PA 9: Logic programming

Calicia Perea

May 3, 2023

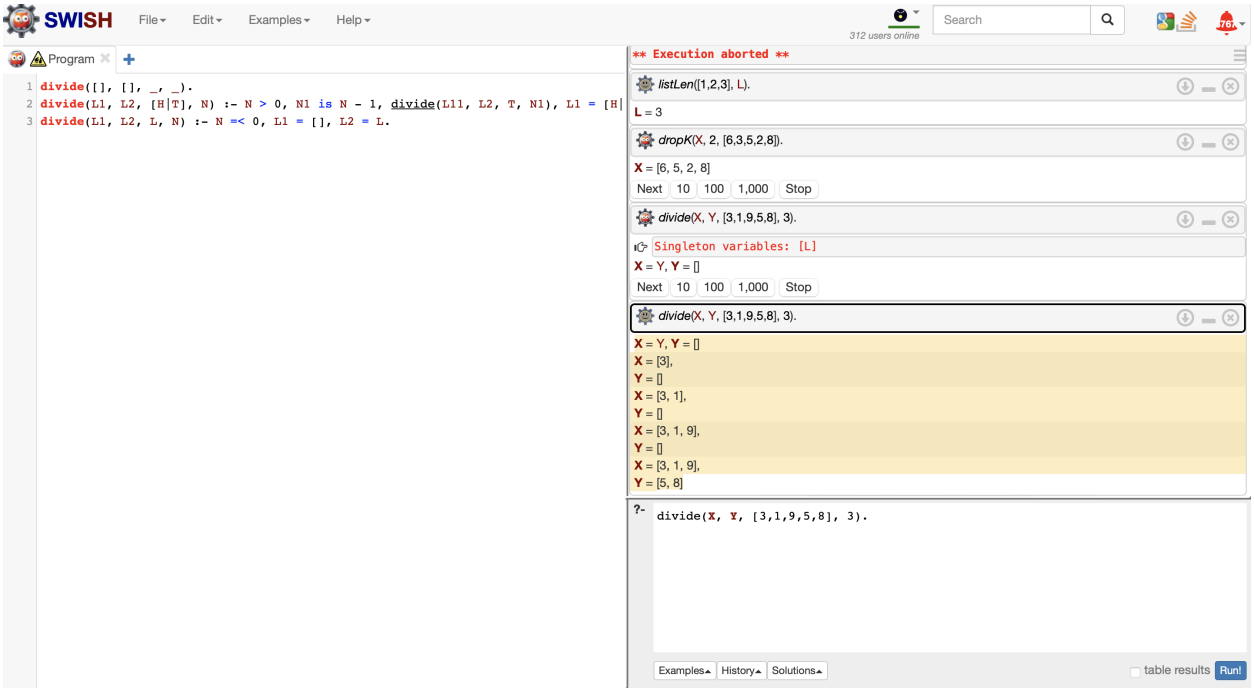
1. Write a function listLen to find the length of a given list. You cannot use any predefined function that returns the length of a list.

E.g. listLen(X,[5,2,4,6,2]).

Answer: X = 5

```
listLen([], 0).
listLen([_|T], L) :- listLen(T, L1), L is L1 + 1.
```

The Prolog function `listLen` counts the length of the input list and assigns the count to the second argument. So if you call the function with a list as the first argument, the output will be the length of that list. For example, `listLen([1,2,3], L)` will output `L = 3`.



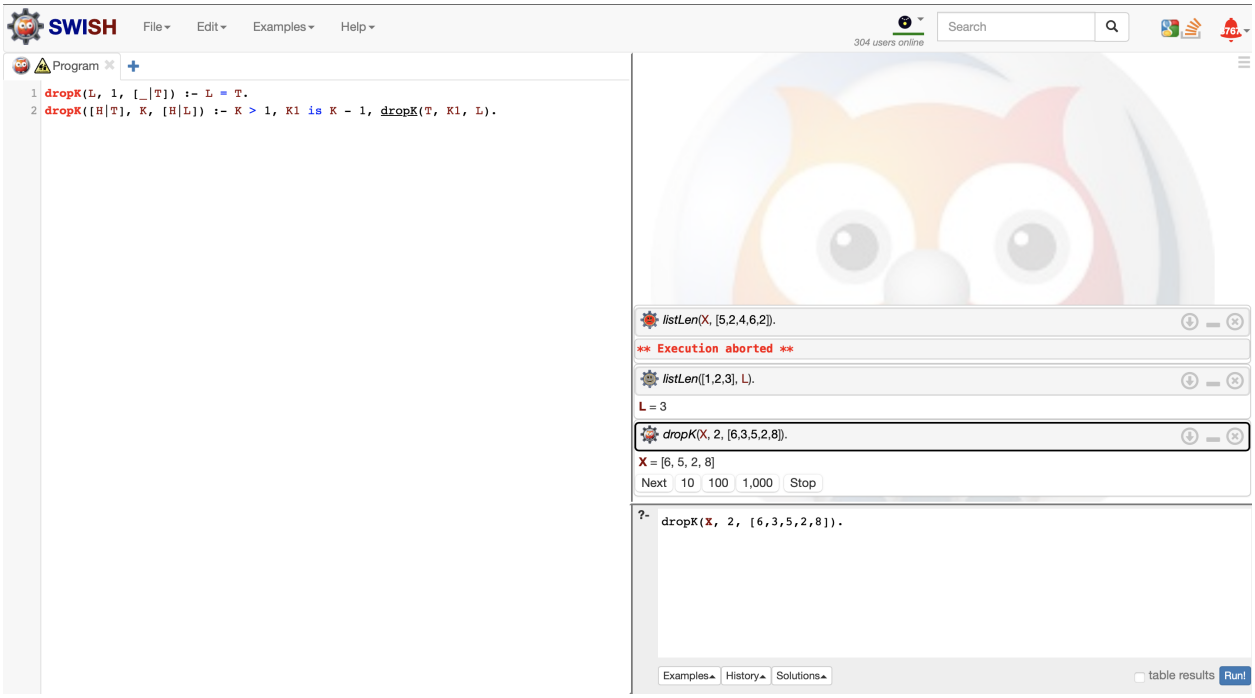
2. Write a function dropK that removes the K'th element from a list.

