

Sample Quiz Questions, Quiz 1

Sample Quiz Questions

Here some sample quiz questions. There are not intended to provide comprehensive coverage of the material covered thus far in 6.00. However, they should give you a sense of the kinds of questions that will be on the quiz. The actual quiz will not be as long as this set of sample questions.

This quiz is **open book and open notes, but do not use a computer.**

1) Are each of the following True or False?

Newton's method will always converge on a correct root of a function.

In Python, lists are immutable.

One should always avoid iteration when a recursive solution is possible.

In Python, retrieving the value associated with a dictionary key takes roughly constant time.

There exist problems that **cannot** be solved in Python **without** using recursive functions.

In Python, all elements in a list must be of the **same** type.

In Python, retrieving the value associated with a dictionary key takes time that is roughly **linear** in the size of the dictionary.

It is **not** a good idea to use `==` when comparing floating point numbers.

There are problems that can be solved using a bisection method that **cannot** be solved using Newton-Raphson.

Given the same input, a program that is $O(n^2)$ will **always** take longer to run than a program that is $O(n)$.

Sample Quiz Questions, Quiz 1

2) Consider the implementations of `compare1` and `compare2`, where `a` and `b` are floats.

2.1) Do `compare1` and `compare2` return the same value for all possible inputs? If not, give a pair of inputs for which they return a different value.

2.2) Do `compare1` and `compare2` print the same thing for all possible inputs? If not, give a pair of inputs for which they print different things.

```
def compare1(a, b):
    if a < 0:
        a = -a
    if b < 0:
        b = -b
    res = (a == b)
    if res:
        print a, 'and', b, 'have the same absolute value.'
    else:
        print a, 'and', b, 'have different absolute values.'
    return res
```

```
-----
def absolute_value(n):
    if n < 0:
        n = -n
    return n
```

```
def compare2(a, b):
    res = absolute_value(a) == absolute_value(b)
    if res:
        print a, 'and', b, 'have the same absolute value.'
    else:
        print a, 'and', b, 'have different absolute values.'
    return res
```

Sample Quiz Questions, Quiz 1

3) Consider the following implementation of a function f , where x is a positive integer:

```
def f(x):
    xs = str(x)
    if len(xs) == 1:
        return int(xs)
    n = int(xs[0]) + int(xs[1])
    if len(xs) == 2:
        return n
    else:
        return n + f(xs[2:])
```

What does $f(2112)$ return?

3.2. Write a specification of f .

4) Provide a Python implementation of a function `first_N` that takes a positive integer, n , as its only argument. The function should print the first n perfect squares that are **not** even numbers. E.g., if n were 2 it should print the perfect squares 1 and 9.

5) Provide a Python implementation for the function `findSide` specified below

```
def findSide():
    """asks the user to enter the area of a rectangle and the
    length of one side of the rectangle. Returns a floating point
    number that is the length of the adjacent side."""
```

6) Does the following function meet its specification? If not, change the program so that it is consistent with the specification.

```
def f(L):
    """Returns a copy of the list L without modifying L."""
    result = []
    for e in L: result.append(e)
    return result
```

7) At McDonalds one can buy chicken nuggets in packages containing 6, 9 or 20 pieces. Write a Python function that accepts an integer, `num`, as an argument and decides whether or not it is possible to buy `num` nuggets at McDonalds.

Sample Quiz Questions, Quiz 1

8) Write an appropriate specification for the function below. Assume that n is an integer.

```
def f(n):
    s = str(n)
    if len(s) <= 1: return s
    return s[-1] + f(int(s[:-1]))
```

9) What does the following code print?

```
def f(L):
    result = []
    for e in L:
        if type(e) == list:
            return f(e)
        else:
            result.append(e)
    return result

print f([1, 2, 3, 4, 5])
```

10) Provide a Python implementation for the function `findPS` specified below. Do not bother copying the specification. (20 points)

```
def findPS(x):
    """Assumes that x is an int.
    Returns the longest substring of digits in x
    that is a perfect square. E.g., for 123, it returns
    the string '1'. For 20645, it returns '064'. (Take
    note of the leading zero.) If there is more than one
    longest substring it may return any of them. If there is
    no such substring, it returns the empty string."""
```

You may assume access to the already implemented functions:

```
def allSubstrs(s):
    """Assumes that s is a string. Returns a list containing all
    of the substrings of s. The order in which the substrings
    appear in the list is not defined. E.g., for '123' it might
    return ['23', '1', '2', '3', '12', '123']. """

def isPS(x):
    """Assumes x is an int.
    Returns True if x is a perfect square; False otherwise."""
```

Sample Quiz Questions, Quiz 1

11) Consider the following code. Hint: It is similar, but not identical, to code from one of the problem sets.

```
import random

VOWELS = 'aeiou'
LETTERS = 'abcdefghijklmnopqrstuvwxyz'

def deal_hand(n, nv):
    hand={}
    num_vowels = n / nv
    for i in range(num_vowels):
        x = VOWELS[random.randrange(0, len(VOWELS))]
        hand[x] = hand.get(x, 0) + 1
    for i in range(num_vowels, n):
        x = LETTERS[random.randrange(0, len(LETTERS))]
        hand[x] = hand.get(x, 0) + 1
    return hand
```

11.1) Is the value {'a': 3, 'k': 1, 'u': 2} something that `deal_hand(6, 2)` might return? (7 points)

11.2) Will `deal_hand` run successfully and return a dictionary for all possible inputs? Why or why not? (6 points)

11.3) What is the order, use big O notation, of `deal_hand`? Assume that `random.randrange` is $O(1)$. (5 points)