

## Question 1

Millised järgnevaist näidistest ei ole korrektsed?

Vali üks või enam vastust.

- a. `(, ) x y` ✗
- b. `(1 , [])` ✗ paari `(1,[])` näidis
- c. `(x , x)` ✓
- d. `([] , (x , y))` ✗
- e. `(1 , 0)` ✗ paari `(1,0)` näidis

Millised järgnevaist näidistest on korrektsed?

Vali üks või enam vastust.

- a. `x : Just xs` ✗
- b. `x : Nothing` ✗
- c. `Just x : xs` ✓
- d. `Just x : Just xs` ✗ kooloni teine argument peab olema listitüüpi, aga on nurjumisega tüüpi
- e. `Just x xs` ✗

Millised järgnevaist näidistest on korrektsed?

Vali üks või enam vastust.

- a. `Just : xs` ✗ kooloni vasakuks argumentiks oleval konstruktoril `Just` liiga vähe argumente
- b. `Just Just` ✗
- c. `x Just` ✗
- d. `Just (x : xs)` ✓
- e. `Just (x : Nothing)` ✗

Millised järgnevaist näidistest ei ole korrektsed?

Vali üks või enam vastust.

- a. `x : y : z` ✗
- b. `(:) x xs` ✗
- c. `x : y` ✗
- d. `1 : 0` ✓ tüüp ei klapi: kooloni teine argument peab olema list
- e. `1 : x` ✗

Millised järgnevaist näidistest on korrektsed?

Vali üks või enam vastust.

- a. `x / y` ✗ / pole konstruktor

- b. `x : y` ✓ mittetühja listi näidis (`x` märgib pead ja `y` saba)
- c. `x * y` ✗ `*` pole konstruktor
- d. `x , y` ✗
- e. `x y` ✗

Millised järgnevaist näidistest on korrektsed?

Vali üks või enam vastust.

- a. `(x && y) : []` ✗
- b. `Just (x : xs)` ✓ nurjumisega tüüpi väärtuse näidis, kus reaalseks andmeks on mittetühi list
- c. `x `Just` y` ✗
- d. `(x : xs) Nothing` ✗
- e. `Nothing (x : xs)` ✗

Millised järgnevaist näidistest on korrektsed?

Vali üks või enam vastust.

- a. `Just (x : Nothing)` ✗
- b. `Just Just` ✗
- c. `x Just` ✗
- d. `Just : xs` ✗
- e. `Just (x : xs)` ✓ nurjumisega tüüpi väärtuse näidis, kus reaalseks andmeks on mittetühi list

Millised järgnevaist näidistest ei ole korrektsed?

Vali üks või enam vastust.

- a. `([], (x, y))` ✗ sellise paari näidis, mille esimene komponent on tühi list ja teine on mingi paar
- b. `(,) x y` ✗
- c. `(1, [])` ✗ paari `(1,[])` näidis
- d. `(1, 0)` ✗
- e. `(x, x)` ✓

Millised järgnevaist näidistest ei ole korrektsed?

Vali üks või enam vastust.

- a. `Just a` ✗
- b. `Just log` ✗ nurjumisega tüüpi, kuid õnnestunud väärtuse näidis
- c. `Just Nothing` ✗

- d. Just ✓ argumente liiga vähe
- e. Nothing ✗

Millised järgnevaist näidistest on korrektsed?  
Vali üks või enam vastust.

- a. (:) x xs xss ✗
- b. x : x ✗
- c. (:) ✗
- d. xs : xs ✗
- e. (x : xs) : xss ✓

Millised järgnevaist näidistest on korrektsed?  
Vali üks või enam vastust.

- a. Just (x : xs) ✓ nurjumisega tüüpi väärtuse näidis, kus reaalseks andmeks on mittetühi list
- b. x `Just` y ✗
- c. Nothing (x : xs) ✗
- d. (x : xs) Nothing ✗
- e. (x && y) : [] ✗

Millised järgnevaist näidistest ei ole korrektsed?  
Vali üks või enam vastust.

- a. [] : xs ✗ sellise listide listi näidis, mille esimene elementlist on tühi
- b. (1 : x) : y ✗
- c. 1 : [] ✗
- d. (x : xs) : xs ✓
- e. 1 : 0 : xs ✗

Millised järgnevaist näidistest ei ole korrektsed?  
Vali üks või enam vastust.

- a. properFraction 2.5 ✓
- b. (sin , cos) ✗ suvalise paari näidis (muutuja sin märgib esimest komponenti ja cos teist)
- c. ei : ole : korrektne : [] ✗
- d. (2 , 0.5) ✗
- e. (pi , e) ✗

Millised järgnevaist näidistest on korrektsed?

Vali üks või enam vastust.

- a.  $\text{Just } x : xs$  ✓ nurjumisega tüüpi elementide listi näidis, kus esimene element on õnnestumine ( $x$  märgib esimese elemendi reaalselt väärtust ja  $xs$  listi saba)
- b.  $x : \text{Just } xs$  ✗ kooloni teine argument peab olema listitüüpi, aga on nurjumisega tüüpi
- c.  $x : \text{Nothing}$  ✗
- d.  $\text{Just } x \ xs$  ✗
- e.  $\text{Just } x : \text{Just } xs$  ✗ kooloni teine argument peab olema listitüüpi, aga on nurjumisega tüüpi

Millised järgnevaist näidistest ei ole korrektsed?

