Se da C = λ x y z . (x z) y ; I = λx.x ; S = λ x y z . x z (y z). Expresia I C S este echivalenta cu:

Select one:

- a. λ p. λ q. λ r. (q r) (p r)
- b. λ p. λ q. p q
- c. I (S C)
- d. I I
- e. none of the above (nicuna din variantele de mai sus nu este corecta)

Consider the following Haskell expressions: a) (/); b) (/2) c) (2/) d) (2/2) . Which of them represent legit FUNCTIONS (choose the best answer: correct and complete):

Select one:

- a. expressions a and b
- b. expression a solely
- c. expressions a, b, c and d
- d. all answers, except this one, are wrong
- e. expressions a, b and c

Your answer is correct.

The function defined by fun f n = if (n=0) then 0 else f (n-1); is:

Select one:

- a. tail recursive
- b. all answers, except this one, are wrong
- c. not recursive
- d. not tail recursive
- e. not a legit ML function

Your answer is correct.

Consider the following sentences about expressions evaluation in Haskell and ML

(1) by default, Haskell uses lazy evaluation while ML uses call-by-value

(2) one can force strict evaluation of parameters in Haskell

(3) it is possible to "freeze" the evaluation of an expression in ML, thus emulating an almost perfect lazy evaluation mechanism in ML

Which of the following sentences are true (choose the best answer: correct and complete):

Select one:

- a. (1) and (2) are false, (3) is true
- b. (1) is true, (2) and (3) are false
- c. (1), (2) and (3) are all true
- d. (1) and (2) are true, (3) is false
- e. (1) and (3) are true, (2) is false

Your answer is correct.

Care dintre expresiile de mai jos reprezinta **functii** Haskell corecte:

i) filter [1]

ii) filter (>1)

iii) filter (+1)

Select one:

- a. doar ii
- b. i, ii si iii
- c. doar i si ii
- d. doar ii si iii
- e. doar i

Clear my choice

Given C = λ x y z . (x z) y, I = λx.x and S = λ x y z . x z (y z) , expression I C S is equivalent to (choose the best answer: correct and complete):

Select one:

- a. (I C) S only
- b. I (C S) and (I C) S only
- c. λ a. λ b. λ c. (b c) (a c) only
- d. I (C S) only
- e. each of the following three expressions: I (C S) , (I C) S and λ a. λ b. λ c. (b c) (a c)

Your answer is correct.

Can we write a lambda calculus expression with argument n (a natural number) which computes the value of Sudan's recursive function S defined in the first lab (i.e.,

S n x y = if (n=0) then (x+y)

else if (y=0) then x

else (S (n-1) (S n x (y-1)) (y+ S n x (y-1))))?

Select one:

- a. no, because even though recursive calls are possible, we cannot have more than one inside the expression's body
- b. all answers, except this one, are wrong
- c. no, because lambda expressions are "nameless functions", so no recursive call is possible
- d. no, because we cannot compute (n-1) for any positive integer n
- e. no, because we cannot implement addition in lambda calculus

Your answer is correct.

Acti

Se da: B = λ x y z . x (y z) ; I = λ x . x. Expresia (B I B I) este echivalenta cu:

Select one:

- a. B (I (B I))
- b. λ p. λ q. p q
- c. B
- d. toate variantele de mai sus sunt corecte
- e. nicuna din variantele de mai sus nu este corecta

Se da: B = λ x y z . x (y z) ; I = λ x . x. Expresia (B I B I) este echivalenta cu:

Select one:

- a. B (I (B I))
- b. λ p. λ q. p q
- c. B
- d. toate variantele de mai sus sunt corecte
- e. nicuna din variantele de mai sus nu este corecta

Clear my choice

Se da: B = λ x y z . x (y z) ; I = λ x . x. Expresia (B I B I) este echivalenta cu:

Select one:

- a. B (I (B I))
- b. λ p. λ q. p q
- c. B
- d. toate variantele de mai sus sunt corecte
- e. nicuna din variantele de mai sus nu este corecta

Clear my choice

Cand definim in ML functia f, obtinem urmatorul warning: "Matches are not exhaustive". Daca apelam functia f, vom obtine o eroare:

Select one:

- a. niciun raspuns, cu exceptia acestuia, nu este corect
- b. uneori, depinzand de valoarea argumentului
- c. niciodata, indiferent de valoarea argumentului
- d. din cauza ca aceasta definitie contine o eroare de sintaxa
- e. intotdeauna, indiferent de valoarea argumentului

Clear my choice

Implicit, in limbajul Haskell, argumentele functiilor sunt evaluate pe baza urmatoarei strategii:

Select one:

- a. call by name
- b. niciun raspuns, cu exceptia acestuia, nu este corect
- c. functiilor Haskell nu li se transmit argumente, deci nu se pune problema evaluarii acestora
- d. call by value
- e. call by need / lazy evaluation

Clear my choice

Fie functia unfold, definita astfel:

```
unfold :: (t1 -> Bool) -> (t1 -> t) -> (t1 -> t1) -> t1 -> [t]
unfold p h t x
 | p x = []
 | otherwise = (h x) : unfold p h t (t x)
```

Fie functiile urmatoare:

```
f1 :: (t1 -> t) -> [t1] -> [t]
f1 _ [] = []
f1 g (xxs) = (g x):f1 g xs

f2 :: (t -> t) -> t -> [t]
f2 g x = x : f2 g (g x)

f3 :: Integral t => t -> [t]
f3 n
 | n == 0 = []
 | otherwise = (n `mod` 2): f3 (n `div` 2)
```

Dorim sa aflam care din functiile f1, f2, f3 de mai sus poate fi implementata fara a folosi apeluri recursive explicite, ci folosind doar functia unfold, careia i se paseaza valori potrivite pentru p, h si t. Care din urmatoarele afirmatii sunt corecte (alegeti cel mai bun raspuns: corect si complet):

Select one:

- a. niciun raspuns, cu exceptia acestuia, nu este corect si complet
- b. doar f1 si f3 pot fi implementate astfel
- c. doar f1 si f2 pot fi implementate astfel
- d. doar f2 si f3 pot fi implementate astfel

Care din urmatoarele limbaje: Haskell, LISP si ML poate fi considerat functional:

Select one:

- a. Doar Haskell si ML
- b. Doar Haskell
- c. Doar ML
- d. Haskell, LISP si ML
- e. Doar LISP

Clear my choice

Fie functia map :: (a->b) -> [a] -> [b], care are doi parametri, anume o functie si o lista de elemente si care aplica functia data pe fiecare din elementele listei date si returneaza lista rezultatelor obtinute. De exemplu,

map odd [15,16,17] va returna [True,False,True]

Fie urmatoarele afirmatii:

(1) map poate fi implementata folosind doar apeluri recursive ale sale

(2) map poate fi implementata folosind expresii lista (list comprehensions), fara a folosi vreun apel recursiv explicit

(3) map poate fi implementata folosind foldr, operatorii cons (:) si compozitie functionala (.), fara a folosi vreun apel recursiv explicit

Care din urmatoarele afirmatii sunt corecte (alegeti cel mai bun raspuns: corect si complet):

Select one:

- a. doar afirmatia (1) este adevarata
- b. doar afirmatia (2) este adevarata
- c. afirmatiile (1), (2) (3) sunt adevarate
- d. niciun raspuns, cu exceptia acestuia, nu este corect
- e. doar afirmatiile (1) si (2) sunt adevarate

Se considera urmatoarea functie ML:

fun f n = if (n=0) then 0 else f (n-1);

Aceasta functie:

Select one:

- a. nu este recursiva
- b. este recursiva, si anume recursiva in coada
- c. este recursiva, dar nu este recursiva in coada
- d. toate variantele de mai sus sunt corecte
- e. niciuna din variantele de mai sus nu e corecta

Question **4**

Not yet answered

Marked out of 1.00

Flag question

Care dintre expresiile de mai jos reprezinta **functii** Haskell corecte:

i) map

ii) map (+1)

iii) map [1]

Select one:

- a. doar i si ii
- b. i, ii si iii
- c. doar iii
- d. doar i
- e. doar ii

Clear my choice

Fie S functia recursiva a lui Sudan definita astfel:

```
S n x y = if (n=0) then (x+y)
          else if (y=0) then x
               else (S (n-1) (S n x (y-1)) (y+ S n x (y-1))))
```

Se poate scrie o expresie in Calcul Lambda care calculeaza valoarea functiei S pentru un numar natural n dat?

