

← Analizare



Nota obținuta
5 din 10 (50%)

Întrebare 1

Incorect
Marcat 0,00 din 1,00

Ce va întoarce următorul apel?

```
(define f (lambda (x) (first x)))
```

```
(define g (lambda (x) (cdddr x)))
```

```
(f (g '(1 (+ 3 4) 2 (- 1 2) 1)))
```

Selectați răspunsul corect:

- a. `'(+ 3 4)` ☐
- b. `'(- 1 2)` ☐
- c. `-1` ☒ ×
- d. `7` ☐

Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

```
'(- 1 2)
```

← Analizare



Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

'(- 1 2)

Întrebare 2

Corect

Marcat 1,00 din 1,00

Ce va afișa următorul cod?

```
(list (equal? (cons 0 null) '(0))  
      (eq?      (cons 0 null) '(0)))
```

Selectați răspunsul corect:

a. '(#f #f)

☐

b. '(#f #t)

☐

c. '(#t #t)

☐

d. '(#t #f)

☒

Răspunsul dumneavoastră este corect.

Răspunsul corect este:

'(#t #f)

Întrebare 3

Corect

Marcat 1,00 din 1,00

Ce va afișa următorul cod?

```
(define (f1 nr)
```

← Analizare



Răspunsul corect este:

'(#t #f)

Întrebare 3

Corect

Marcat 1,00 din 1,00

Ce va afișa următorul cod?

```
(define (f1 nr)
  (if (= nr 0)
      0
      (f2 (- nr 1))))
(define (f2 nr)
  (if (= nr 0)
      1
      (f1 (- nr 1))))
(f1 4)
(f2 4)
```

Selectați răspunsul corect:

a. 0 0

☐

b. 1 1

☐

c. 0 1

☒

d. 1 0

☐

← Analizare



Răspunsul dumneavoastră este corect.

Răspunsul corect este: 0 1

Întrebare 4

Inc corect

Marcat 0,00 din 1,00

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

```
(define (reverse l)
  (append (reverse (cdr l)) (list
    → (car l))))
(reverse '(1 2 3))
```

Selectați răspunsul corect:

- a. Nu generează un rezultat, ci eroare. ☐
- b. '(1 2 3) ☐
- c. '(3 (2 (1))) ☐
- d. '(3 2 1) ☒ ✗

Răspunsul dumneavoastră este inc corect.

Răspunsul corect este:

Nu generează un rezultat, ci eroare.

← Analizare



Întrebare 1

Incorect

Marcat 0,00 din 1,00

Ce tip de recursivitate folosește funcția f și ce rezultat întoarce?

```
(define (f n)
  (cond ((zero? n) 2)
        ((= n 1) 1)
        (else (+ (f (- n 1)) (f (- n
→ 2))))))
```

(f 4)

Selectați răspunsul corect:

- a. recursivitate arborescentă și 7 ☐
- b. recursivitate pe stivă și 7 ☐
- c. recursivitate pe coadă și 4 ☐
- d. recursivitate arborescentă și 4 ☒

Răspunsul dumneavoastră este incorrect.

Răspunsul corect este:

recursivitate arborescentă și 7

← Analizare



Întrebare 2

Incorect

Marcat 0,00 din 1,00

Fie următoarea funcție recursivă:

```
(define (func L)
  (if (<= (length L) 3)
      L
      (cons (car L) (func (cdr L)))))
```

Care este rezultatul următorului apel?

```
(func '(1 2 3 4 5 6))
```

Selectați răspunsul corect:

- a. `'(1 2 3 4 5 6)` ☐
- b. `'(1 2 3)` ☐
- c. `'(4 5 6)` ☒ ✗
- d. `'(3 2 1 4 5 6)` ☐

Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

```
'(1 2 3 4 5 6)
```

Întrebare 3

Corect

Marcat 1,00 din 1,00

Ce spațiu auxiliar pe stivă (complexitate spațială) necesită apelurile funcțiilor f și g?

← Analizare



Întrebare 3

Corect


Marcat 1,00 din 1,00

Ce spațiu auxiliar pe stivă (complexitate spațială) necesită apelurile funcțiilor f și g ?

```
(define (f L)
  (if (null? L)
      1
      (* (car L) (f (cdr L)))))

(define (g L acc)
  (if (null? L)
      acc
      (g (cdr L) (* (car L) acc))))
```

Selectați răspunsul corect:

- a. $O(n)$ pentru f și $O(1)$ pentru g  ☒
- b. $O(n)$ pentru f și $O(n)$ pentru g ☐
- c. $O(1)$ pentru f și $O(1)$ pentru g ☐
- d. $O(1)$ pentru f și $O(n)$ pentru g ☐

Răspunsul dumneavoastră este corect.