Vali üks või enam vastust.

- a.  $1 : []$  ✗
- b.  $(x : xs) : xs$  ✓ muutuja kordub
- c.  $(1 : x) : y$  ✗
- d.  $[] : xs$  ✗
- e.  $1 : 0 : xs$  ✗

Millised järgnevaist näidistest ei ole korrektsed?

Vali üks või enam vastust.

- a.  $\text{Just}$  ✓ argumente liiga vähe
- b.  $\text{Nothing}$  ✗
- c.  $\text{Just Nothing}$  ✗ nurjumisega tüüpi, kuid õnnestunud väärtuse näidis, kus mähitav väärtus on omakorda nurjumine
- d.  $\text{Just log}$  ✗
- e.  $\text{Just a}$  ✗

Millised järgnevaist näidistest on korrektsed?

Vali üks või enam vastust.

- a.  $x \ y$  ✗
- b.  $x / y$  ✗
- c.  $x * y$  ✗
- d.  $x , y$  ✗
- e.  $x : y$  ✓ mittetühja listi näidis ( $x$  märgib pead ja  $y$  saba)

Millised järgnevaist näidistest ei ole korrektsed?

Vali üks või enam vastust.

- a.  $1 : []$  ✗
- b.  $[] : xs$  ✗
- c.  $(x : xs) : xs$  ✓ muutuja kordub
- d.  $1 : 0 : xs$  ✗
- e.  $(1 : x) : y$  ✗

## Question 2

Kas deklaratsioon

```
[p : ps]
  = "Has" : "kell" : []
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

- a. "Has" ✗
- b. täht H ✗
- c. bottom ✓
- d. [] ✗
- e. ei ole tüübikorrektne ✗

Näidis on struktuurilt  $(p : ps) : []$ .

Näidis ei sobitu, sest [] ei sobitu mittetühja listiga "kell":[].

Seega  $p$  väärtus on bottom.

Kas deklaratsioon

```
[p : ps]
  = "Haskell"
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

- a. täht H ✗
- b. [] ✗
- c. bottom ✗
- d. ei ole tüübikorrektne ✓
- e. "Haskell" ✗

Vasakul on listinäidis, mille elemendinäidis on omakorda listinäidis.

Paremal on sümbolite list, sümbolid ei ole ise listid. Seega tüübid ei klapi.

Kas deklaratsioon

```
[p : ps]
  = "Has" : "kell" : []
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

- a. [] ✗
- b. ei ole tüübikorrektne ✗

- c. "Has" ✗
- d. bottom ✓
- e. täht H ✗

Näidis on struktuurilt  $(p : ps) : []$ .

Näidis ei sobitu, sest  $[]$  ei sobitu mittetühja listiga "kell": $[]$ .

Seega  $p$  väärtus on bottom.

Kas deklaratsioon

```
p : ~(q : qs)
  = [replicate 5 'X']
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

- a. "XXXXXX" ✓
- b. täht X ✗
- c. bottom ✗
- d. ei ole tüübikorrektne ✗
- e. "replicate" ✗

$[replicate 5 'X']$  väärtus on kujul "XXXXXX": $[]$ .

Näidis sobitub, sobituselemendi  $p$  väärtus on "XXXXXX" ja sobituselemendi  $q : qs$  väärtus on  $[]$ .

Kas deklaratsioon

```
p : ~(q : qs)
  = [replicate 5 'X']
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

- a. bottom ✗
- b. täht X ✗
- c. ei ole tüübikorrektne ✗
- d. "XXXXXX" ✓
- e. "replicate" ✗

$[replicate 5 'X']$  väärtus on kujul "XXXXXX": $[]$ .

Näidis sobitub, sobituselemendi  $p$  väärtus on "XXXXXX" ja sobituselemendi  $q : qs$  väärtus on  $[]$ .

Kas deklaratsioon

```
~(p , q) : qs
  = [replicate 5 'X']
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

- a. bottom ✗
- b. täht X ✗

- c. "replicate" ✗
- d. ei ole tüübikorrektne ✓
- e. "XXXXX" ✗

Vasakul pool on listinäidis, mille elemendinäidis on paarinäidis.

Paremal pool on sõnede list. Sõne on (sümbolite) list, mitte paar. Järelikult tüübid ei klapi.

Kas deklaratsioon

```
p : ~(q : qs)
  = [replicate 5 'X']
```

on tüübikorrektne? Kui jah, siis mis saab muutuja p väärtuseks?

Vali üks vastus.

- a. ei ole tüübikorrektne ✗
- b. täht X ✗
- c. "XXXXX" ✓
- d. bottom ✗
- e. "replicate" ✗

[replicate 5 'X'] väärtus on kujul "XXXXX":[].

Näidis sobitub, sobituselemendi p väärtus on "XXXXX" ja sobituselemendi q : qs väärtus on [].

