

Problem Wk.3.3.1: Map

Part 1: mapList

Define a procedure `mapList` that takes two arguments, a procedure of one argument and a list. It returns the list of the results of applying the procedure to each of the elements of the list.

```
>>> def sq(x): return x*x
>>> mapList(sq, [1,2,3,4])
[1, 4, 9, 16]
```

You must use a list comprehension.

Part 2: sumAbs

Use `mapList` to define a procedure called `sumAbs` that given a list of numbers, returns the sum of the absolute values of the numbers.

Your procedure must use `mapList`. You should be aware of the `sum` and `abs` built in functions in Python.

Part 3: mapSquare

Define a procedure `mapSquare` that takes two arguments, a procedure of two arguments and a list. It returns a list of lists of all the results of applying the procedure to all combinations of the values in the list.

```
>>> def diff(x, y): return x - y
>>> mapSquare(diff, [1,2,3])
[[0, -1, -2], [1, 0, -1], [2, 1, 0]]
```

Note that this list is:

```
[[1-1, 1-2, 1-3], [2-1, 2-2, 2-3], [3-1, 3-2, 3-3]]
```

You must use a list comprehension. Hint: Think about using nested list comprehensions.



MIT OpenCourseWare
<http://ocw.mit.edu>

6.01SC Introduction to Electrical Engineering and Computer Science
Spring 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.