

Kontrolltöö nr 1 aines “Programmerimiskeeled”

2014 sügis

Variant 1

1. Kirjutada programm, mis toob ekraanile listi paaridest, mille esimesed komponendid on järjest kõik tekstisümbolid koodiga 32 kuni 127 ning teised komponendid on samad sümbolid tsükliliselt kolme koha võrra nihutatult (st esimeses paaris on teiseks komponendiks sümbol koodiga 35 ning viimases koodiga 34). Väljund vormistada ühe korrektse eestikeelse täislausena (näiteks kujul “Nõutud list on ...”; täpitahtede asemel võib kasutada vastavaid täppideta tähti).

2. Vastavuses signatuuriga

```
tahejama
:: IO String
```

defineerida muutuja `tahejama` väärtuseks protseduur, mis küsib eesti keeles kasutajalt sõnesisendit ja kui see algab suurtähega, siis edastab selle muutmata kujul. Kui kasutaja antud sõne algab väiketähega, siis edastab protseduur sellest esitähe suureks muutmise teel saadud sõne. Kui kasutaja antud sõne ei alga üldse tähega, siis küsib protseduur kasutajalt eesti keeles uue sõne ja edastab selle muutmata kujul.

3. Vastavuses signatuuriga

```
eemaldaUle
:: Int -> [a] -> [a]
```

defineerida muutuja `eemaldaUle` väärtuseks funktsioon, mis võtab argumendiks täisarvu n ja listi l ja koostab tulemuseks listi, mille saab listist l iga n -nda elemendi väljajätmisel. Kui esimene argument pole positiivne, peab arvutus lõppema sobiva eestikeelse veateatega.

Näiteks:

- `eemaldaUle 2 [1 .. 10]` väärtus on list elementidega 1, 3, 5, 7, 9.

4. Vastavuses signatuuriga

```
jarjestikusteArv
:: (a -> a -> Bool) -> [a] -> Int
```

defineerida muutuja `jarjestikusteArv` väärtuseks funktsioon, mis võtab argumendiks binaarse predikaadi p ja listi l ja loendab listis l kõik need kohad, kus p kahel järjestikusel elemendil kehtib.

Näiteks:

- `jarjestikusteArv (<=) [1, 2, 5, 0, 1]` väärtus on 3.

5. Defineerida uus ühe parameetriga tüüpipere `OmaList`, mille tüüpidesse kuuluvad järjendid, milles elemendid paiknevad kolmekaupa rühmades, kus kaks elementi on parameetriga määratud tüüpi ning kolmas tüüpi `Int`. Struktuur võib olla ka tühi.

Vastavuses signatuuriga

```
parisList
:: OmaList Int -> [Int]
```

defineerida muutuja `parisList` väärtuseks funktsioon, mis teisendab `OmaList`-tüüpi järjendi tavaliseks Haskellis listiks, säilitades elementide omavahelise järjestuse.

Kontrolltöö nr 1 aines “Programmerimiskeeled”

2014 sügis

Variant 2

1. Kirjutada programm, mis toob ekraanile sõne, mille alguses on kõik suurtähed tähestiku järjestuses, seejärel suurtähed vastupidises järjestuses, siis jälle tähestiku järjestuses, siis vastupidises järjestuses jne, kokku mõlemat pidi 20 korda. Suuna vahetamisel peavad tähed Z ja A esinema ühekordselt. Väljund vormistada ühe korrektse eestikeelse täislausena (näiteks kujul “Nõutud sõne on ...”; täpitähtede asemel võib kasutada vastavaid täppideta tähti), milles nõutud sümbolite rivi asub jutumärkide vahel.

2. Vastavuses signatuuriga

```
juhud
:: [a] -> Int -> [a]
```

defineerida muutuja `juhud` väärtuseks funktsioon, mis võtab argumentiks listi l ja täisarvu n ja koostab tulemuseks listi, mille saab listi l pea vahetamisel elemendiga positsioonil n (kus positsioone nummerdame 0st), kui positsioon n listis leidub, vastasel korral listi l muutmata kujul. Kui argumentiks saadud täisarv on negatiivne, peab arvutus lõpetama sobiva eestikeelse veateatega.

