CISC280 Spring 2007 Lab 3

In class we wrote procedures **accumulate** and **enumerate**. You will use them in this lab. We will also use the built-in **map** function (not the one we wrote in class) and **apply**.

Write the function **assert** that takes four arguments: a test number, a function, a list of arguments, and an answer. It should display the test number, report pass or fail, then return true for pass and false for fail:

```
> (assert 23 (lambda (x) (* x x)) '(3) 9)
test 23 PASS
#t
```

To write **assert**, you may use the **apply** function, which takes a single procedure and applies it to arguments passed as a list:

```
> (apply + (enumerate 1 4))
10
> (apply (lambda (x) (* x x)) (list 3)) ;NOTE that 3 is in a list
9
```

Recall that the built-in map function takes a function and lists of arguments, as follows:

```
> (map * '(1 2 3) '(4 5 6) '(7 8 9))
(28 80 162)
>
```

You could use map to generate code testing results for some function f by having map call your assert function:

```
(map assert list-of-nums list-of-f-functions lists-of-parameters list-of-answers)
```

Try this. Then put assert and f inside a lambda so you don't need the list of f functions:

```
(map lambda-fcn list-of-nums lists-of-parameters list-of-answers)
```

Now use accumulate to combinie all the results and return a single true or false.

If you need to submit (see syllabus if you are unsure) submit your code file(s) and a script of several well-chosen test cases via MyCourses (due Thursday midnight) and on paper (to your TA's mailbox Friday by 1 p.m.) to receive full credit.

When you use MyCourses, remember that you can "upload" files multiple times, but you only click "submit" once.