Kas deklaratsioon

```
p : q : qs
  = [2]
```

on tüübikorrektne? Kui jah, siis mis saab muutuja q väärtuseks?

Vali üks vastus.

- a. 2 ✗
- b. ei ole tüübikorrektne ✗
- c. list, mille ainus element on bottom ✗
- d. bottom ✓
- e. [] ✗

[2] väärtuse struktuur on 2:[].

Näidis ei sobitu, sest q : qs ei sobitu tühja listiga.

Seega q väärtus on bottom.

Kas deklaratsioon

```
p : q : qs
  = [2]
```

on tüübikorrektne? Kui jah, siis mis saab muutuja p väärtuseks?

Vali üks vastus.

- a. ei ole tüübikorrektne ✗
- b. [] ✗
- c. list, mille ainus element on 2 ✗

d. bottom ✓

e. 2 ✗

[2] väärtuse struktuur on 2:[].

Näidis ei sobitu, sest  $q : qs$  ei sobitu tühja listiga.

Seega  $p$  väärtus on bottom.

Kas deklaratsioon

```
(p : ps) : pss  
= "Has" : "kell" : []
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $pss$  väärtuseks?

Vali üks vastus.

a. täht l ✗

b. bottom ✗

c. list ainsa elemendiga "kell" ✓

d. ei ole tüübikorrektne ✗

e. "kell" ✗

$p : ps$  väärtus on "Has" ja  $pss$  väärtus on "kell":[].

Kas deklaratsioon

```
~(p , q) : qs  
= [replicate 5 'X']
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

a. täht X ✗

b. "XXXXXX" ✗

c. "replicate" ✗

d. ei ole tüübikorrektne ✓

e. bottom ✗

Vasakul pool on listinäidis, mille elemendinäidis on paarinäidis

Paremal pool on sõnede list. Sõne on (sümbolite) list, mitte paar. Järelikult tüübid ei klapi.

Kas deklaratsioon

```
p : q : qs  
= [2]
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $q$  väärtuseks?

Vali üks vastus.

a. ei ole tüübikorrektne ✗

b. list, mille ainus element on bottom ✗

c. [] ✗

d. 2 ✗

e. bottom ✓



[2] väärtuse struktuur on 2:[].

Näidis ei sobitu, sest  $q : qs$  ei sobitu tühja listiga.

Seega  $q$  väärtus on bottom.

Kas deklaratsioon

```
p : q : qs  
= [2]
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

- a. [] ✗
- b. bottom ✓
- c. list, mille ainus element on 2 ✗
- d. 2 ✗
- e. ei ole tüübikorrektne ✗

[2] väärtuse struktuur on 2:[].

Näidis ei sobitu, sest  $q : qs$  ei sobitu tühja listiga.

Seega  $p$  väärtus on bottom.

Kas deklaratsioon

```
~(p , q) : qs  
= [replicate 5 'X']
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

- a. "replicate" ✗
- b. täht X ✗
- c. bottom ✗
- d. "XXXXX" ✗
- e. ei ole tüübikorrektne ✓

Vasakul pool on listinäidis, mille elemendinäidis on paarinäidis.

Paremal pool on sõnade list. Sõne on (sümbolite) list, mitte paar. Järelikult tüübid ei klapi.

Kas deklaratsioon

```
p : q : qs  
= [2]
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

- a. bottom ✓
- b. list, mille ainus element on 2 ✗
- c. ei ole tüübikorrektne ✗
- d. 2 ✗
- e. [] ✗

[2] väärtuse struktuur on 2:[].

Näidis ei sobitu, sest  $q : qs$  ei sobitu tühja listiga.

Seega  $p$  väärtus on bottom.

Kas deklaratsioon

```
p : ~(q : qs)
  = [replicate 5 'X']
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

- a. bottom ✗
- b. "XXXXXX" ✓
- c. "replicate" ✗
- d. ei ole tüübikorrektne ✗
- e. täht X ✗

[replicate 5 'X'] väärtus on kujul "XXXXXX":[].

Näidis sobitub, sobituselemendi  $p$  väärtus on "XXXXXX" ja sobituselemendi  $q : qs$  väärtus on [].

Kas deklaratsioon

```
~(p , q) : qs
  = [replicate 5 'X']
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

- a. täht X ✗
- b. bottom ✗
- c. "XXXXXX" ✗
- d. "replicate" ✗
- e. ei ole tüübikorrektne ✓

Vasakul pool on listinäidis, mille elemendinäidis on paarinäidis.

Paremal pool on sõnede list. Sõne on (sümbolite) list, mitte paar. Järelikult tüübid ei klapi.

Kas deklaratsioon

```
p : ~(q : qs)
  = [replicate 5 'X']
```

on tüübikorrektne? Kui jah, siis mis saab muutuja  $p$  väärtuseks?

Vali üks vastus.

- a. ei ole tüübikorrektne ✗
- b. bottom ✗
- c. täht X ✗
- d. "replicate" ✗
- e. "XXXXXX" ✓

[replicate 5 'X'] väärtus on kujul "XXXXX":[].

Näidis sobitub, sobituselemendi p väärtus on "XXXXX" ja sobituselemendi q : qs väärtus on [].

