect.

The correct answer is:

a -> a



Question 3

Incorrect Mark 0.00 out of 1.00

Pentru definiția:

data Lst a = Nil | Cons a (Lst a)

, ce tip are expresia

(flip Cons) . Cons 'a'





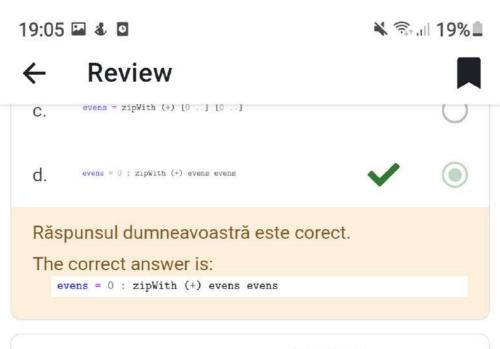
Incorrect

Mark 0.0		ıt of 1.00
Ne dorim să definim două funcții care lucrează cu grafuri. Una dintre ele va returna un drum pe care îl putem parcurge pentru a ajunge dintr-un nod anume în altul; a doua va calcula numărul de componente conexe din graf. În semnătura căror funcții ar fi indicat să folosim tipul		
Maybe		
?		
<pre>pathBetween :: Node -> Node -> Graph -> [Node]</pre>		
<pre>countComponents :: Graph -> Integer</pre>		
Select one:		
	Doar pentru	
a.	pathBetween	
	, deoarece este posibil ca cele două noduri să nu fie conectate.	
b.	Ambele semnături necesită tipul Maybe	
C.	Doar pentru countComponents , deoarece o putem apela pe un graf fără noduri și nu există un întreg potrivit pentru acest input.	0
d.	Nu se justifică utilizarea tipului Maybe pentru niciuna din cele două funcții.	0
Răspunsul dumneavoastră este incorect. The correct answer is:		
Doar pentru		
	pathBetween	
	deoarece este posibil ca cele două noduri să nu fie conectate.	

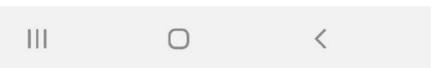
Question 2

111

Incorrect Mark 0.00 out of 1.00



Incorrect Question 4 Mark 0.00 out of 1.00 Funcția f definită ca (cu ord :: Char -> Int) f = zipWith (,) . map ord întoarce: Select one: O listă de perechi între două caractere. a. Funcția f nu are un tip sintetizabil. b. O listă de perechi între un caracte și o C. → valoare oarecare. O listă de perechi între un întreg 🗳 o d. → valoare oarecare. Răspunsul dumneavoastră este incorect. The correct answer is: O listă de perechi între un întreg și → valoare oarecare.







a.

Răspunsul dumneavoastră este incorect.

The correct answer is:

[True, True]



Question 3

Correct

Mark 1.00 out of 1.00

 ${\it Care \ dintre \ urm \~ato} are le \ expresii \ *NU* \ produce \ fluxul \ tuturor \ numerelor \ pare?$

Select one:

a. evens = 0 : map (+ 2) evens

b. evens = [0, 2 ..]

C. evens = zipWith (+) [0 ..] [0 ..]

d. evens = 0 : zipWith (+) evens evens





Răspunsul dumneavoastră este corect.

The correct answer is:

evens = 0 : zipWith (+) evens evens

Question 4

Incorrect

Mark 0.00 out of 1.00

Funcția f definită ca (cu ord :: Char -> Int)

f = zipWith (,) . map ord

111

întoarce:

Select one:

a. O listă de perechi între două caractere.



Răspunsul dumneavoastră este corect.

The correct answer is:

4.5



Question 2

Incorrect

Mark 0.00 out of 1.00

La ce se va evalua următoare expresie Haskell?

Select one:

a. [False, False, True, True]



b. [False, False]



C. [True, True, False, False]





d. [True, True]



Răspunsul dumneavoastră este incorect.

The correct answer is:

[True, True]



Question 3

Correct

Mark 1.00 out of 1.00

Care dintre următoarele expresii *NU* produce fluxul tuturor numerelor pare?