Răspunsul corect este:

$O(n)$ pentru f și $O(1)$ pentru g

Corect

← Analizare



Întrebare 4

Corect


Marcat 1,00 din 1,00

Ce complexitate temporală și ce complexitate spațială (spațiu suplimentar pe stivă, nu cel ocupat de lista de intrare) are această funcție?

```
(define (is-member l e)
  (cond
    ((null? l) #f)
    ((equal? e (car l)) #t)
    (else (is-member (cdr l) e))))
```

Notă: n reprezintă numărul de elemente din lista l

Selectați răspunsul corect:

- a. Complexitate temporală: $O(n)$ Complexitate spațială: $O(1)$  ☒
- b. Complexitate temporală: $O(n)$ Complexitate spațială: $O(n)$ ☐
- c. Complexitate temporală: $O(1)$ Complexitate spațială: $O(n)$ ☐
- d. Complexitate temporală: $O(1)$ Complexitate spațială: $O(1)$ ☐

Răspunsul dumneavoastră este corect.

Răspunsul corect este:

Complexitate temporală: $O(n)$ Complexitate spațială: $O(1)$

← Analizare



Puncte

3/4

Notă obținută

7,5 din 10 (75%)

Întrebare 1

Corect

Marcat 1,00 din 1,00

Ce întoarce apelul (map list '(1 2 3) '(4 5 6))?

Selectați răspunsul corect:

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

☐

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

☒

c. ((1 2 3) 4 5 6)

☐

d. ((1 2 3) (4 5 6))

☐

Răspunsul dumneavoastră este corect.

Răspunsul corect este:

((1 4) (2 5) (3 6))

Întrebare 2

Inc corect

Marcat 0,00 din 1,00

AFIȘAȚI CÂTE O PAGINĂ PE RÂND

← Analizare



Răspunsul dumneavoastră este corect.

Răspunsul corect este:

((1 4) (2 5) (3 6))

Întrebare 2

Incorect

Marcat 0,00 din 1,00

Fie funcția `string-append` care face concatenarea între oricâte string-uri (de exemplu (`string-append "aa" "bb" "cc"`) va returna `"aabbcc"`). Ce va afișa următorul cod?

```
(foldr string-append "" '("aa" "ba" "ca")  
  ↪ '("ab" "bb" "cb") '("ac" "bc" "cc"))
```

Selectați răspunsul corect:

- a. `"ccbcaccbbbabcabaaa"` ☐
- b. `"aabacaabbbcbacbcc"` ☒
- c. `"aaabacbabbbccacbcc"` ☐
- d. `"cacbccbabbbcaabac"` ☐

Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

`"aaabacbabbbccacbcc"`

Întrebare 3

Corect

Marcat 1,00 din 1,00

Care dintre următoarele exemple care uti-

← Analizare



"aaabacbabbccacbcc"

Întrebare 3

Corect

Marcat 1,00 din 1,00

Care dintre următoarele exemple care utilizează funcționale NU implementează corect ridicarea la pătrat a elementelor unei liste L?

```
(map (lambda (x) (* x x)) L)
(foldr (lambda (x acc) (cons (* x x)
→ acc)) null L)
(reverse (foldl (lambda (x acc)
→ (cons (* x x) acc)) null L))
```

- a. exemplul cu map ☐
- b. exemplul cu foldr ☐
- c. exemplul cu foldl ☐
- d. toate implementează corect operația ✓ ☒

Răspunsul dumneavoastră este corect.

Răspunsul corect este:

toate implementează corect operația

Întrebare 4

Corect

Marcat 1,00 din 1,00

← Analizare



d. toate implementează corect operația



Răspunsul dumneavoastră este corect.

Răspunsul corect este:

toate implementează corect operația

Întrebare 4

Corect

Marcat 1,00 din 1,00

Ce se va afișa la rularea următoarei secvențe de cod?

```
(foldr (λ (x acc) (cons (add1 x) acc)) '()
  ↪ (list 1 2 3 4))
```

Selectați răspunsul corect:

a. '(2 3 4 5)



b. '(5 4 3 2)



c. 2



d. 5



Răspunsul dumneavoastră este corect.

