What should the value of

```
j-reduce(lam(x,y): x - y end, [join-list : 1, 2, 3])
```

be? Hint: There is more than one possible answer. List all.

Unfortunately, having more than one answer violates the expectation that `j-reduce` is a function. The problem is not with `j-reduce` but with the use of – (subtraction) as an argument. What property should the first argument to `j-reduce` demonstrate? Can you argue this from the description of `j-reduce` above?

The function could produce two answers, depending in what order operators are processed:

$$[\,(1-2)-3) = \ -4\,] \ \ or \ \ [\,(1-(2-3)) = 2\,]$$

The more elements in the list the greater the number of possible answers.

This is the case because with subtraction the order of operands matters---unlike with other operations like addition or multiplication. Operators with this property are said to be non-associative.

This is only an issue because of the lack of specificity in the function definition with regards to the order of processing operations. It could theoretically define any mathematically valid evaluative order as correct: evaluation from left to right; or from right to left; or branching from the centre… The point is there are lots of “right” definitions of what a correct output of j-reduce would be for a non-associative function. The function definition just has to explicitly state which one it is using.

An alternative approach would be limiting the domain of j-reduce to not include non-associative functions. This would, given the current function definition, effectively ensure each input maps to exactly one output.