Select one:

a. evens = 0 : map (+ 2) evens







2/4

Grade

5 out of 10 (50%)

Question 1

Correct

Mark 1.00 out of 1.00

Ce produce expresia:

((/) . (+1)) 8 2

Select one:

a. 4.5





b. 5

C.

d.

2.66666666



Răspunsul dumneavoastră este corect.

The correct answer is:

4.5



Question 2

Ш

Incorrect

Mark 0.00 out of 1.00

La ce se va evalua următoare expresie Haskell?

filter id \$ zipWith (<) [1, 2, 3, 4] [4, 3, 2, 1]

SHOW ONE PAGE AT A TIME



(1) True; (2) Eroare; (3) True 🗶 d.





Răspunsul dumneavoastră este incorect.

The correct answer is:

(1) True; (2) False; (3) Eroare

Question 4

Incorrect

Mark 0.00 out of 1.00

Care este rezultatul următoarei expresii?

foldl (\ y x -> (:) (head x) y) [] [[x] | x
$$\leftarrow$$
 <- [1..10], x 'mod' 2 == 0]

Select one:





[[2], [4], [6], [8], [10]] b.



va produce eroare C.



[10,8,6,4,2] d.



Răspunsul dumneavoastră este incorect.

The correct answers are:

va produce eroare



The correct answer is:

```
["a"] ++ "zi"
```

Question 3

Incorrect Mark 0.00 out of 1.00

```
Ce se obţine la evaluarea fiecărei expresii de mai jos?

(1):

expression1 = [1, 2, 3, 4, 5] == foldr (\x acc → x : acc) []

∴ [1, 2, 3, 4, 5]

(2):

expression2 = [1, 2, 3, 4, 5] == foldl (\acc x → x : acc) []

∴ [1, 2, 3, 4, 5]

(3):

expression3 = left == right

where

left = foldr (\x acc → x : acc) [] list

right = reverse  foldl (\x acc → x : acc) []

∴ list

list = [1, 2, 3, 4, 5]
```

Select one:

- a. (1) False; (2) Eroare; (3) True
- b. (1) True; (2) False; (3) True
- c. (1) True; (2) False; (3) Eroare
- d. (1) True; (2) Eroare; (3) True 🗶 🔘

Răspunsul dumneavoastră este incorect.

The correct answer is:

(1) True; (2) False; (3) Eroare

Incorrect







Răspunsul dumneavoastră este incorect.

The correct answer is:

Question 2

Incorrect

Mark 0.00 out of 1.00

Care din următoarele expresii generează eroare?

Select one:











Răspunsul dumneavoastră este incorect.

The correct answer is:

Question 3

Incorrect

Mark 0.00 out of 1.00

Ce se obține la evaluarea fiecărei expresii de mai jos?
(1):

expression1 = [1, 2, 3, 4, 5] == foldr (\x acc -> x : acc) []
$$\rightarrow$$
 [1, 2, 3, 4, 5]

(2):

(3):

Ш



0 out of 10 (0%)

Question 1

Incorrect

Mark 0.00 out of 1.00

Care din următoarele expresii nu e echivalentă cu celelalte?

Select one:

a. 0 == sum (filter (==2) xs)



b. notElem 2 xs



C. null [1 | 2<-xs]





d. null [x==2 | x<- xs]



Răspunsul dumneavoastră este incorect.

The correct answer is:

Question 2

Incorrect

Mark 0.00 out of 1.00

Care din următoarele expresii generează eroare?

Select one:

a. ["a"] ++ "zi"

111



b. 'a':"zi"



Răspunsul dumneavoastră este incorect.

The correct answer is:

$$(\lambda a. \lambda b. (* a b) 3 (\lambda a. (- a 1) 3))$$

 $(\lambda a. \lambda b. (* a b) 3 2)$
 $(* 3 2)$

Question 4

Correct Mark 1.00 out of 1.00

Care NU este o metodă de a construi fluxul numerelor naturale?

Select one:

a. (define naturals (stream-cons 0 naturals))





 \bigcirc

 \bigcirc

 \bigcirc

Răspunsul dumneavoastră este corect.