Select one:

- a. nu, pentru ca o expresie lambda nu se poate apela recursiv, neavand asociat un nume
- b. nu, pentru ca operatia de adunare (ex. x+y) nu se poate implementa printr-o expresie lambda
- c. nu, pentru ca operatia de scadere (ex. n-1) nu se poate implementa printr-o expresie lambda
- d. nu, pentru ca in cazul unei expresii lambda, putem avea doar cel mult un apel recursiv al acesteia din interiorul corpului sau (body)
- e. toate raspunsurile, cu exceptia acestuia, sunt gresite

Clear my choice

Fie functia map :: (a->b) -> [a] -> [b], care are doi parametri, anume o functie si o lista de elemente si care aplica functia data pe fiecare din elementele listei date si returneaza lista rezultatelor obtinute. De exemplu,

map odd [15,16,17] va returna [True,False,True]

Fie urmatoarele afirmatii:

(1) map poate fi implementata folosind doar apeluri recursive ale sale

(2) map poate fi implementata folosind expresii lista (list comprehensions), fara a folosi vreun apel recursiv explicit

(3) map poate fi implementata folosind foldr, operatorii cons (:) si compozitie functionala (.), fara a folosi vreun apel recursiv explicit

Care din urmatoarele afirmatii sunt corecte (alegeti cel mai bun raspuns: corect si complet):

Select one:

- a. doar afirmatiile (1) si (2) sunt adevarate
- b. afirmatiile (1), (2) (3) sunt adevarate
- c. doar afirmatia (2) este adevarata
- d. doar afirmatia (1) este adevarata
- e. niciun raspuns, cu exceptia acestuia, nu este corect

Clear my choice

i) filter [1]

ii) filter (>1)

iii) filter (+1)

Select one:

- a. doar ii
- b. i, ii si iii
- c. doar i si ii
- d. doar ii si iii
- e. doar i

Clear my choice

Question **2**

Not yet answered

Marked out of 1.00

Flag question

Se considera urmatoarea functie ML:

fun f n = if (n=0) then 0 else f (n-1);

Aceasta functie:

Select one:

- a. nu este recursiva
- b. este recursiva, si anume recursiva in coada
- c. este recursiva, dar nu este recursiva in coada
- d. toate variantele de mai sus sunt corecte
- e. niciuna din variantele de mai sus nu e corecta

Clear my choice

Se considera urmatoarea functie ML:

fun f n = if (n=0) then 0 else f (n-1);

Aceasta functie:

Select one:

- a. nu este recursiva
- b. este recursiva, si anume recursiva in coada
- c. este recursiva, dar nu este recursiva in coada
- d. toate variantele de mai sus sunt corecte
- e. niciuna din variantele de mai sus nu e corecta

In Haskell, variabilele isi pot schimba valorile:

Select one:

- a. niciodata
- b. doar prin intermediul unui **let** in interiorul unei functii
- c. oricand, fara nicio conditie
- d. toate variantele de mai sus sunt corecte
- e. niciuna din variantele de mai sus nu este corecta

Question **1**

Not yet answered

Marked out of 1.00

Flag question

Se da: B = λ x y z . x (y z) ; I = λ x . x. Expresia (B I B I) este echivalenta cu:

Select one:

- a. B (I (B I))
- b. λ p. λ q. p q
- c. B
- d. toate variantele de mai sus sunt corecte
- e. nicuna din variantele de mai sus nu este corecta

Implicit, in limbajul Haskell, argumentele functiilor sunt evaluate pe baza urmatoarei strategii:

Select one:

- a. call by name
- b. niciun raspuns, cu exceptia acestuia, nu este corect
- c. call by value
- d. functiilor Haskell nu li se transmit argumente, deci nu se pune problema evaluarii acestora
- e. call by need / lazy evaluation

Clear my choice

In Haskell, variabilele isi pot schimba valorile:

Select one:

- a. niciodata
- b. doar prin intermediul unui **let** in interiorul unei functii
- c. oricand, fara nicio conditie
- d. toate variantele de mai sus sunt corecte
- e. niciuna din variantele de mai sus nu este corecta