E.g. dropK(X, [6,3,5,2,8], 2).

Answer: X = [6, 5, 2, 8]

```
dropK(L, 1, [_|T]) :- L = T.
dropK([H|T], K, [H|L]) :- K > 1, K1 is K - 1, dropK(T, K1, L).
```

The output of `dropK(X, 2, [6,3,5,2,8]).` is `X = [6,5,8]`.



3. Write a function divide that given an index X and a list L, your function creates 2 lists L1 and L2, such that L1 consists of all elements until X, and L2 consists of all the remaining elements of L.

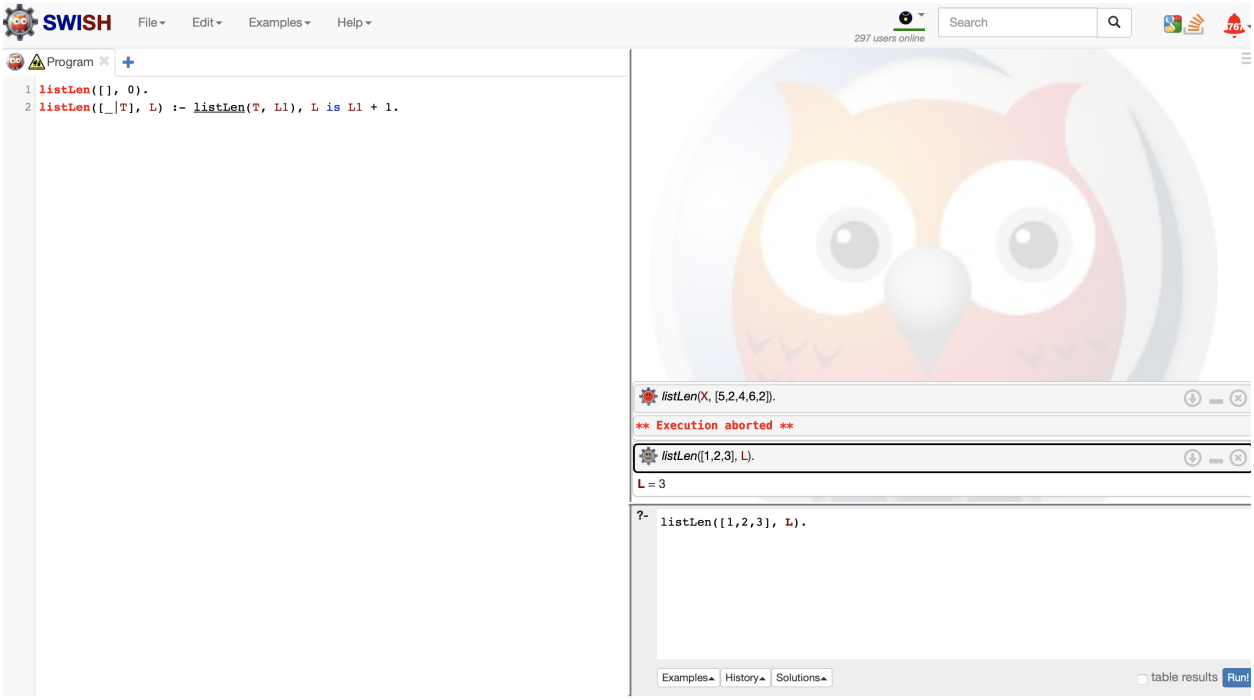
E.g. divide(X, Y, [3,1,9,5,8], 3).

Answer: X = [3, 1, 9],

Y = [5, 8]

```
divide([], [], L, _).
divide(L1, L2, [H|T], N) :- N > 0, N1 is N - 1, divide(L11, L2, T, N1), L1 = [H|L11].
divide(L1, L2, L, N) :- N =< 0, L1 = [], L2 = L.
```

```
X = Y, Y = []
X = [3],
Y = []
X = [3, 1],
Y = []
X = [3, 1, 9],
Y = []
X = [3, 1, 9],
Y = [5, 8]
```



Here is the combined Prolog code:

```
listLen([], 0).
listLen([_|T], L) :- listLen(T, L1), L is L1 + 1.
```

```
dropK(L, 1, [_|T]) :- L = T.
dropK([H|T], K, [H|L]) :- K > 1, K1 is K - 1, dropK(T, K1, L).

divide([], [], L, _).
divide(L1, L2, [H|T], N) :- N > 0, N1 is N - 1, divide(L11, L2, T, N1), L1 = [H|L11].
divide(L1, L2, L, N) :- N =< 0, L1 = [], L2 = L.

% Example calls
% listLen
% listLen([1, 2, 3], L).
% L = 3.

% dropK
% dropK(X, 2, [6,3,5,2,8]).
% X = [6,5,8].

% divide
% divide(X, Y, [], 0).
% X = Y, Y = []
% divide(X, Y, [3], 0).
% X = [3],
% Y = []
% divide(X, Y, [3, 1], 0).
% X = [3, 1],
% Y = []
% divide(X, Y, [3, 1, 9], 0).
% X = [3, 1, 9],
% Y = []
% divide(X, Y, [3, 1, 9, 5, 8], 3).
% X = [3, 1, 9],
% Y = [5, 8]
```