The correct answer is:

(define naturals (stream-cons 0 naturals))



Question 3

Incorrect Mark 0.00 out of 1.00

O modalitate de evaluare specifică Racket aplicată pe funcțiile următoare este:

$$(\lambda \quad a.\lambda \quad b.(* \quad a \quad b) \quad (\lambda \quad a.\lambda \quad b.3 \quad 1 \quad 1)$$

 $\rightarrow \quad (\lambda \quad a.(- \quad a \quad 1) \quad 3))$

Select one:

a.
$$(\lambda a. \lambda b. (* a b) 3 (\lambda a. (- a 1) 3))$$

 $(\lambda a. \lambda b. (* a b) 3 2)$
 $(* 3 2)$

 \bigcirc

C. nu se poate evalua, parametrii a si b din
$$\rightarrow$$
 (λ a. λ b.(3) 1 1) nu sunt folosiți

×



d.
$$(\lambda a. \lambda b. (* a b) 3 (\lambda a. (- a 1) 3))$$

 $(\lambda b. (* 3 b) (\lambda a. (- a 1) 3))$
 $(\lambda b. (* 3 b) 2)$
 $(* 3 2)$

 \subset

Răspunsul dumneavoastră este incorect.

The correct answer is:

$$(\lambda a. \lambda b. (* a b) 3 (\lambda a. (- a 1) 3))$$

 $(\lambda a. \lambda b. (* a b) 3 2)$
 $(* 3 2)$

Question 4

III

Correct Mark 1.00 out of 1.00





d. o procedură

 \mathbb{C}

Răspunsul dumneavoastră este incorect.

The correct answer is:

o promisiune

Question 2

Correct

Mark 1.00 out of 1.00

Ce va returna următorul program?

Select one:

a. 10000



b. Eroare



C. 2 ^ 10000 - 1





d. 2 ^ 10001 - 1



Răspunsul dumneavoastră este corect.

The correct answer is:

III





Question 1

Incorrect
Mark 0.00 out of 1.00

Ce rezultat produce codul următor?

```
(define f(\lambda) () (delay f(\lambda)))

(define g(f(\lambda)))
```

Select one:

a. o listă care conține o promisiune





b. o promisiune



C. eroare



d. o procedură



Răspunsul dumneavoastră este incorect.

The correct answer is:

o promisiune

Question 2

Correct

Mark 1.00 out of 1.00

Ce va returna următorul program?

```
(define (stream-gen n1 n2 n3)
  (stream-cons
  (* n1 (expt n2 n3))
   (stream-gen n1 n2 (add1 n3))))

(fold1 + 0 (stream-take (stream-gen 1 2 0))
```





Răspunsul dumneavoastră este incorect.

The correct answer is:

Question 4

Incorrect

Mark 0.00 out of 1.00

Ce va afișa secvența Racket de mai jos?

Select one:

a. 11

b. Eroare

- X

c. 10

 \mathbb{C}

d. 12

Răspunsul dumneavoastră este incorect.

The correct answer is:





The correct answer is:

88

Question 3

Incorrect

Mark 0.00 out of 1.00

Care este rezultatul rulării următorului cod?

(magic '(1 1 4 7 7 7 10) '())

Select one:

a. '(7 7 1)

- X

b. '(10 7 7 1)

c. '(7 1)

C

d. '(1 7)

 \mathbb{C}

Răspunsul dumneavoastră este incorect.

The correct answer is:

(10 7 7 1)

Question 4

111

Incorrect

Mark 0.00 out of 1.00



The correct answer is:

2

Question 2

Correct

Mark 1.00 out of 1.00

Ce va afișa următorul program Racket?

Select one:

a. 88





b. 6



c. 47



d Programul intră în buclă infinită.



Răspunsul dumneavoastră este corect.

The correct answer is:

88

Question 3

Incorrect

Mark 0.00 out of 1.00

Care este rezultatul rulării următorului cod?

(define (magic 1 n)





Question 1

Correct

Mark 1.00 out of 1.00

Ce rezultat produce codul următor?