Clear my choice

In Haskell, variabilele isi pot schimba valorile:

Select one:

- a. niciodata
- b. doar prin intermediul unui **let** in interiorul unei functii
- c. oricand, fara nicio conditie
- d. toate variantele de mai sus sunt corecte
- e. niciuna din variantele de mai sus nu este corecta

Clear my choice

Se da C = λ x y z . (x z) y ; I = λx.x ; S = λ x y z . x z (y z). Expresia I C S este echivalenta cu:

Select one:

- a. λ p. λ q. λ r. (q r) (p r)
- b. λ p. λ q. p q
- c. I (S C)
- d. I I
- e. none of the above (nicuna din variantele de mai sus nu este corecta)

In Haskell, variabilele isi pot schimba valorile:

Select one:

- a. niciodata
- b. doar prin intermediul unui **let** in interiorul unei functii
- c. oricand, fara nicio conditie
- d. toate variantele de mai sus sunt corecte
- e. niciuna din variantele de mai sus nu este corecta

Clear my choice

tion 1
et
ered
ed out of
g question

Care dintre expresiile de mai jos reprezinta **functii** Haskell corecte:

i) filter [1]

ii) filter (>1)

iii) filter (+1)

Select one:

- a. doar ii si iii
- b. i, ii si iii
- c. doar ii
- d. doar i
- e. doar i si ii

Se considera urmatoarea functie ML:

fun f n = if (n=0) then 0 else f (n-1);

Aceasta functie:

Select one:

- a. nu este recursiva
- b. este recursiva, si anume recursiva in coada
- c. este recursiva, dar nu este recursiva in coada
- d. toate variantele de mai sus sunt corecte
- e. niciuna din variantele de mai sus nu e corecta

Se da: B = λ x y z . x (y z) ; I = λ x . x. Expresia (B I B I) este echivalenta cu:

Select one:

- a. B (I (B I))
- b. λ p. λ q. p q
- c. B
- d. toate variantele de mai sus sunt corecte
- e. nicuna din variantele de mai sus nu este corecta

Clear my choice

Implicit, in limbajul Haskell, argumentele functiilor sunt evaluate pe baza urmatoarei strategii:

Select one:

- a. functiilor Haskell nu li se transmit argumente, deci nu se pune problema evaluarii acestora
- b. call by name
- c. niciun raspuns, cu exceptia acestuia, nu este corect
- d. call by value
- e. call by need / lazy evaluation

Care dintre expresiile de mai jos reprezinta **functii** Haskell corecte:

i) filter [1]

ii) filter (>1)

iii) filter (+1)

Select one:

- a. doar ii si iii
- b. doar ii
- c. doar i si ii
- d. i, ii si iii
- e. doar i

Clear my choice

Se da C = λ x y z . (x z) y ; I = λx.x ; S = λ x y z . x z (y z). Expresia I C S este echivalenta cu:

Select one:

- a. λ p. λ q. λ r. (q r) (p r)
- b. λ p. λ q. p q
- c. I (S C)
- d. I I
- e. none of the above (nicuna din variantele de mai sus nu este corecta)

Implicit, in limbajul ML, argumentele functiilor sunt evaluate pe baza urmatoarei strategii:

Select one:

- a. niciun raspuns, cu exceptia acestuia, nu este corect
- b. functiilor ML nu li se transmit argumente, deci nu se pune problema evaluarii acestora
- c. call by value
- d. call by need / lazy evaluation
- e. call by name

Care dintre expresiile de mai jos reprezinta **functii** Haskell corecte:

i) map

ii) map (+1)

iii) map [1]

Select one:

- a. doar i
- b. doar i si ii
- c. doar ii
- d. i, ii si iii
- e. doar iii

Clear my choice

Implicit, in limbajul ML, argumentele functiilor sunt evaluate pe baza urmatoarei strategii:

Select one:

- a. niciun raspuns, cu exceptia acestuia, nu este corect
- b. functiilor ML nu li se transmit argumente, deci nu se pune problema evaluarii acestora
- c. call by value
- d. call by need / lazy evaluation
- e. call by name

Se da: B = λ x y z . x (y z) ; I = λ x . x. Expresia (B I B I) este echivalenta cu:

Select one:

- a. B (I (B I))
- b. λ p. λ q. p q
- c. B
- d. toate variantele de mai sus sunt corecte
- e. nicuna din variantele de mai sus nu este corecta

Fie functia map :: (a->b) -> [a] -> [b], care are doi parametri, anume o functie si o lista de elemente si care aplica functia data pe fiecare din elementele listei date si returneaza lista rezultatelor obtinute. De exemplu,

map odd [15,16,17] va returna [True,False,True]

Fie urmatoarele afirmatii:

(1) map poate fi implementata folosind doar apeluri recursive ale sale

(2) map poate fi implementata folosind expresii lista (list comprehensions), fara a folosi vreun apel recursiv explicit

(3) map poate fi implementata folosind foldr, operatorii cons (:) si compozitie functionala (.), fara a folosi vreun apel recursiv explicit

Care din urmatoarele afirmatii sunt corecte (alegeti cel mai bun raspuns: corect si complet):

Select one:

- a. doar afirmatiile (1) si (2) sunt adevarate
- b. afirmatiile (1), (2) (3) sunt adevarate
- c. doar afirmatia (2) este adevarata
- d. niciun raspuns, cu exceptia acestuia, nu este corect
- e. doar afirmatia (1) este adevarata

In Haskell, variabilele isi pot schimba valorile:

Select one:

- a. niciodata
- b. doar prin intermediul unui **let** in interiorul unei functii
- c. oricand, fara nicio conditie
- d. toate variantele de mai sus sunt corecte
- e. niciuna din variantele de mai sus nu este corecta

Fie S functia recursiva a lui Sudan definita astfel:

```
S n x y = if (n=0) then (x+y)
          else if (y=0) then x
               else (S (n-1) (S n x (y-1)) (y+ S n x (y-1))))
```

Se poate scrie o expresie in Calcul Lambda care calculeaza valoarea functiei S pentru un numar natural n dat?

Select one:

- a. nu, pentru ca in cazul unei expresii lambda, putem avea doar cel mult un apel recursiv al acesteia din interiorul corpului sau (body)
- b. toate raspunsurile, cu exceptia acestuia, sunt gresite
- c. nu, pentru ca o expresie lambda nu se poate apela recursiv, neavand asociat un nume
- d. nu, pentru ca operatia de adunare (ex. x+y) nu se poate implementa printr-o expresie lambda
- e. nu, pentru ca operatia de scadere (ex. n-1) nu se poate implementa printr-o expresie lambda

Fie functia unfold, definita astfel:

```
unfold :: (t1 -> Bool) -> (t1 -> t) -> (t1 -> t1) -> t1 -> [t]
unfold p h t x
 | p x = []
 | otherwise = (h x) : unfold p h t (t x)
```

Fie functiile urmatoare:

```
f1 :: (t1 -> t) -> [t1] -> [t]
f1 _ [] = []
f1 g (xxs) = (g x):f1 g xs

f2 :: (t -> t) -> t -> [t]
f2 g x = x : f2 g (g x)

f3 :: Integral t => t -> [t]
f3 n
 | n == 0 = []
 | otherwise = (n `mod` 2): f3 (n `div` 2)
```

Dorim sa aflam care din functiile f1, f2, f3 de mai sus poate fi implementata fara a folosi apeluri recursive explicite, ci folosind doar functia unfold, careia i se paseaza valori potrivite pentru p, h si t.
Care din urmatoarele afirmatii sunt corecte (alegeti cel mai bun raspuns: corect si complet):

Select one:

- a. doar f3 poate fi implementata astfel
- b. niciun raspuns, cu exceptia acestuia, nu este corect si complet
- c. doar f2 si f3 pot fi implementate astfel

Care din urmatoarele limbaje: Haskell, LISP si ML poate fi considerat functional:

Select one:

- a. Doar LISP
- b. Doar ML
- c. Doar Haskell si ML
- d. Doar Haskell
- e. Haskell, LISP si ML

Clear my choice

Fie functia unfold, definita astfel:

```
unfold :: (t1 -> Bool) -> (t1 -> t) -> (t1 -> t1) -> t1 -> [t]
unfold p h t x
 | p x = []
 | otherwise = (h x) : unfold p h t (t x)
```

