MASSACHVSETTS INSTITVTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.01—Introduction to EECS I
Fall Semester, 2007

NanoQuiz #1 (R 13 Sep)

Solutions

Name: _____ **Athena userid:** _____`@mit.edu`

1. Write a procedure `comparer` that takes two procedures `f1` and `f2` (each of which takes a single argument `x`) as input and returns a new procedure that takes a single argument `x`. The new procedure should return -1, 1, or 0 if `f1(x)` is respectively less than, greater than, or equal to `f2(x)`.

```
def comparer(f1, f2):
  def comp (x):
    r1 = f1(x)
    r2 = f2(x)
    if ( r1 < r2 ):
      return -1
    elif ( r1 > r2 ):
      return 1
    else:
      return 0
  return comp
```

Note that you cannot return a lambda, since that is limited to a single expression in Python (as opposed to Scheme).

Several people wanted to end the function with
return comp(x)
but there is no x known when the function is defined

2. Write a procedure `sqSome` that takes a list of numbers as input and returns a list of the squares of those elements that are greater than 5, so that (for example)

   `sqSome([0, 5, 6, 9])` ⇒ [36, 81]

   and

   `sqSome([1, 8, 5, 2, 7, 3, 10])` ⇒ [64, 49, 100].

   Use a list comprehension.

```
def sqSome(nums):
  return [x*x for x in nums if x > 5]
```