Răspunsul corect este:

'(2 3 4 5)

← Analizare



```
(define (alternate-list L)
  (letrec
    ((sum-list
      (lambda (L)
        (if (null? L)
            0
            (+ (diff-list (cdr L))
               → (car L))))))
     (diff-list
      (lambda (L)
        (if (null? L)
            0
            (- (sum-list (cdr L))
               → (car L)))))
    (sum-list L)))

(alternate-list '(1 2 3 4 5))
```

Selectați răspunsul corect:

- a. Eroare - diff-list nedefinit în momentul referirii ☐
- b. -1 ✗ ☒
- c. 3 ☐
- d. 15 ☐

Răspunsul dumneavoastră este incorrect.

Răspunsul corect este: 3

← Analizare



Răspunsul corect este:

3

Întrebare 2

Corect

Marcat 1,00 din 1,00

Ce va afișa următoarea secvență de cod?

```
(define (functie x y)
  (let [(+ (lambda(x y) (* x y)))]
    (display (+ x y))
    (display (+ x y)))
(functie 2 4)
(display (+ 2 3))
```

Selectați răspunsul corect:

a. 665

☐

b. 866

☐

c. 865

☒

d. 886

☐

Răspunsul dumneavoastră este corect.

Răspunsul corect este:

865

← Analizare



Răspunsul corect este: 000

Întrebare 3

Corect

Marcat 1,00 din 1,00

Ce va afisa urmatorul cod racket?

```
(define (max-three-numbers x y z)
  (let* ((a x) (b y) (c a))
    (+ a 5)
    (max a b c)))
```

```
(max-three-numbers 2 6 4)
```

Selectați răspunsul corect:

- a. 9 ☐
- b. 6 ☒
- c. 11 ☐
- d. 7 ☐

Răspunsul dumneavoastră este corect.

Răspunsul corect este:

6

Întrebare 4

Inc corect

Marcat 0,00 din 1,00

Care este rezultatul rularii următorului cod?

```
(define f
  (lambda (x) (g x)))
```

```
(define h
  (lambda (x) (sub1 (g x))))
```


← Analizare



Intrebare 4

Marcat 0,00 din 1,00

Care este rezultatul rulării următorului cod?

```
(define f
  (lambda (x) (g x)))

(define h
  (lambda (x) (sub1 (g x))))


(define g
  (let ((f add1)) f))

(f 1)

(g 1)

(h 1)
```

Selectați răspunsul corect:

- a. eroare la
(f 1) ☐
- b. codul rulează la infinit  ☒
- c. #<procedure:f>
2
1 ☐
- d. 2
2
1 ☐

Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

2
2
1

← Analizare



1/4

Notă obținută

2,5 din 10 (25%)

Întrebare 1

Incorect

Marcat 0,00 din 1,00

Definim funcția inc:

```
(define (inc x)
  (λ () (+ x 1)))
```

Care dintre următoarele expresii întoarce 1?

Selectați răspunsul corect:

- a. `(inc 0)` ☐
- b. `(force (inc 0))` ☒
- c. `inc (0)` ☐
- d. `((inc 0))` ☐

Răspunsul dumneavoastră este incorrect.

Răspunsul corect este:

`((inc 0))`

Întrebare 2

Incorect

Marcat 0,00 din 1,00

Care sunt primele 5 elemente ale următorului flux?

← Analizare



Răspunsul corect este:
((inc 0))

Întrebare 2

Incorect

Marcat 0,00 din 1,00

Care sunt primele 5 elemente ale următorului flux?

```
(define stream
  (stream-cons 1
    (stream-map (λ
      ↪ (x) (* -1 x))
    ↪ stream)))
```

Selectați răspunsul corect:

a. -1 1 -1 1 -1



b. 1 -1 1 -1 1



c. 1 1 1 1 1



d. 1 -1 -1 -1 -1



Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

1 -1 1 -1 1

Întrebare 3

Corect

Marcat 1,00 din 1,00

Fie următoarele 2 fluxuri:

```
(define my-stream-1
  (let loop ((n 5))
```

← Analizare



Întrebare 3

Corect

Marcat 1,00 din 1,00


Fie următoarele 2 fluxuri:

```
(define my-stream-1
  (let loop ((n 5))
    (stream-cons n (loop (* n n)))))
```

```
(define my-stream-2
  (stream-cons 5 (stream-zip-with *
    ↪ my-stream-2 my-stream-2)))
```

Care dintre următoarele afirmații este adevărată?