Näiteks:

- `juhud "HELLO" 3` väärtus on sõne “LELHO”;
- `juhud "HELLO" 5` väärtus on sõne “HELLO”.

3. Vastavuses signatuuriga

```
mang
:: [Int] -> IO Bool
```

defineerida muutuja `mang` väärtuseks funktsioon, mis tsüklis kirjutab standardväljundisse argumentiks saadud listist järjest arve ja loeb pärast iga arvu standardsisendist ühe rea. Kui loetud reas on täpselt esitatud arvu jagu sümboleid, siis tegevus jätkub, vastasel korral protseduur lõpetab kohe ja edastab väära. Kui listis saavad arvud otsa, edastab protseduur tõese.

4. Vastavuses signatuuriga

```
tabel
:: (Num a, Ord a)
=> [a] -> a -> [[a]]
```

defineerida muutuja `tabel` väärtuseks funktsioon, mis võtab argumentiks arvulisti l ja arvulise ülempiiri m ning koostab listide listi kujul lõpmatu ridade arvuga tabeli, mille esimene rida koosneb ainult nullidest, iga järgmine rida on saadud jooksvast reast listi l vastaval positsioonil asuvate elementide juurdeliitmisel ning kõigis ridades on tabelisse võetud ainult pikim algusjupp, kus elemendid ei ületa piiri m .

Näiteks `tabel [1 ..] 10` annab tabeli, mille esimeses reas on lõpmatuseni ainult nullid, teises reas on järjestikused naturaalarvud 1st 10ni, kolmandas paarisarvud 2st 10ni, neljandas 3ga jaguvad arvud jne.

5. Defineerida uus ühe parameetriga tüübibipere `UusKahend`, mille tüüpi-desse kuuluvad kahendpuustruktuurid, kus lehtedes on väärtused tüüpi `Int` ning igas vahetipus on üks väärtus parameetriga määratud tüüpi ja üks väärtus tüüpi `Int`. Tühja puud need tüübid ei sisalda.

Vastavuses signatuuriga

```
satiKorgus
:: UusKahend a -> UusKahend a
```

defineerida muutuja `satiKorgus` väärtuseks funktsioon, mis võtab argumentiks `UusKahend`-tüüpi puu ja annab välja sama kujuga kahendpuu, mille saab algsest, seades tippudes `Int`-tüüpi väärtused võrdseks vastava alampuu kõrgusega.

Kontrolltöö nr 1 aines “Programmerimiskeeled”

2014 sügis

Variant 3

1. Kirjutada programm, mis toob ekraanile kõigi kolmekohaliste 1-ga lõpevate ja 9-ga lõppevate naturaalarvude korrutise. Väljund vormistada ühe korrektse eestikeelse täislausena (näiteks kujul “Nõutud korrutis on ...”; täpitahtede asemel võib kasutada vastavaid täppideta tähti).
2. Vastavuses signatuuriga

```
kasvud
:: String -> String
```

defineerida muutuja `kasvud` väärtuseks funktsioon, mis koostab tulemuseks sõne, milles on pikkuse kasvu järjestuses ladina tähestiku kõikvõimalikud algused (A-st suvalise täheni), kasutades eraldajana argumentiks saadud sõnet. Igas “alguses” on esitäht (A) suur ja ülejäänud väiksed.

Näiteks `kasvud " "` puhul on tulemusena oodatava sõne algus “A, Ab, Abc, Abcd, ...”, lõpus aga on “... Abcdefghijklmnopqrstuvwxyz”.

3. Vastavuses signatuuriga

```
jarjMaxSum
:: (Ord a, Num a)
=> [a] -> a
```

defineerida muutuja `jarjMaxSum` väärtuseks funktsioon, mis leiab argumentiks saadud arvulisti suurima järjestikuste mittenegatiivsete elementide summa.

Näiteks:

- `jarjMaxSum [2, 7, -1, 1, 5, 1, 1]` väärtus on 9.