### Question 3

Kas lambdaavaldise

$\backslash (\_ : \_ : z) \rightarrow z$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv lõpmatu listil, mille iga element on 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. list ainsa elemendiga 2 ✗
- b. bottom ✗
- c. 2 ✗
- d. lõpmatu list, mille iga element on 2 ✓
- e. ei ole tüübikorrektelt rakenduv ✗

Argumendinäidis on vähemalt kahe elemendiga listi näidis, kus z märgib ülejäävat sabaosa (alates kolmandast elemendist). Seega näidis sobitub lõpmatu listiga.

Välja tuleb avaldise z väärtus ehk kahest esimesest elemendist üle jääv sabaosa ise. Lõpmatu listil, mille iga element on 2, on ka iga sabaosa lõpmatu list, mille iga element on 2.

Kas lambdaavaldise

$\backslash (x : \_ : \_) \rightarrow x$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv paaril, mille komponendid on 1 ja 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. bottom ✗
- b. list elementidega 1 ja 2 ✗
- c. paar komponentidega 1 ja 2 ✗
- d. 1 ✗
- e. ei ole tüübikorrektelt rakenduv ✓

Argumendinäidis on vähemalt kahe elemendiga listi näidis. Paar ei ole list, seega tüübid ei klapi.

Kas lambdaavaldise

$\backslash ((x , y) : \_) \rightarrow x + y$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv kaheelemendilisel listil elementidega 1 ja 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. 2 ✗
- b. 3 ✗
- c. ei ole tüübikorrektelt rakenduv ✓
- d. 1 ✗

e. bottom **x**

Argumendinäidis on mittetühja listi näidis, mille elemendinäidis on paarinäidis. Argumentlisti elemendid – arvud 1 ja 2 – ei ole paarid. Seega tüübid ei klapi.

Kas lambdaavaldise

$\backslash (\_ : \_ : z) \rightarrow z$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv lõpmatul listil, mille iga element on 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. ei ole tüübikorrektelt rakenduv **x**
- b. 2 **x**
- c. list ainsa elemendiga 2 **x**
- d. bottom **x**
- e. lõpmatu list, mille iga element on 2 **✓**

Argumendinäidis on vähemalt kahe elemendiga listi näidis, kus  $z$  märgib ülejäävat sabaosa (alates kolmandast elemendist). Seega näidis sobitub lõpmatu listiga.

Välja tuleb avaldise  $z$  väärtus ehk kahest esimesest elemendist üle jääv sabaosa ise. Lõpmatul listil, mille iga element on 2, on ka iga sabaosa lõpmatu list, mille iga element on 2.

Kas lambdaavaldise

$\backslash (\_ : y : \_) \rightarrow y$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv lõpmatul listil, mille elemendid on suuruse järjestuses kõik mittenegatiivsed täisarvud? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. mingi list **x**
- b. bottom **x**
- c. 1 **✓**
- d. 0 **x**
- e. ei ole tüübikorrektelt rakenduv **x**

Argumendinäidis on vähemalt kahe elemendiga listi näidis, kus  $y$  märgib teist elementi (indeksiga 1). Seega näidis sobitub lõpmatu listiga.

Välja tuleb avaldise  $y$  väärtus ehk teine element ise. Lõpmatul listil, mille elemendid on järjest kõik mittenegatiivsed täisarvud, on sel positsioonil arv 1.

Kas lambdaavaldise

$\backslash (\_ : \_ : z) \rightarrow z$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv kaheelemendilisel listil elementidega 1 ja 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. tühi list **✓**
- b. ei ole tüübikorrektelt rakenduv **x**
- c. bottom **x**
- d. 2 **x**

- e. list ainsa elemendiga 2 ✘

Argumendinäidis on vähemalt kahe elemendiga listi näidis, kus  $z$  märgib ülejäävat sabaosa. Seega näidis sobitub kaheelemendilise listiga.

Välja tuleb avaldise  $z$  väärtus ehk see sabaosa ise. Kaheelemendilise listi puhul on see sabaosa tühi.

Kas lambdaavaldise

$\backslash ((x, y) : \_) \rightarrow x + y$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv kolmeelemendilisel listil, mille elemendid on järjest paarid (1,3), (2,4) ja (3,5)? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. ei ole tüübikorrektelt rakenduv ✘
- b. 4 ✔
- c. 3 ✘
- d. 2 ✘
- e. bottom ✘

Argumendinäidis on mittetühja listi näidis, mille elemendinäidis on paarinäidis, kusjuures  $x$  ja  $y$  märgivad esimese paari komponente. Argument on paaride list, seega näidis sobitub.

Välja tuleb  $x$  ja  $y$  väärtuste ehk esimese paari komponentide summa. Elementide (1,3), (2,4), (3,5) puhul on see  $1+3$  ehk 4.

Kas lambdaavaldise

$\backslash (\_ : y : \_) \rightarrow y$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv lõpmatul listil, mille elemendid on suuruse järjestuses kõik mittenegatiivsed täisarvud? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. mingi list ✘
- b. 0 ✘
- c. ei ole tüübikorrektelt rakenduv ✘
- d. bottom ✘
- e. 1 ✔

Argumendinäidis on vähemalt kahe elemendiga listi näidis, kus  $y$  märgib teist elementi (indeksiga 1). Seega näidis sobitub lõpmatu listiga.

