Question 1

On antud kood

```
kasFac1
  :: (Num a, Integral b)
  => a -> b -> a
kasFac1 x k
  = case compare k 0 of
      GT
        -> x * kasFac1 (x + 1) (k - 1)          -- (1)
      EQ
        -> 1                                    -- (2)
      _
        -> error "kasFac1: neg. teine argument" -- (3)
```

Millised on operaatori `kasFac1` (täisargumenteeritud) väljakutsete tulemused avaldise

```
kasFac1 3 4
```

väärtustamise käigus?

Õige vastus: 1,6,30,120,360

On antud kood

```
kasFac1
  :: (Num a, Integral b)
  => a -> b -> a
kasFac1 x k
  = case compare k 0 of
      GT
        -> x * kasFac1 (x + 1) (k - 1)          -- (1)
      EQ
        -> 1                                    -- (2)
      _
        -> error "kasFac1: neg. teine argument" -- (3)
```

Millised haruvalikud tehakse avaldise

```
kasFac1 3 4
```

väärtustamise käigus?

Õige vastus: 11112

On antud kood

```
kasFac2
  :: (Num a, Integral b)
  => a -> b -> a
kasFac2 x k
  = case compare k 0 of
```

```
      GT
        -> kasFac2 x (k - 1) * (x + fromIntegral (k - 1)) -- (1)
      EQ
        -> 1                                              -- (2)
      _
        -> error "kasFac2: neg. teine argument"           -- (3)
```

Millised haruvalikud tehakse avaldise

```
kasFac2 3 4
```

väärtustamise käigus?

Õige vastus:11112

On antud kood

```
kasFac1
  :: (Num a, Integral b)
  => a -> b -> a
kasFac1 x k
  = case compare k 0 of
      GT
        -> x * kasFac1 (x + 1) (k - 1)          -- (1)
      EQ
        -> 1                                    -- (2)
      _
        -> error "kasFac1: neg. teine argument" -- (3)
```

Millised on operaatori `kasFac1` esimese argumendi väärtused tema väljakutsetel avaldise

```
kasFac1 (-3) 1
```

väärtustamise käigus?

Õige vastus: -3,-2

Question 2

On antud kood

```
võrdlePikkust
  :: [a] -> [a] -> Ordering
võrdlePikkust (_ : xs) (_ : ys)
  = võrdlePikkust xs ys         -- (1)
võrdlePikkust (_ : _ ) _
  = GT                          -- (2)
võrdlePikkust _        (_ : _ )
  = LT                          -- (3)
võrdlePikkust _        _
  = EQ                          -- (4)
```

Millised on operaatori `võrdlePikkust` esimese argumendi väärtused tema väljakutsetel avaldise

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

väärtustamise käigus?

Õige vastus: [1,2,3],[2,3],[3]

On antud kood

```
võrdlePikkust
  :: [a] -> [a] -> Ordering
võrdlePikkust (_ : xs) (_ : ys)
  = võrdlePikkust xs ys          -- (1)
võrdlePikkust (_ : _ ) _
  = GT                           -- (2)
võrdlePikkust _         (_ : _ )
  = LT                           -- (3)
võrdlePikkust _         _
  = EQ                           -- (4)
```

Millised on operaatori `võrdlePikkust` teise argumendi väärtused tema väljakutsetel avaldise

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

väärtustamise käigus?

Õige vastus: [4,5],[5],[]

On antud kood

```
võrdlePikkust
  :: [a] -> [a] -> Ordering
võrdlePikkust (_ : xs) (_ : ys)
  = võrdlePikkust xs ys          -- (1)
võrdlePikkust (_ : _ ) _
  = GT                           -- (2)
võrdlePikkust _         (_ : _ )
  = LT                           -- (3)
võrdlePikkust _         _
  = EQ                           -- (4)
```

Millised haruvalikud tehakse avaldise

```
võrdlePikkust [1, 3, 5] [2, 4, 6]
```

väärtustamise käigus?

Õige vastus: 1114

Question 3

On antud kood

```
needi
  :: (Eq a)
  => [a] -> [a] -> [a]
needi (x : xs) bs@ ~(y : ys)
  | not (null xs)
    = x : needi xs bs        -- (1)
  | not (null bs) && x == y
    = bs                     -- (2)
needi as       _
  = as                       -- (3)
```

Millised on operaatori `needi` teise argumendi väärtused tema väljakutsetel avaldise

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

väärtustamise käigus?

Õige vastus: [3,4,5],[3,4,5],[3,4,5]

On antud kood

```
needi
  :: (Eq a)
  => [a] -> [a] -> [a]
needi (x : xs) bs@ ~(y : ys)
  | not (null xs)
    = x : needi xs bs        -- (1)
  | not (null bs) && x == y
    = bs                     -- (2)
needi as       _
  = as                       -- (3)
```

Millised haruvalikud tehakse avaldise

```
needi [1, 3, 6] []
```

väärtustamise käigus?

Õige vastus: 113

On antud kood

```
needi
  :: (Eq a)
  => [a] -> [a] -> [a]
needi (x : xs) bs@ ~(y : ys)
  | not (null xs)
    = x : needi xs bs        -- (1)
  | not (null bs) && x == y
    = bs                     -- (2)
```

```
needi as
  = as                  -- (3)
```

Millised on operaatori `needi` esimese argumendi väärtused tema väljakutsetel avaldise

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

väärtustamise käigus?

Õige vastus: [1,2,3,4],[2,3,4],[3,4],[4]

On antud kood

```
needi
  :: (Eq a)
  => [a] -> [a] -> [a]
needi (x : xs) bs@ ~(y : ys)
  | not (null xs)
    = x : needi xs bs        -- (1)
  | not (null bs) && x == y
    = bs                     -- (2)
needi as
  = as                       -- (3)
```

Millised on operaatori `needi` esimese argumendi väärtused tema väljakutsetel avaldise

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

väärtustamise käigus?

Õige vastus: [1,2,3],[2,3],[3]

Question 4

On antud kood

```
vaheliti2
  :: [a] -> [a] -> [a]
vaheliti2 (x : xs) (y : ys)
  = x : y : vaheliti2 xs ys -- (1)
vaheliti2 as       []
  = as                      -- (2)
vaheliti2 _        bs
  = bs                      -- (3)
```

Millised haruvalikud tehakse avaldise

```
vaheliti2 [1, 2, 3, 4] [5]
```

väärtustamise käigus?

Õige vastus: 12

On antud kood

```
vaheliti1
  :: [a] -> [a] -> [a]
vaheliti1 (x : xs) bs
  = x : vahelitiA xs bs -- (1)
vaheliti1 _        bs
  = bs                -- (2)
vahelitiA as (y : ys)
  = y : vaheliti1 as ys -- (3)
vahelitiA as _
  = as                -- (4)
```

Millised on operaatorite `vaheliti1` ja `vahelitiA` teise argumendi väärtused nende väljakutsetel avaldise

```
vaheliti1 [1, 2, 3, 4] [5]
```

väärtustamise käigus?

Õige vastus: [5],[5],[],[]

On antud kood

```
vaheliti1
  :: [a] -> [a] -> [a]
vaheliti1 (x : xs) bs
  = x : vahelitiA xs bs -- (1)
vaheliti1 _        bs
  = bs                -- (2)
vahelitiA as (y : ys)
  = y : vaheliti1 as ys -- (3)
vahelitiA as _
  = as                -- (4)
```

Millised haruvalikud tehakse avaldise

```
vaheliti1 [1, 2, 3, 4] [5]
```

väärtustamise käigus?

Õige vastus: 1314