```
<u>Dashboard</u> / My courses / <u>03-ACS-L-A2-S2-PP-CA-CB-CC</u> / <u>20 March - 26 March</u> / <u>Test - Laborator 3</u>
      Started on Tuesday, 28 March 2023, 2:06 PM
           State Finished
   Completed on Tuesday, 28 March 2023, 2:14 PM
      Time taken 7 mins 44 secs
          Marks 2.00/4.00
          Grade 5.00 out of 10.00 (50%)
Question 1
Incorrect
Mark 0.00 out of 1.00
                                        implementări
  Care
           dintre
                      următoarele
  reprezintă
                     implementare
                                           corectă
  funcționalei filter? 1)
  (define (my-filter f 1)
     (foldl (lambda (x acc)
                   (if (f x)
                        (cons x acc)
                        acc)) null 1))
  2)
  (define (my-filter f 1)
     (foldr (lambda (x acc)
                   (if x
                        (cons (f x) acc)
                        acc)) null 1))
```

- a. 2)
- b. niciuna
- c. ambele
- d. 1) \*

Răspunsul dumneavoastră este incorect.

The correct answer is:

niciuna

```
Question 2

Correct

Mark 1.00 out of 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)

\rightarrow acc)) null L)

(reverse (foldl (lambda (x acc)

\rightarrow (cons (* x x) acc)) null L))
```

- a. exemplul cu foldl
- b. toate implementează corect operația \*
- c. exemplul cu foldr
- od. exemplul cu map

Răspunsul dumneavoastră este corect.

## The correct answer is:

toate implementează corect operația

```
Question 3
Correct
Mark 1.00 out of 1.00
```

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

```
(range x)
 va întoarce
 '(0 1 2 ... x-1)
Select one:
a.
       (define (tabulate idx L)
         (if (null? L)
             null
              (cons (cons idx (car L)) (tabulate (+

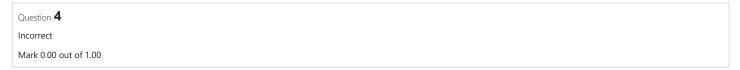
   idx 1) (cdr L)))))
       (tabulate 0 L)
b.
       (define (tabulate L)
          (apply map cons (append (list (range

    (length L))) (list L))))

       (tabulate L)
c.
       (define (tabulate L)
          (map cons (range (length L)) L))
       (tabulate L)
d.
       (define (tabulate L)
         (foldr cons 0 L))
       (tabulate L)
Răspunsul dumneavoastră este corect.
                (define (tabulate L)
                  (foldr cons 0 L))
```

(tabulate L)

The correct answer is:



Fie următoarea funcție:

(define f (
$$\lambda$$
 (e L) (apply list e L)))

Ce funcție (deja implementată în Racket) reprezintă echivalentul funcției de mai sus?

Select one:

- a. cons
- b. list
- ⊚ c. map 🗶
- od. append

Răspunsul dumneavoastră este incorect.

The correct answer is: cons

■ Test - Laborator 2

Jump to...

Test - Laborator 4 ►