4. Vastavuses signatuuriga

```
vastavateArv
:: (Char -> Char -> Bool) -> IO Int
```

defineerida muutuja `vastavateArv` väärtuseks funktsioon, mis argumentiks saadud binaarsel predikaadil p annab tulemuseks protseduuri, mis eestikeelse dialoogi abil küsib kasutajalt kaks rida ja edastab arvu, kui mitmel positsioonil on esimese ja teise rea sümbolid omavahel seoses p .

5. Defineerida uus tüüp `Valjaanne` raamatute ja perioodikaväljaannete kirjade kodeerimiseks. Raamatute puhul peavad eksisteerima väljad pealkirja ja ilmumisaasta hoidmiseks, perioodika puhul peab olema ka numbriväli.

Vastavuses signatuuriga

```
sisaldabSarnast
:: Valjaanne -> [Valjaanne] -> Bool
```

defineerida muutuja `sisaldabSarnast` väärtuseks predikaat, mis võtab argumentiks ühe väljaandekirje v ja väljaandekirjete listi l ning kontrollib, kas listis l leidub väljaandega v sama pealkirja ja ilmumisaastaga väljaanne.

Kontrolltöö nr 1 aines “Programmerimiskeeled”

2014 sügis

Variant 4

1. Kirjutada programm, mis toob ekraanile paaride listi, kus paaride esimesed komponendid on kasvavas järjestuses kõik kahe- ja kolmekohalised 0ga või 5ga lõppevad naturaalarvud ning iga paari teine komponent on 2 või 3 vastavalt sellele, kas esimene komponent on 2- või 3-kohaline. Väljund vormistada ühe korrektse eestikeelse täislausena (näiteks kujul “Nõutud list on ...”; täpitähtede asemel võib kasutada vastavaid täppideta tähti).

2. Vastavuses signatuuriga

```
sulanda
  :: (Num a)
  => [a] -> [a] -> [a]
```

defineerida muutuja `sulanda` väärtuseks funktsioon, mis võtab argumentideks kaks arvulist: kui üks neist on tühi, siis on tulemuseks ülejääv list; kui kumbki argumentlist pole tühi, kuid üks neist on üheelemendiline, siis on tulemuseks ülejääv list, mille esimesele elemendile on liidetud üheelemendilise listi ainus element; kui mõlemas argumentlistis on vähemalt kaks elementi, siis on tulemuseks teine argumentlist.

Näiteks:

- `sulanda [1, 5, 6] [8]` annab tulemuseks listi elementidega 9, 5, 6;
- `sulanda [5, 6] [7, 2]` annab tulemuseks listi elementidega 7, 2.

3. Vastavuses signatuuriga

```
diagonaal
  :: [[a]] -> a -> [a]
```

defineerida muutuja `diagonaal` väärtuseks funktsioon, mis võtab argumentideks listide listi, mida interpreteerime kahemõõtmelise tabeli ridade listina, ja `n`-ö vaikeelemendi ning koostab tulemuseks listi, mis koosneb tabeli ülemisest vasakust nurgast algava diagonaali elementidest. Tulemuslistis peab olema samapalju elemente kui tabelis ridu; kui mõni rida on nii lühike, et ei ulatugi diagonaalini, tuleb vastaval positsioonil kasutada vaikeelementi.

Näiteks

- `diagonaal ["D", "Mo", "!", "Koll"] ' #'` väärtus on “Do#l”.

4. Vastavuses signatuuriga

```
dialloog
  :: String -> IO ()
```

defineerida muutuja `dialloog` väärtuseks funktsioon, mis annab tulemuseks protseduuri, mis laseb kasutajal tippida standardsisendist ühe rea, kuid kuvab iga sisestatud sümboli eraldi reale. Ridade ees kasutab protseduur viibana funktsiooni argumentist saadud sõnet. Sissejuhatuseks tuleb samasuguse viibaga esitada kasutajale tegevuse üleskutse (nt “Kirjuta rida!”) ning ka lõpetuseks samal viisil mõni sobiv lause.

5. Defineerida uus ühe parameetriga tüüpipere `RikasKahend`, mille tüüpidesse kuuluvad kahendpuustruktuurid, mille igas tipus on üks parameetriga määratud tüüpi anne ja üks tõeväärtus. Puu võib olla ka tühi.