Selectați răspunsul corect:

- a. my-stream-1 și my-stream-2 nu sunt
↪ echivalente, my-stream-1 este fluxul
↪ puterilor lui 5, iar my-stream-2 este
↪ fluxul (5 25 625) ☐
- b. my-stream-1 și my-stream-2 sunt
↪ echivalente, dar my-stream-1 este
↪ definit explicit și my-stream-2 este  ☒
↪ definit implicit.
- c. my-stream-2 va genera o eroare, deoarece
↪ my-stream-2 nu este implementat la
↪ momentul apelării sale în cadrul
↪ stream-zip-with. ☐
- d. my-stream-1 și my-stream-2 sunt
↪ echivalente, dar my-stream-1 este
↪ definit implicit și my-stream-2 este
↪ definit explicit. ☐

Răspunsul dumneavoastră este corect.

Răspunsul corect este:

my-stream-1 și my-stream-2 sunt
↪ echivalente, dar my-stream-1 este

← Analizare



my-stream-1 și my-stream-2 sunt
→ echivalente, dar my-stream-1 este
→ definit explicit și my-stream-2 este
→ definit implicit.

Întrebare 4

Incorect

Marcat 0,00 din 1,00

Ce afișează (prin display și valoare de retur) codul Racket următor?

```
(define x (delay ((lambda (x) (  
  display x) (+ x x)) 5)))  
(define y (delay ((lambda (x) (  
  display x) (+ x x)) 5)))  
(define z x)  
(* (force x) (force y) (force z))
```

Selectați răspunsul corect:

- a. 5102000 ☐
- b. 5551000  ☒
- c. 551000 ☐
- d. 51000 ☐

Răspunsul dumneavoastră este incorrect.

Răspunsul corect este:

551000

← Analizare



Notă obținută

0 din 10 (0%)

Întrebare 1

Incorect

Marcat 0,00 din 1,00

Ce rezultate se obțin la apelurile calc 4, calc 5, calc 6 (în această ordine)?

```
calc x
| x < 10 = calc (x + 2)
| x < 5 = calc (x + 3)
| x == 10 = 1
| x == 11 = 0
| otherwise = 2
```

Selectați răspunsul corect:

- a. 1 2 1 ☐
- b. 2 1 2 ☒
- c. 0 1 0 ☐
- d. 1 0 1 ☐

Răspunsul dumneavoastră este incorect.

Răspunsul corect este: 1 0 1

Întrebare 2

Incorect

Marcat 0,00 din 1,00

AFIȘAȚI CÂTE O PAGINĂ PE RÂND

← Analizare



u. 1 0 1

Răspunsul dumneavoastră este incorect.

Răspunsul corect este: 1 0 1

Întrebare 2

Incorect

Marcat 0,00 din 1,00

Fie funcția matematică $f(x) = (x + 2)/3$. Cum s-ar putea implementa această funcție în Haskell?

Selectați răspunsul corect:

a. `f = (2 +) $ (3 /)`

☐

b. `f = (+ 2) . (/ 3)`

☒

c. `f = (/ 3) . (2 +)`

☐

d. `f = (2 +) . (3 /)`

☐

Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

`f = (/ 3) . (2 +)`

Întrebare 3

Incorect

Marcat 0,00 din 1,00

Se dau următoarele funcții:

`g1 l = head l`

`g2 [x,y,z] = x`

`g3 [x,_] = x`

`g4 (x : xs) = x`

← Analizare



`f = (/ 3) . (2 +)`

Întrebare 3

Incorect

Marcat 0,00 din 1,00

Se dau următoarele funcții:

```
g1 l = head l
g2 [x,y,z] = x
g3 [x,_] = x
g4 (x : xs) = x
```

Care afirmație este falsă?