Välja tuleb avaldise  $y$  väärtus ehk teine element ise. Lõpmatul listil, mille elemendid on järjest kõik mittenegatiivsed täisarvud, on sel positsioonil arv 1.

Kas lambdaavaldise

$\backslash (\_ : \_ : z) \rightarrow z$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv lõpmatul listil, mille iga element on 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. lõpmatu list, mille iga element on 2 ✔
- b. bottom ✘

- c. 2 ✗
- d. ei ole tüübikorrektelt rakenduv ✗
- e. list ainsa elemendiga 2 ✗

Argumendinäidis on vähemalt kahe elemendiga listi näidis, kus  $z$  märgib ülejäävat sabaosa (alates kolmandast elemendist). Seega näidis sobitub lõpmatu listiga.

Välja tuleb avaldise  $z$  väärtus ehk kahest esimesest elemendist üle jääv sabaosa ise. Lõpmatul listil, mille iga element on 2, on ka iga sabaosa lõpmatu list, mille iga element on 2.

Kas lambdaavaldise

$\backslash (\_ : \_ : z) \rightarrow z$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv lõpmatul listil, mille iga element on 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. lõpmatu list, mille iga element on 2 ✓
- b. ei ole tüübikorrektelt rakenduv ✗
- c. bottom ✗
- d. 2 ✗
- e. list ainsa elemendiga 2 ✗

Argumendinäidis on vähemalt kahe elemendiga listi näidis, kus  $z$  märgib ülejäävat sabaosa (alates kolmandast elemendist). Seega näidis sobitub lõpmatu listiga.

Välja tuleb avaldise  $z$  väärtus ehk kahest esimesest elemendist üle jääv sabaosa ise. Lõpmatul listil, mille iga element on 2, on ka iga sabaosa lõpmatu list, mille iga element on 2.

Kas lambdaavaldise

$\backslash (x : \_ : \_) \rightarrow x$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv üheelemendilisel listil ainsa elemendiga 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. list ainsa elemendiga 2 ✗
- b. ei ole tüübikorrektelt rakenduv ✗
- c. tühi list ✗
- d. bottom ✓
- e. 2 ✗

Argumendinäidis on vähemalt kahe elemendiga listi näidis. Seega näidis kaheelemendilise listiga ei sobitu. Tulemus on bottom.

Kas lambdaavaldise

$\backslash (x : \_ : \_) \rightarrow x$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv üheelemendilisel listil ainsa elemendiga 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. bottom ✓
- b. ei ole tüübikorrektelt rakenduv ✗

- c. 2 ✗
- d. tühi list ✗
- e. list ainsa elemendiga 2 ✗

Argumendinäidis on vähemalt kahe elemendiga listi näidis. Seega näidis kaheelemendilise listiga ei sobitu. Tulemus on bottom.

Kas lambdaavaldise

$\backslash ((x, y) : \_) \rightarrow x + y$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv kaheelemendilisel listil elementidega 1 ja 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. 2 ✗
- b. ei ole tüübikorrektelt rakenduv ✓
- c. 3 ✗
- d. bottom ✗
- e. 1 ✗

Argumendinäidis on mittetühja listi näidis, mille elemendinäidis on paarinäidis. Argumentlisti elemendid – arvud 1 ja 2 – ei ole paarid. Seega tüübid ei klapi.

Kas lambdaavaldise

$\backslash (x : \_ : \_) \rightarrow x$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv paaril, mille komponendid on 1 ja 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. list elementidega 1 ja 2 ✗
- b. paar komponentidega 1 ja 2 ✗
- c. 1 ✗
- d. bottom ✗
- e. ei ole tüübikorrektelt rakenduv ✓

Argumendinäidis on vähemalt kahe elemendiga listi näidis. Paar ei ole list, seega tüübid ei klapi.

Kas lambdaavaldise

$\backslash (\_ : \_ : z) \rightarrow z$

väärtuseks olev funktsioon on tüübikorrektelt rakenduv kaheelemendilisel listil elementidega 1 ja 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. tühi list ✓
- b. list ainsa elemendiga 2 ✗
- c. 2 ✗
- d. bottom ✗

- e. ei ole tüübikorrektset rakenduv ✗

Argumendinäidis on vähemalt kahe elemendiga listi näidis, kus  $z$  märgib ülejäävat sabaosa. Seega näidis sobitub kaheelemendilise listiga.

Välja tuleb avaldise  $z$  väärtus ehk see sabaosa ise. Kaheelemendilise listi puhul on see sabaosa tühi.

Kas lambdaavaldise

$\backslash (\_ : y : \_) \rightarrow y$

väärtuseks olev funktsioon on tüübikorrektset rakenduv lõpmatul listil, mille iga element on 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. 2 ✓
- b. ei ole tüübikorrektset rakenduv ✗
- c. list ainsa elemendiga 2 ✗
- d. bottom ✗
- e. lõpmatu list, mille iga element on 2 ✗

Argumendinäidis on vähemalt kahe elemendiga listi näidis, kus  $y$  märgib teist elementi (indeksiga 1). Seega näidis sobitub lõpmatu listiga.