Vastavuses signatuuriga

```
paremaJargi
  :: RikasKahend a -> RikasKahend a
```

defineerida muutuja `paremaJargi` väärtuseks funktsioon, mis koostab argumentideks saadud puustruktuuri järgi sama kujuga puu, mis erineb originaalpuust selle poolest, et igas lehttipus on tõeväärtus tõene ning iga vahetipu tõeväärtus on vastupidine `parema` alluva tõeväärtusele.

Kontrolltöö nr 1 aines “Programmerimiskeeled”

2014 sügis

Variant 5

1. Kirjutada programm, mis toob ekraanile 200-elementilise paaride listi, kus paaride esimeste komponentidena korduvad tsükliliselt kõik väiketähed a-st z-ni ning teiste komponentidena kõik arvud 0-st 9-ni. Väljund vormistada ühe korrektse eestikeelse täislausena (näiteks kujul “Nõutud list on ...”; täpitahtede asemel võib kasutada vastavaid täppideta tähti). Tähed paarides peavad olema esitatud ülakomade vahel, kuid numbrid mitte.

2. Vastavuses signatuuriga

```
korda
  :: (Int , Int) -> Char -> IO ()
```

defineerida muutuja `korda` väärtuseks funktsioon, mis võtab argumentiks täisarvude paari (u, v) ja tekstisümboli c ning annab tulemuseks protseduuri, mis kirjutab ekraanile ühte ritta u sümbolit c , kui $u \geq 0$ (vastasel korral lõpetab kohe töö, heites kasutajaerindi olukorda kirjeldava eestikeelse veateatega) ja järgmisele reale v sümbolit c , kui $v \geq 0$ (jällegi, juhul $v < 0$ heidab protseduur kasutajaerindi olukorda kirjeldava eestikeelse veateatega, aga kui samal ajal $u \geq 0$, siis teeb seda alles pärast esimese rea kirjutamist).

3. Vastavuses signatuuriga

```
uleUheVastavateArv
  :: String -> [Int] -> Int
```

defineerida muutuja `uleUheVastavateArv` väärtuseks funktsioon, mis võtab argumentideks sõne ja arvulisti ja loeb kokku nende paarisarvulise indeksiga (alustades nullist) positsioonide arvu, kus sõnes paikneva sümboli kood võrdub listis samal positsioonil paikneva arvuga.

Näiteks:

- `uleUheVastavateArv "abcdef" [97, 98, 99]` annab 2.

4. Vastavuses signatuuriga

```
huppa
  :: (Eq a)
  => (a -> a) -> [a] -> ([a] , [a])
```

defineerida muutuja `huppa` väärtuseks funktsioon, mis võtab argumentideks funktsiooni f ja listi l ning jaotab listi l elemendid kahte listi u ja v , kummaski elementide omavahelist järjestust muutmata; mittetühja listi l korral peab listi u esimene element võrduma listi l esimese elemendiga ja iga järgmine element olema algses listis jooksvale elemendile järgnevatest elementidest selline, mis saadakse jooksvale elemendile funktsiooni f rakendamisel (kui sellist pole, siis list u ei jätku).

Näiteks:

- `huppa (2 *) [1 .. 10]` annab paari, mille esimene komponent on list elementidega 1, 2, 4, 8 ning teine komponent on list elementidega 3, 5, 6, 7, 9, 10.

5. Kalapüügivõistlusel lähivad arvesse püütud täiemõõdulised haugid, ahvenad ja latikad. Haugi puhul loetakse täiemõõdulieks kaalu vähemalt 7.5 kg, ahvena puhul kaalu vähemalt 0.8 kg ning latika puhul vähemalt 1.2 kg. Defineerida uus tüüp `Kala`, millesse kuuluvate väärtustega saaks kodeerida haugi-, ahvena- ja latikaisendeid koos lipuga, mis tähistab täiemõõdulisust või alamõõdulisust, kuid mitte mingeid muid kalu ega muid andmeid.

Vastavuses signatuuriga

```
minKogukaal
  :: [Kala] -> Double
```

defineerida muutuja `minKogukaal` väärtuseks funktsioon, mis püütud kalade andmete listi alusel arvutaks vähima võimaliku täiemõõduliste kalade kogukaalu.