Haskell — Definition of higher-order functions (2) P71775_en

This problem explores the definition of higher-order functions on lists.

1. Define a function *countIf* :: (**Int** → **Bool**) → [**Int**] → **Int** that, given a predicate on integers and a list of integers, returns the number of elements in the list that satify the predicate.

2. Define a function *pam* :: [**Int**] → [**Int** → **Int**] → [[**Int**]] that, given a list of integers and a list of functions from integers to integers, returns the list consisting if applying each of the functions in the second list to the elements in the first list.

3. Define a function *pam2* :: [**Int**] → [**Int** → **Int**] → [[**Int**]] that, given a list of integers and a list of functions from integers to integers, returns the list of lists where each list if the result of applying, one after the other, the function in the second list to each element in the first list.

4. Define a function *filterFoldl* :: (**Int** → **Bool**) → (**Int** → **Int** → **Int**) → **Int** → [**Int**] → **Int** that returns a fold of all the elements that satisfy the given predicate.

5. Define a function *insert* :: (**Int** → **Int** → **Bool**) → [**Int**] → **Int** → [**Int**] that, given a relation between integers, a list and un element, return the list with the inserted element according to the relation.

   Use function *insert*, in order to define function *insertionSort* :: (**Int** → **Int** → **Bool**) → [**Int**] → [**Int**] that orders a list according to the given relation.

Scoring

Each item scores 20 points.

Sample input

```
countIf (>5) [1..10]
pam [1,2,3] [(+1),(*2),(^2)]
pam2 [1,2,3] [(+1),(*2),(^2)]
filterFoldl even (*) 1 [4,7,2,4,9,3]
insert (<) [1,4,6,9,12] 8
insertionSort (>) [4,5,2,3,1,3]
```

Sample output

```
5
[[2,3,4],[2,4,6],[1,4,9]]
[[2,2,1],[3,4,4],[4,6,9]]
32
[1,4,6,8,9,12]
[5,4,3,3,2,1]
```

Problem information

Author : Albert Rubio / Jordi Petit
Translator : Jordi Petit
Generation : 2024-05-02 22:41:39