Välja tuleb avaldise  $y$  väärtus ehk teine element ise. Lõpmatul listil, mille iga element on 2, on ka teine element 2.

Kas lambdaavaldise

$\backslash \sim((x, y) : \_) \rightarrow \text{False}$

väärtuseks olev funktsioon on tüübikorrektset rakenduv tühjal arvupaaride listil? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. ei ole tüübikorrektset rakenduv ✗
- b. bottom ✗
- c. False ✓
- d. (False,False) ✗
- e. tühi list ✗

Argumendinäidis on laisk ja sobitub iga väärtusega.

Välja tuleb tõeväärtus False.

Kas lambdaavaldise

$\backslash (\_ : \_ : z) \rightarrow z$

väärtuseks olev funktsioon on tüübikorrektset rakenduv kaheelemendilisel listil elementidega 1 ja 2? Kui on, siis milline on tulemus?

Vali üks vastus.

- a. ei ole tüübikorrektset rakenduv ✗
- b. 2 ✗
- c. list ainsa elemendiga 2 ✗
- d. tühi list ✓



e. bottom ✗

Argumendinäidis on vähemalt kahe elemendiga listi näidis, kus  $z$  märgib ülejäävat sabaosa. Seega näidis sobitub kaheelemendilise listiga.

Välja tuleb avaldise  $z$  väärtus ehk see sabaosa ise. Kaheelemendilise listi puhul on see sabaosa tühi.

#### Question 4

Kas avaldis

$(\ \ a \in (b, c) \ d \rightarrow b * (c + b) - c) \ (1, -1) \ (0, 1)$

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $c$  väärtuseks? Vali üks vastus.

a. -1 ✓

b. ei ole tüübikorrektne ✗

c. bottom ✗

d. 0 ✗

e. (0,1) ✗

Näidis  $a \in (b, c)$  seotakse avaldisega  $(1, -1)$  ja näidis  $d$  seotakse avaldisega  $(0, 1)$ .

Ehk näidise tõttu sobitatakse nii  $a$  kui ka  $(b, c)$  avaldisega  $(1, -1)$ . Siit  $c$  saab väärtuseks  $-1$ .

Kas avaldis

$(\ \ (a : as) \ (b : bs) \rightarrow a * b + a + b) \ (\text{repeat } (-1)) \ [1 \dots ]$

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $b$  väärtuseks? Vali üks vastus.

a. kõigi positiivsete täisarvude list ✗

b. ei ole tüübikorrektne ✗

c. bottom ✗

d. 1 ✓

e. -1 ✗

Näidis  $(a : as)$  seotakse avaldisega  $\text{repeat } (-1)$  ja näidis  $(b : bs)$  seotakse avaldisega  $[1 \dots ]$ .

Sobitamine õnnestub, sest nii  $\text{repeat } (-1)$  kui ka  $[1 \dots ]$  väärtus on mittetühi list.

Muutuja  $a$  väärtuseks saab  $-1$  ja muutuja  $as$  väärtuseks saab lõpmatu list, mille iga element on  $-1$ . Samuti, muutuja  $b$  väärtuseks saab  $1$  ja muutuja  $bs$  väärtuseks saab lõpmatu list elementidega  $2, 3, \dots$ .

Kas avaldis

$(\ \ ([a, b], [c, d]) \rightarrow a * d - b * c) \ [-1, -1] \ [0, 1]$

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $c$  väärtuseks? Vali üks vastus.

a. list elementidega 0 ja 1 ✗

b. 0 ✗

c. ei ole tüübikorrektne ✓

- d. -1 ✗
- e. bottom ✗

Lambdaavaldise argumendinäidis  $([a, b], [c, d])$  on paarinäidis, kuid argumendiks antakse listitüüpi  $[-1, -1]$ . Seega tüübid ei klapi.

Kas avaldis

$(\lambda [a, b] [c, d] \rightarrow a * d - b * c) (\text{repeat } (-1)) [0, 1]$

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $b$  väärtuseks? Vali üks vastus.

- a. bottom ✓
- b. list elementidega 0 ja 1 ✗
- c. 0 ✗
- d. -1 ✗
- e. ei ole tüübikorrektne ✗

Näidis  $[a, b]$  seotakse avaldisega  $\text{repeat } (-1)$  ja näidis  $[c, d]$  seotakse avaldisega  $[0, 1]$ .

Kuid näidise  $[a, b]$  sobitamine avaldisega  $\text{repeat } (-1)$  vastu ebaõnnestub, sest näidis on kaheelemendilise listi oma, aga  $\text{repeat } (-1)$  väärtus on lõpmatu list.

Seega kõik muutujad saavad väärtuse bottom.

Kas avaldis

$(\lambda (a : as, b : bs) \rightarrow a * b + a + b) (\text{repeat } (-1)) [1 .. ]$

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $b$  väärtuseks? Vali üks vastus.

