

TÖL304G

Forritunarmál

Verkefnablað 11

Snorri Agnarsson

25. október 2024

Verkefni — Assignments

Íslenska

Munið að öll föll þurfa lýsingu með Notkun/Fyrir/Gildi eða Notkun/Fyrir/Eftir.

Hópverkefni

Sækið skrána `condMap.hs-skeleton` úr Canvas og vistið hana sem `condMap.hs`. Klárið að forrita föllin tíu sem eru skilgreind í skránni, þýðið forritið og keyrið það. Skilið skránni í Gradescope og sýnið einnig útkomuna úr keyrslu forritsins.

Einstaklingsverkefni

1. Skrifið fall `listAll` í Haskell sem tekur viðföng i , n og f með $i \leq n + 1$ og skilar listanum $[f(i), f(i + 1), \dots, f(n)]$. Athugið að `listAll 1 0 f` skal skila `[]`.
2. Skrifið Haskell fall `powerList` sem tekur heiltölur i og j sem viðföng og skilar lista af öllum undirlistum listans $[i, \dots, j]$. Við reiknum með að $[i, \dots, j]$ tákni lista þeirra heiltalna k þar sem $i \leq k \leq j$, í vaxandi röð, og við leyfum að $i > j$. Athugið, að eins og áður, þá telst listi $[y_1, y_2, \dots, y_k]$ vera undirlisti listans $[x_1, x_2, \dots, x_n]$ ef unnt er að fá $[y_1, y_2, \dots, y_k]$ með því að fjarlægja núll eða fleiri x_i úr $[x_1, x_2, \dots, x_n]$ og breyta ekki röð gildanna sem eftir eru. Ekki nota nein hjálparföll önnur en þau sem innbyggð eru í Haskell.

English

Remember that all function need a description with Use/Pre/Value or Use/Pre/Post.

Group Assignments

Fetch the file `condMap.hs-skeleton` from Canvas and store it as `condMap.hs`. Finish programming the ten functions defined in the file, compile the program and run it. Turn in the file to Gradescope and also show the result of running the program.

Individual Assignments

1. Write a function `listAll` in Haskell which takes arguments i , n , and f with $i \leq n + 1$ and returns the list $[f(i), f(i + 1), \dots, f(n)]$. Note that `listAll 1 0 f` should return `[]`.
2. Write a Haskell function `powerList` which takes integers i and j as arguments and returns a list of all sublists of the list $[i, \dots, j]$. We assume that $[i, \dots, j]$ denotes the list of those integers k where $i \leq k \leq j$, in ascending order, and we allow $i > j$. Note that, as before, the list $[y_1, y_2, \dots, y_k]$ is considered to be a sublist of the list $[x_1, x_2, \dots, x_n]$ if we can produce $[y_1, y_2, \dots, y_k]$ by removing zero or more x_i 's from $[x_1, x_2, \dots, x_n]$ and not change the order of the values left. Do not use any helper functions other than those that are built-in with Haskell.