Selectați răspunsul corect:

- a. `g1` și `g4` fac același lucru ☐
- b. `apelul` `g2 [1,2,2,3]` dă eroare ✗ ☒
- c. `în` `g3, _` face match pe tail de lista
↪ primită ca parametru ☐
- d. `toate` funcțiile vor da eroare dacă primesc
↪ ca parametru lista vidă ☐

Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

```
în g3, _ face match pe tail de lista
↪ primită ca parametru
```

Întrebare 4

Incorect

Marcat 0,00 din 1,00

Care va fi rezultatul evaluării expresiilor de mai jos?

```
my fun :: (Num a) => [a] -> a
```



← Analizare



↪ primită ca parametru

Întrebare 4

Incorrect

Marcat 0,00 din 1,00

Care va fi rezultatul evaluării expresiilor de mai jos?

```
my_fun :: (Num a) => [a] -> a
my_fun (x:xs) = x + my_fun2 xs

my_fun2 :: (Num a) => [a] -> a
my_fun2 (x:xs) = my_fun xs
my_fun2 [] = 0
```

```
my_fun [1,2,3,4,5]
my_fun [1,2,3,4]
```

Selectați răspunsul corect:

- a. 9 și 4 ☐
- b. 9 și excepție ☐
- c. 15 și 10 ☒
- d. excepție în ambele cazuri ☐

Răspunsul dumneavoastră este incorrect.

Răspunsul corect este: 9 și excepție

← Analizare



Puncte

3/4

Notă obținută

7,5 din 10 (75%)

Întrebare 1

Corect

Marcat 1,00 din 1,00

Care este rezultatul următoarei expresii?

`take 5 $ zip (iterate (+1) 0) (repeat 'a')`

Selectați răspunsul corect:

- a. `[(4, 'e'), (3, 'd'), (2, 'c'), (1, 'b'), (0, 'a')]` ☐
- b. `[(4, 'a'), (3, 'a'), (2, 'a'), (1, 'a'), (0, 'a')]` ☐
- c. `[(0, 'a'), (1, 'b'), (2, 'c'), (3, 'd'), (4, 'e')]` ☐
- d. `[(0, 'a'), (1, 'a'), (2, 'a'), (3, 'a'), (4, 'a')]` ☒

Răspunsul dumneavoastră este corect.

Răspunsul corect este:

`[(0, 'a'), (1, 'a'), (2, 'a'), (3, 'a'), (4, 'a')]`

Întrebare 2

Corect

Marcat 1,00 din 1,00

AFIȘAȚI CÂTE O PAGINĂ PE RÂND

← Analizare



d. [(0, 'a'), (1, 'a'), (2, 'a'), (3, 'a'), (4, 'a')]



Răspunsul dumneavoastră este corect.

Răspunsul corect este:

[(0, 'a'), (1, 'a'), (2, 'a'), (3, 'a'), (4, 'a')]



Întrebare 2

Corect

Marcat 1,00 din 1,00

Dându-se suficienta putere de calcul, care va fi rezultatul apelului de mai jos? (obs: all primește un predicat și o listă și testează dacă toate elementele listei satisfac predicatul)

```
all ((==6) . sum) $ (take (10^20) $ repeat  
  ↳ [1, 2, 3]) ++ (repeat [1, 2, 4])
```

Selectați răspunsul corect:

a. `programul` nu se va termina, va rula
↳ infinit



b. `programul` va întoarce o eroare



c. `True`



d. `False`



Răspunsul dumneavoastră este corect.

Răspunsul corect este:

`False`



d. Răspunsul dumneavoastră este corect.

Răspunsul corect este:

False



Întrebare 3

Corect

Marcat 1,00 din 1,00

Expresia

```
map (div 2) $ filter odd
```

Dă eroare pentru că:

Selectați răspunsul corect:

- a. Operatorul \$ primește o funcție în ↩
↪ doilea argument. ☐
- b. În această situație div trebuie să ↩ibă
↪ backquotes. ☐
- c. Funcția odd nu își primește argumentul. ☐
- d. Filter nu primește al doilea argument. ☒



Răspunsul dumneavoastră este corect.

Răspunsul corect este:

Filter nu primește al doilea argument.



Întrebare 4

Inc corect


Marcat 0,00 din 1,00

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

Selectați răspunsul corect:

← Analizare



b. În această situație div trebuie să ibă
↪ backquotes.



c. Funcția odd nu își primește argumentul.



d. Filter nu primește al doilea argument.



Răspunsul dumneavoastră este corect.

Răspunsul corect este:

Filter nu primește al doilea argument.



Întrebare 4

Inc corect

Marcat 0,00 din 1,00

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

Selectați răspunsul corect:

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



b. evens = [0, 2 ..]