- a. -1 ✗
- b. bottom ✗
- c. ei ole tüübikorrektne ✓
- d. kõigi positiivsete täisarvude list ✗
- e. 1 ✗

Lambdaavaldise argumendinäidis  $(a : as, b : bs)$  on paarinäidis, kuid argumendiks antakse  $\text{repeat } (-1)$ , mis on listitüüpi. List ei ole paar, seega tüübid ei klapi.

Kas avaldis

$(\lambda a @ (b, c) d \rightarrow b * (c + b) - c) (-1) (0, 1) 1$

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $c$  väärtuseks? Vali üks vastus.

- a. bottom ✗
- b. ei ole tüübikorrektne ✓
- c. 1 ✗
- d. 0 ✗
- e. -1 ✗

Lambdaavaldise esimene argumendinäidis  $a \in (b, c)$  on paarinäidis. Argumendiks tuleb aga arvutüüpi  $-1$ . Arv pole paar, seega tüübid ei klapi. .

Kas avaldis

$(\lambda [a, b] [c, d] \rightarrow a * d - b * c) (\text{repeat } (-1)) [0, 1]$

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $b$  väärtuseks?

Vali üks vastus.

- a. bottom ✓
- b. ei ole tüübikorrektne ✗
- c. 0 ✗
- d. list elementidega 0 ja 1 ✗
- e. -1 ✗

Näidis  $[a, b]$  seotakse avaldisega  $\text{repeat } (-1)$  ja näidis  $[c, d]$  seotakse avaldisega  $[0, 1]$ .

Kuid näidise  $[a, b]$  sobitamine avaldisega  $\text{repeat } (-1)$  vastu ebaõnnestub, sest näidis on kahelemendilise listi oma, aga  $\text{repeat } (-1)$  väärtus on lõpmatu list.

Seega kõik muutujad saavad väärtuse bottom.

Kas avaldis

$(\lambda ([a, b], [c, d]) \rightarrow a * d - b * c) [-1, -1] [0, 1]$

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $c$  väärtuseks?

Vali üks vastus.

- a. -1 ✗
- b. list elementidega 0 ja 1 ✗
- c. bottom ✗
- d. 0 ✗
- e. ei ole tüübikorrektne ✓

Lambdaavaldise argumendinäidis  $([a, b], [c, d])$  on paarinäidis, kuid argumendiks antakse listitüüpi  $[-1, -1]$ . Seega tüübid ei klapi.

Kas avaldis

$(\lambda (a : as) (b : bs) \rightarrow a * b - a + b) (\text{repeat } 1) [1, 3 \dots]$

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $bs$  väärtuseks?

Vali üks vastus.

- a. 1 ✗
- b. bottom ✗
- c. lõpmatu list elementidega 3, 5, 7, jne ✓
- d. ei ole tüübikorrektne ✗
- e. 3 ✗

Näidis  $(a : as)$  seotakse avaldisega  $\text{repeat } 1$  ja näidis  $(b : bs)$  seotakse avaldisega  $[1, 3 \dots]$ .

Sobitamine õnnestub, sest nii `repeat 1` kui ka `[1, 3 .. ]` väärtus on mittetühi list. Muutuja `a` väärtuseks saab 1 ja muutuja `as` väärtuseks saab lõpmatu list, mille iga element on 1. Samuti, muutuja `b` väärtuseks saab 1 ja muutuja `bs` väärtuseks saab lõpmatu list elementidega 3, 5, jne.

Kas avaldis

`(\ a@ (b , c) d -> b * (c - b) + c) (-1 , -1) (0 , 1)`

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja `d` väärtuseks? Vali üks vastus.

- a. -1 ✗
- b. (0,1) ✓
- c. bottom ✗
- d. 1 ✗
- e. ei ole tüübikorrektne ✗

Näidis `a@ (b , c)` seotakse avaldisega `(-1 , -1)` ja näidis `d` seotakse avaldisega `(0 , 1)`. Ehknäidise tõttu sobitatakse nii `a` kui ka `(b , c)` avaldisega `(-1 , -1)`. Kõik sobitamised õnnestuvad.

Seega `d` väärtuseks saab paar `(0,1)`.

Kas avaldis

`(\ (a : as) (b : bs) -> a * b - a + b) (repeat 1) [1, 3 .. ]`

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja `bs` väärtuseks?

Vali üks vastus.

- a. ei ole tüübikorrektne ✗
- b. 1 ✗
- c. lõpmatu list elementidega 3, 5, 7, jne ✓
- d. 3 ✗
- e. bottom ✗

Näidis `(a : as)` seotakse avaldisega `repeat 1` ja näidis `(b : bs)` seotakse avaldisega `[1, 3 .. ]`.

Sobitamine õnnestub, sest nii `repeat 1` kui ka `[1, 3 .. ]` väärtus on mittetühi list.

Muutuja `a` väärtuseks saab 1 ja muutuja `as` väärtuseks saab lõpmatu list, mille iga element on 1. Samuti, muutuja `b` väärtuseks saab 1 ja muutuja `bs` väärtuseks saab lõpmatu list elementidega 3, 5, jne.

Kas avaldis

`(\ [a, b] [c, d] -> a * d - b * c) [-1, -1] [0, 1]`

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja `c` väärtuseks? Vali üks vastus.