Select one:

a. 2





b. 1



c. 11



d. 12



Răspunsul dumneavoastră este corect.

The correct answer is:

2

Question 2

Correct

Mark 1.00 out of 1.00

Ce va afișa următorul program Racket?

Select one:

Review \leftarrow



Question 4

Mark 0.00 out of 1.00

Funcția tabulate adaugă un indice, pornind de la 0, elementelor unei liste L. Elementele noii liste contin perechi indice - valoare. Care este varianta care NU implementează corect tabulate? Funcția range este implementată în Racket și apelul

(range x)

va întoarce

Select one:

```
(define (tabulate L)
a.
         (apply map cons (append (list (range
         → (length L))) (list L))))
       (tabulate L)
```

```
(define (tabulate L)
b.
          (map cons (range (length L)) L))
        (tabulate L)
```

```
(define (tabulate idx L)
C.
          (if (null? L)
             null
             (cons (cons idx (car L)) (tabulate (+
             → idx 1) (cdr L)))))
        (tabulate 0 L)
```





(define (tabulate L) d. (foldr cons 0 L)) (tabulate L)

Răspunsul dumneavoastră este incorect.

The correct answer is:

```
(define (tabulate L)
  (foldr cons 0 L))
(tabulate L)
```



Question 2

Incorrect

Mark 0.00 out of 1.00

Ce se va afișa la rularea următoarei secvențe de cod?

Select one:

a. '(1 5 9 2 6 10 3 7 11 4 8 12)

- b. '((1 5 9) (2 6 10) (3 7 11) (4 8 12))



C. '(1 2 3 4 5 6 7 8 9 10 11 12)



d. '((1 2 3 4) (5 6 7 8) (9 10 11 12))





Răspunsul dumneavoastră este incorect.

The correct answer is:

'(1 2 3 4 5 6 7 8 9 10 11 12)

Question 3

Correct

Mark 1.00 out of 1.00

Care este varianta ce reprezintă echivalentul expresiei de mai jos?

Select one:

a. (append '(1 2 3) '(4 5 6))





b. (cons '(1 2 3) '(4 5 6))









Question 1

Incorrect
Mark 0.00 out of 1.00

Care dintre următoarele implementări reprezintă însumarea tuturor elementelor dintr-o matrice $N \times N$ (exemplu: '((1 2

1)

2)

3)

a. 1), 3)



b. toate



c. 2), 3)





d. 1), 2)



Răspunsul dumneavoastră este incorect.

The correct answer is:

toate



penura a preructa rezultatur.

Question 4

Incorrect Mark 0.00 out of 1.00

Ce complexitate spațială (spațiu suplimentar pe stivă, nu cel ocupat de lista de intrare) are funcția inc definită mai jos?

Notăm cu n numărul de elemente dintr-o listă oare-care ${\bf L}.$

Select one:

a. $\Theta(n)$

 \bigcirc

b. $\Theta(1)$

c. $\Theta(n^2)$

X

d. $\Theta(n^3)$

Răspunsul dumneavoastră este incorect.

The correct answer is:

 $\Theta(n)$





Question 3

Correct

Mark 1.00 out of 1.00

Care este afirmația adevărată?

```
(define (f L acc)
  (if (null? L)
    acc
    (+ 1 (f (cdr L) (+ 1 acc)))))
```

Select one:

- a. Funcția f este recursivă pe stivă deoarece necesită întoarcerea în apelul recursiv părinte pentru a prelucra rezultatul.
- b. Funcția f este recursivă pe stivă deoarece folosește un acumulator.
- \bigcirc
- c. Compilatorul poate aplica tail-call optimization pentru a reduce spațiul de memorie necesar apelului cu succes al funcției
- 0
- d. Funcția f este recursivă pe coadă deoarece foloseste un acumulator

 \bigcirc

Răspunsul dumneavoastră este corect.

The correct answer is:

Funcția f este recursivă pe stivă deoarece necesită întoarcerea în apelul recursiv părinte pentru a prelucra rezultatul.

