Задача 3. *Задачата да се реши на един от езиците Scheme или Haskell. По-долу оградете името на езика, който сте избрали за вашето решение.*

Даден е списък от списъци от числа **ll** и числова функция **f**. Числото **x** наричаме "корен" на **f**, ако **f(x) = 0**. Да се попълнят по подходящ начин празните полета по-долу, така че функцията **sumMaxRoots** да намира сумата на корените на **f** в този списък от **ll**, в който **f** има най-много корени. Ако има няколко такива списъка, функцията да връща сумата на корените в първия по ред списък. Ако функцията няма корен сред числата в списъците на **ll**, функцията да връща **0**.
Упътване: можете да използвате наготово функциите `apply`, `filter`, `foldr`, `length`, `map`, `max`, `maximum`, както и всички стандартни функции в R5RS за Scheme и Prelude за Haskell.

Scheme

```
(define (selectList l1 l2)
  (if ____________________________________________ l1 l2))

(define (sumMaxRoots f ll)
  (__________
      (__________ selectList __________
           (__________ (lambda (l)
                  (__________ (lambda (x) ____________________) l)) ll))))
```

Пример:
`(sumMaxRoots (lambda (x) (- (* x x x) x)) '((1 2 3) (-1 0 5) (1 4 -1))) → -1`

Haskell

```
selectList l1 l2 = if ____________________________________________ then l1 else l2

sumMaxRoots f ll =
  __________
      (__________ selectList __________
         (__________ (\l -> [ __________ | x <- l, ____________________ ]) ll))
```

Пример:
`sumMaxRoots (\x -> x^3 - x) [ [1, 2, 3], [-1, 0, 5], [1, 4, -1] ] → -1`