- a. list elementidega 0 ja 1 ✗
- b. 0 ✓
- c. bottom ✗

- d. ei ole tüübikorrektne ✗
- e. -1 ✗

Näidis  $[a, b]$  seotakse avaldisega  $[-1, -1]$  näidis  $[c, d]$  seotakse avaldisega  $[0, 1]$ .  
 Sobitamisel muutujad  $a, b, c, d$  sobitatakse vastavalt  $-1, -1, 0, 1$  vastu.  
 Sobitamised õnnestuvad ja  $c$  väärtuseks saab  $0$ .

Kas avaldis

$(\setminus [a, b] [c, d] \rightarrow a * d - b * c) [-1, -1] [0, 1]$

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $c$  väärtuseks?  
 Vali üks vastus.

- a. bottom ✗
- b. list elementidega 0 ja 1 ✗
- c. -1 ✗
- d. ei ole tüübikorrektne ✗
- e. 0 ✓

Näidis  $[a, b]$  seotakse avaldisega  $[-1, -1]$  näidis  $[c, d]$  seotakse avaldisega  $[0, 1]$ .  
 Sobitamisel muutujad  $a, b, c, d$  sobitatakse vastavalt  $-1, -1, 0, 1$  vastu.  
 Sobitamised õnnestuvad ja  $c$  väärtuseks saab  $0$ .

Kas avaldis

$(\setminus (a : as) (b : bs) \rightarrow a * b - a + b) (\text{repeat } 1) [1, 3 \dots ]$

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $bs$  väärtuseks?  
 Vali üks vastus.

- a. lõpmatu list elementidega 3, 5, 7, jne ✓
- b. 1 ✗
- c. 3 ✗
- d. ei ole tüübikorrektne ✗
- e. bottom ✗

Näidis  $(a : as)$  seotakse avaldisega  $\text{repeat } 1$  ja näidis  $(b : bs)$  seotakse avaldisega  $[1, 3 \dots ]$ .

Sobitamine õnnestub, sest nii  $\text{repeat } 1$  kui ka  $[1, 3 \dots ]$  väärtus on mittetühi list.

Muutuja  $a$  väärtuseks saab  $1$  ja muutuja  $as$  väärtuseks saab lõpmatu list, mille iga element on  $1$ . Samuti, muutuja  $b$  väärtuseks saab  $1$  ja muutuja  $bs$  väärtuseks saab lõpmatu list elementidega  $3, 5, 7, \dots$ .

Kas avaldis

$(\setminus (a : as, b : bs) \rightarrow a * b + a + b) (\text{repeat } (-1)) [1 \dots ]$

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $b$  väärtuseks?  
 Vali üks vastus.

- a. ei ole tüübikorrektne ✓
- b. bottom ✗

- c. kõigi positiivsete täisarvude list ✗
- d. 1 ✗
- e. -1 ✗

Lambdaavaldise argumendinäidis ( $a : as, b : bs$ ) on paarinäidis, kuid argumendiks antakse `repeat (-1)`, mis on listitüüpi. List ei ole paar, seega tüübid ei klapi.

Kas avaldis

`(\ [a, b] [c, d] -> a * d - b * c) [-1, -1] [0, 1]`

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $c$  väärtuseks? Vali üks vastus.

- a. 0 ✔
- b. list elementidega 0 ja 1 ✗
- c. bottom ✗
- d. -1 ✗
- e. ei ole tüübikorrektne ✗

Näidis `[a, b]` seotakse avaldisega `[-1, -1]` näidis `[c, d]` seotakse avaldisega `[0, 1]`.

Sobitamisel muutujad  $a, b, c, d$  sobitatakse vastavalt  $-1, -1, 0, 1$  vastu.

Sobitamised õnnestuvad ja  $c$  väärtuseks saab 0.

Kas avaldis

`(\ a@ (b, c) d -> b * (c + b) - c) (1, -1) (0, 1)`

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $c$  väärtuseks? Vali üks vastus.

- a. 0 ✗
- b. ei ole tüübikorrektne ✗
- c. bottom ✗
- d. (0,1) ✗
- e. -1 ✔

Näidis `a@ (b, c)` seotakse avaldisega `(1, -1)` ja näidis `d` seotakse avaldisega `(0, 1)`.

Ehk näidise tõttu sobitatakse nii  $a$  kui ka  $(b, c)$  avaldisega `(1, -1)`. Siit  $c$  saab väärtuseks  $-1$ .

Kas avaldis

`(\ a@ (b, c) d -> b * (c + b) - c) (-1) (0, 1) 1`

on tüübikorrektne? Kui on, siis mis saab väärtustamise käigus lokaalse muutuja  $c$  väärtuseks? Vali üks vastus.

- a. bottom ✗
- b. 1 ✗
- c. ei ole tüübikorrektne ✔
- d. 0 ✗
- e. -1 ✗

Lambdaavaldisel esimesel argumentinäidis  $a \in (b, c)$  on paarinäidis. Argumendiks tuleb aga arvutüüpi  $-1$ . Arv pole paar, seega tüübid ei klapi.