Question 4

Ш

Incorrect
Mark 0.00 out of 1.00



sumat de informația depusă pe stivă, necesară pentru revenirea din recursivitate

Question 2

Correct

Mark 1.00 out of 1.00

Fie următoarea functie recursivă:

```
(define (func x y acc)
  (if (= y 0)
        acc
        (func x (- y 1) (+ x acc))))
```

și funcția calc, definită pe 2 numere naturale x și y:

```
(define (calc x y) (func x y 0))
```

Ce putem spune despre funcția calc?

Select one:

folosește o funcție recursivă pe coadă pentru a calcula x * y





b. $\begin{array}{c} \text{folosește o funcție recursivă pe stivă pentru a calcula} \\ \text{x * y} \end{array}$

 \bigcirc

C. folosește o funcție recursivă pe coadă pentru a calcula x + y

0

d. $\begin{array}{c} \text{folosește o funcție recursivă pe coadă pentru a calcula} \\ \text{x} + \text{x} * \text{y} \end{array}$

Răspunsul dumneavoastră este corect.

The correct answer is:

folosește o funcție recursivă pe coadă pentru a calcula x * y

Question 3

Correct

Mark 1.00 out of 1.00









Question 1

Correct Mark 1.00 out of 1.00

De ce recursivitatea arborescentă este ineficientă? Select one:

Din cauza timpului consumat de numărul mare de apeluri recursive lansate și din cauza spațiului consumat de informația depusă pe stivă, necesară pentru revenirea din recursivitate



- b. Pentru că se folosește memoizarea apelurilor recursive
- 0
- c. Deoarece are adesea complexitate spațială exponențială
- 0
- d. Deoarece avem apeluri recursive unice, fapt care duce la stocarea multor apeluri pe stivă

\bigcirc

Răspunsul dumneavoastră este corect.

The correct answer is:

Din cauza timpului consumat de numărul mare de apeluri recursive lansate și din cauza spațiului consumat de informația depusă pe stivă, necesară pentru revenirea din recursivitate

Question 2

Correct

Mark 1.00 out of 1.00

Fie următoarea funcție recursivă:

(define (func x y acc)
 (if (= y 0)

SHOW ONE PAGE AT A TIME



Răspunsul dumneavoastră este incorect.

The correct answer is:

Nu genereaza un rezultat, ci eroare.

Question 4

Correct

Mark 1.00 out of 1.00

La ce se evaluează ultima expresie din secvența de mai jos?

Select one:

a. '(2 . 1)

b. '((2) . 1)

c. eroare

d. '(2 1)

Răspunsul dumneavoastră este corect.

The correct answer is:

'((2) . 1)



The correct answer is:

eroare

Question 3

Incorrect
Mark 0.00 out of 1.00

Ce rezultat este generat la rularea secvenței de cod de mai jos?

Select one:

a. Nu genereaza un rezultat, ci eroare.



b. '(3 (2 (1)))



c. '(1 2 3)



d. '(3 2 1)





Răspunsul dumneavoastră este incorect.

The correct answer is:

Nu genereaza un rezultat, ci eroare.

Question 4

Correct

Mark 1.00 out of 1.00

La ce se evaluează ultima expresie din



'(2 0 1)

Question 2

Incorrect

Mark 0.00 out of 1.00

Ce se va afișa dacă rulăm următoarea bucată de cod:

Select one:

a. 'm3r3

b. 'ar3

c. eroare

0

d. 'ana

0

Răspunsul dumneavoastră este incorect.

The correct answer is:

eroare

Question 3

Incorrect

Mark 0.00 out of 1.00

Ce rezultat este generat la rularea secvenței





Question 1

Incorrect

Mark 0.00 out of 1.00

Ce va afișa următorul cod?

Select one:

a '(2 1 0)

0

b. '(1 0 2)

c. '(2 0 1)

0

d. '(0 1 2)

X



Răspunsul dumneavoastră este incorect.

The correct answer is:

Question 2

Incorrect

Mark 0.00 out of 1.00

Ce se va afișa dacă rulăm următoarea bucată de cod:

16 / 1/4 0 0) 1/4 5 0)