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



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



Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

evens = 0 : zipWith (+) evens evens

← Analizare



Notă obținută
10 din 10 (100%)

Întrebare 1

Corect

Marcat 1,00 din 1,00

Se dă programul:


```
type Point = (Double, Double)
type Points2D = (Point, Point)

getSlope :: Points2D -> Double
getSlope x = ((snd(snd x)) - (snd(fst x)))
  → / ((fst(snd x)) - (fst(fst x)))
```

Ce se va afișa la rularea

```
getSlope ((1,2),(2,3))
```

Selectați răspunsul corect:

- a. 1.0  ☒
- b. eroare, Haskell nu face cast automat între
→ valori de tip Int si Double ☐
- c. 1 ☐
- d. eroare, nu se poate folosi o construcție
→ type în declararea altui type ☐

Răspunsul dumneavoastră este corect.

Răspunsul corect este:

1.0



Întrebare 2

Corect

Marcat 1,00 din 1,00

Pentru definiția:

```
data List a b = Nil | Cons a (List a b) |
```


← Analizare



Răspunsul dumneavoastră este corect.

Răspunsul corect este:

1.0



Întrebare 2

Corect

Marcat 1,00 din 1,00

Pentru definiția:

```
data Lst a b = Nil | Consa a (Lst a b) |
↳ Consb b (Lst a b)
```

care dintre următoarele valori nu este validă?

Selectați răspunsul corect:

a. toate sunt valide



b. Consa Nil \$ Consb Nil Nil



c. Consa [1] \$ Consa [2.5,3] \$ Consb (Add 2)
↳ Nil



d. Consa "a" \$ Consb [2,3] \$ Consa ['1', '2']
↳ Nil



Răspunsul dumneavoastră este corect.

Răspunsul corect este:

toate sunt valide



Întrebare 3

Corect

Marcat 1,00 din 1,00

Fie următorul tip de date definit în Haskell pentru o listă:

```
data LinkedList a = Node a (LinkedList a) | Nil deriving (Show)
```

← Analizare



Întrebare 3

Corect

Marcat 1,00 din 1,00

Fie următorul tip de date definit în Haskell pentru o listă:

```
data LinkedList a = Node a (LinkedList a) | Nil deriving (Show)
```

Pentru acest tip de date implementăm următoarea funcție:

```
func x Nil = Node x Nil  
func x (Node y xs) = Node y (func x xs)
```

Ce va realiza funcția?

Selectați răspunsul corect:

- a. va elimina un element dintr-o listă ☐
- b. va insera un element pe poziția a doua într-o listă ☐
- c. va insera un element la finalul unei liste ✓ ☒
- d. va insera un element la începutul unei liste ☐

Răspunsul dumneavoastră este corect.

Răspunsul corect este:

va insera un element la finalul unei liste

Întrebare 4

Corect

Marcat 1,00 din 1,00

Dorim să definim tipul Graph a pentru reprezentarea grafurilor cu noduri de tip a, folosind liste de adiacență. Alegeți varianta corectă:

Selectați răspunsul corect:

← Analizare



c. va insera un element la finalul unei liste ✓



d. va insera un element la începutul unei liste



Răspunsul dumneavoastră este corect.

Răspunsul corect este:

va insera un element la finalul unei liste

Întrebare 4

Corect

Marcat 1,00 din 1,00

Dorim să definim tipul Graph a pentru reprezentarea grafurilor cu noduri de tip a, folosind liste de adiacență. Alegeți varianta corectă:

Selectați răspunsul corect:

a. `data Graph a = [(a, [a])]`



b. `newtype Graph a = [(a, [a])]`



c. `type Graph a = [(a, [a])]`



d. `struct Graph a = [(a, [a])]`



Răspunsul dumneavoastră este corect.

Răspunsul corect este:

`type Graph a = [(a, [a])]`

← Analizare



0 din 10 (0%)


Întrebare 1

Inc corect

Marcat 0,00 din 1,00

Care ar fi tipul corect al unei funcții care caută o cheie într-o listă de perechi (cheie, valoare) și întoarce Just valoare dacă găsește cheia în listă, respectiv Nothing în caz contrar?

