6.00 Introduction to Computer Science and Programming
Fall 2008

6.00 Quiz 1 Practice Problem Solutions

6.00 Staff

October 2, 2008

Problem 1

1. False. Recursion means your program can run indefinitely.
2. False. You may end up jumping back and forth between the same two forever, given an S-shaped function (draw a diagram).
3. False. `mydict[somekey] = somevalue`.
4. False. Precision is finite.
5. False. Recursion may be a more natural way to express certain problems (e.g.: fib, Towers of Hanoi).
6. True. Code reuse.
7. True. A quick lookup.

Problem 2

1. Yes, they return the same value for all possible inputs (at least of the types that we've learned about so far in class).
2. No, they print different things for negative inputs. This is because `a` and `b` are updated to refer to a different number in `compare1`, whereas they are not updated in `compare2`.

Problem 3

Note about this function: it is a bit strange in that it handles multiple argument types.

1. `f(2112)` returns `2+1+f('12') ==> 2+1+1+2 ==> 6`.
2. Given an integer or a string representation of an integer, `f` returns the sum of its digits.

Problem 4

```
def first_N(n):
  count = 0
  current_sqrt = 1
  while count < n:
    square = current_sqrt * current_sqrt
    # If square is not even
    if square % 2 != 0:
      print square
      count += 1
    current_sqrt += 1
```

Problem 5

```
def guess_and_check(criteria):
  for a in range(...):
    for b in range(...):
      for c in range(...):
        ...
          if satisfies_criteria(a, b, c, ..., criteria):
            return a, b, c,...
```

Problem 6

```
def findSide():
  area = float( raw_input('Enter the area of the rectangle: ') )
  side1 = float( raw_input('Enter the length of one side of the rectangle: ') )
  return area / side1
```

Problem 7

Yes, it meets its specification, because the list being modified is a brand-new list (`result`) that is created inside the function, then returned. `L` is only traversed.

Problem 8

Note the resemblance to the exhaustive enumeration for guess-and-check, in Problem 5.

We're assuming that by "decides," we just need to return `True`/`False`.

```
def nuggets(num):
  for a in range(num/6+1):
    for b in range(num/9+1):
      for c in range(num/20+1):
        if 6*a + 9*b + 20*c == num:
          return True
  return False
```

Problem 9

Given an integer, take the string representation of that integer and reverse its digits (returning this as a string).