List or vector reduction iteratively combines the first element of a vector with the second element of a vector, then that combined result is combined with the third element of the vector, and so on until the end of the vector is reached. The function to be applied should take at least two arguments. Where mapping returns a vector or a list, reducing should return a single value. Some examples usingreduce() are illustrated below:

```
reduce(c(1, 3, 5, 7), function(x, y){
  message("x is ", x)
  message("y is ", y)
  message("")
  x + y
})
x is 1
y is 3

x is 4
y is 5

x is 9
y is 7

[1] 16
```

On the first iteration x has the value 1 and y has the value 3, then the two values are combined (they're added together). On the second iteration x has the value of the result from the first iteration (4) and y has the value of the third element in the provided numeric vector (5). This process is repeated for each iteration. Here's a similar example using string data:

```
reduce(letters[1:4], function(x, y){
  message("x is ", x)
  message("y is ", y)
  message("")
  paste0(x, y)
})
x is a
y is b

x is ab
y is c

x is abc
y is d

[1] "abcd"
```

By default reduce() starts with the first element of a vector and then the second element and so on. In contrast the reduce_right()function starts with the last element of a vector and then proceeds to the second to last element of a vector and so on:

```
reduce_right(letters[1:4], function(x, y){
  message("x is ", x)
  message("y is ", y)
  message("")
  paste0(x, y)
})
x is d
y is c

x is dc
y is b

x is dcb
y is a

[1] "dcba"
```