Selectați răspunsul corect:

- a. `Ord a => a -> [(a, a)] -> Maybe a` ☐
- b. `(Eq a, Eq b) => a -> [(a, b)] -> Maybe b`  ☒
- c. `Eq a => a -> [(a, a)] -> Maybe a` ☐
- d. `Eq a => a -> [(a, b)] -> Maybe b` ☐

Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

`Eq a => a -> [(a, b)] -> Maybe b`

Întrebare 2

Inc corect

Marcat 0,00 din 1,00

Fie următorul tip de date care exprimă un număr fracționar, prin numărător și numitor:

```
data Fraction = MakeFraction { getNumerator :: Int,  
  ↪ getDenominator :: Int } deriving (Show)
```

Care este o supraîncărcare corectă a operatorului

(+)

pentru tipul de date definit mai sus?

← Analizare



Întrebare 2

Incorect

Marcat 0,00 din 1,00

Fie următorul tip de date care exprimă un număr fracționar, prin numărător și numitor:

```
data Fraction = MakeFraction { getNumerator :: Int,  
  ↪ getDenominator :: Int } deriving (Show)
```

Care este o supraîncărcare corectă a operatorului

(+)

pentru tipul de date definit mai sus?

Selectați răspunsul corect:

a.

```
instance Show Fraction where  
  (MakeFraction x1 y1) + (MakeFraction x2 y2) = MakeFraction (x1  
  ↪ * y2 + x2 * y1) (y1 * y2)
```



b.

```
instance Num Fraction where  
  (MakeFraction x1 y1) + (MakeFraction x2 y2) = MakeFraction (x1  
  ↪ * y2 + x2 * y1) (y1 * y2)
```



c.

```
instance Show Fraction where  
  show (MakeFraction x1 y1 + MakeFraction x2 y2) = show $  
  ↪ MakeFraction (x1 * y2 + x2 * y1) (y1 * y2)
```



d.

```
instance Num Fraction where  
  (MakeFraction x) + (MakeFraction y) = MakeFraction $  
  ↪ (getNumerator x * getDenominator y + getNumerator y *  
  ↪ getDenominator x) / (getDenominator x * getDenominator y)
```



Răspunsul dumneavoastră este incorrect.

Răspunsul corect este:

```
instance Num Fraction where  
  (MakeFraction x1 y1) + (MakeFraction x2 y2) = MakeFraction (x1  
  ↪ * y2 + x2 * y1) (y1 * y2)
```

Întrebare 3

Incorect

Marcat 0,00 din 1,00

← Analizare



↪ * yz + xz * y1) (y1 * yz)

Întrebare 3

Incorect

Marcat 0,00 din 1,00

Ponind de la urmatorul cod

```
instance Show (a->b) where
  show f = "f : x -> y"
```

```
data Person = Person Int String
```

Ce va genera rularea codului:

```
> Person 1
```

Selectați răspunsul corect:

- a. `Programul` va genera eroare, tipul de date
↪ `Person` nu este instanta a clasei `Show` ☐
- b. `Programul` va genera eroare, instantierea
↪ trebuia sa impuna ca a si b sa fie deja
↪ in clasa `Show` ☐
- c. `Person 1 : x -> y` ✗ ☒
- d. `f : x -> y` ☐

Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

```
f : x -> y
```

Întrebare 4

Incorect

Marcat 0,00 din 1,00

La ce se va evalua expresia

```
sort [T 2, T 4, T 5, T 6, T 8, T 9]
```

pentru codul Haskell de mai jos?

← Analizare



Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

```
f : x -> y
```

Întrebare 4

Incorect

Marcat 0,00 din 1,00

La ce se va evalua expresia

```
sort [T 2, T 4, T 5, T 6, T 8, T 9]
```

pentru codul Haskell de mai jos?

```
data T = T Int deriving (Eq, Show)
f (T x) = sum [ 1 | a <- [1 .. x],
                 mod x a < 1 ]
instance Ord T where
    x <= y = f x <= f y
```

Selectați răspunsul corect:

a. [T 2, T 4, T 5, T 6, T 8, T 9]

☐

b. [T 2, T 4, T 8, T 6, T 9, T 5]

☐

c. [T 2, T 5, T 4, T 9, T 6, T 8]

☐

d. [T 2, T 4, T 5, T 8, T 9, T 6]

☒

Răspunsul dumneavoastră este incorect.

Răspunsul corect este:

```
[T 2, T 5, T 4, T 9, T 6, T 8]
```