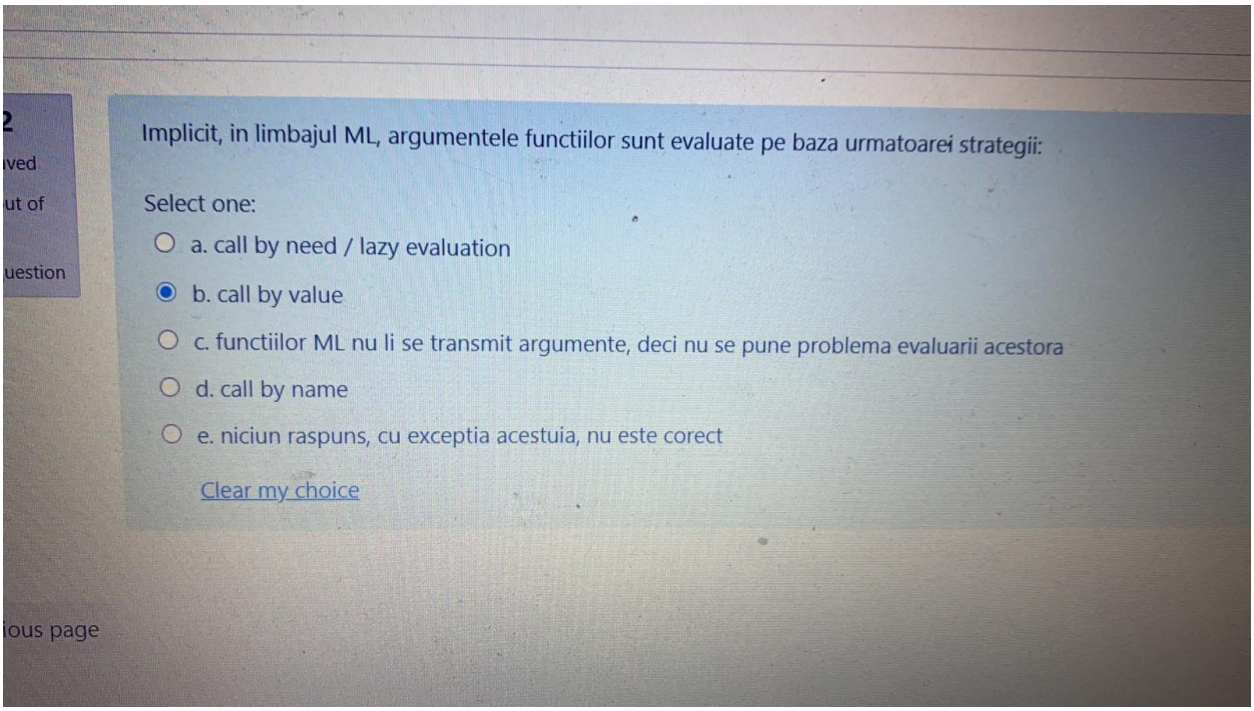
Fie functiile urmatoare:

```
f1 :: (t1 -> t) -> [t1] -> [t]
f1 _ [] = []
f1 g (x:xs) = (g x):f1 g xs

f2 :: (t -> t) -> t -> [t]
f2 g x = x : f2 g (g x)

f3 :: Integral t => t -> [t]
f3 n
 | n == 0 = []
 | otherwise = (n `mod` 2): f3 (n `div` 2)
```

Dorim sa aflam care din functiile f1, f2, f3 de mai sus poate fi implementata
Care din urmatoarele afirmatii sunt corecte (alegeti cel mai bun raspuns: co

Select one:

- a. doar f1 si f2 pot fi implementate astfel
- b. doar f1 si f3 pot fi implementate astfel
- c. doar f2 si f3 pot fi implementate astfel
- d. doar f3 poate fi implementata astfel
- e. niciun raspuns, cu exceptia acestuia, nu este corect si complet



Se da: B = λ x y z . x (y z) ; I = λ x . x. Expresia (B I B I) este echivalenta cu:

Select one:

- a. B (I (B I))
- b. λ p. λ q. p q
- c. B
- d. toate variantele de mai sus sunt corecte
- e. nicuna din variantele de mai sus nu este corecta

```
f2 g x = x : f2 g (g x)

f3 :: Integral t => t -> [t]
f3 n
  | n == 0 = []
  | otherwise = (n `mod` 2): f3 (n `div` 2)
```

Dorim sa aflam care din functiile f1, f2, f3 de mai sus poate fi implementata fara a folosi apeluri recursive explicite, ci folosind doar functia unfold careia i se paseaza valori potrivite pentru p, h si t. Care din urmatoarele afirmatii sunt corecte (alegeti cel mai bun raspuns: corect si complet):

Select one:

- a. doar f3 poate fi implementata astfel
- b. doar f2 si f3 pot fi implementate astfel
- c. doar f1 si f3 pot fi implementate astfel
- d. doar f1 si f2 pot fi implementate astfel
- e. niciun raspuns, cu exceptia acestuia, nu este corect si complet

Fie S functia recursiva a lui Sudan definita astfel:

```
S n x y = if (n=0) then (x+y)
          else if (y=0) then x
               else (S (n-1) (S n x (y-1)) (y+ S n x (y-1))))
```

Se poate scrie o expresie in Calcul Lambda care calculeaza valoarea functiei S pentru un numar natural n dat?

Select one:

- a. toate raspunsurile, cu exceptia acestuia, sunt gresite
- b. nu, pentru ca operatia de adunare (ex. x+y) nu se poate implementa printr-o expresie lambda
- c. nu, pentru ca o expresie lambda nu se poate apela recursiv, neavand asociat un nume
- d. nu, pentru ca in cazul unei expresii lambda, putem avea doar cel mult un apel recursiv al acesteia din interiorul corpului sau (body)
- e. nu, pentru ca operatia de scadere (ex. n-1) nu se poate implementa printr-o expresie lambda

Clear my choice

Se da C = λ x y z . (x z) y ; I = λx.x ; S = λ x y z . x z (y z). Expresia I C S este echivalenta cu:

Select one:

- a. λ p. λ q. λ r. (q r) (p r)
- b. λ p. λ q. p q
- c. I (S C)
- d. I I
- e. none of the above (nicuna din variantele de mai sus nu este corecta)

Clear my choice

Cand definim in ML functia f, obtinem urmatorul warning: "Matches are not exhaustive". Daca apelam functia f, vom obtine o eroare:

Select one:

- a. niciodata, indiferent de valoarea argumentului
- b. din cauza ca aceasta definitie contine o eroare de sintaxa
- c. niciun raspuns, cu exceptia acestuia, nu este corect
- d. uneori, depinzand de valoarea argumentului
- e. intotdeauna, indiferent de valoarea argumentului

Se da: B = λ x y z . x (y z) ; I = λ x . x. Expresia (B I B I) este echivalenta cu:

Select one:

- a. B (I (B I))
- b. λ p. λ q. p q
- c. B
- d. toate variantele de mai sus sunt corecte
- e. nicuna din variantele de mai sus nu este corecta

Clear my choice

Cand definim in ML functia f, obtinem urmatorul warning: "Matches are not exhaustive". Daca apelam functia f, vom obtine o eroare:

Select one:

- a. niciun raspuns, cu exceptia acestuia, nu este corect
- b. intotdeauna, indiferent de valoarea argumentului
- c. niciodata, indiferent de valoarea argumentului
- d. uneori, depinzand de valoarea argumentului
- e. din cauza ca aceasta definitie contine o eroare de sintaxa

Se da: B = λ x y z . x (y z) ; I = λ x . x. Expresia (B I B I) este echivalenta cu:

Select one:

- a. B (I (B I))
- b. λ p. λ q. p q
- c. B
- d. toate variantele de mai sus sunt corecte
- e. nicuna din variantele de mai sus nu este corecta

Clear my choice

Cand definim in ML functia f, obtinem urmatorul warning: "Matches are not exhaustive". Daca apelam functia f, vom obtine o eroare:

Select one:

- a. din cauza ca aceasta definitie contine o eroare de sintaxa
- b. uneori, depinzand de valoarea argumentului
- c. intotdeauna, indiferent de valoarea argumentului
- d. niciodata, indiferent de valoarea argumentului
- e. niciun raspuns, cu exceptia acestuia, nu este corect