CS:1210 Quiz 4 Version (a)

You have 15 minutes to complete this quiz. Please put away your books, notes, and all electronic devices

1. Here is a partially completed program that simulates 1 million rolls of three six-sided dice and reports the number of times *exactly* two out of the three dice rolls were identical. For example, if the first die rolled 3, the second rolled 4, and the third rolled 3, we would count this event. However, if all three dice rolled 3, we would not count this event. Also, if the first die rolled 3, the second rolled 4 and the third rolled 5, we would not count this event because all three dice rolls are distinct.

   The code below is complete except for two boolean expressions that are missing from `if`-statements. Your task is to understand our code and supply the missing pieces. Please read our comments carefully – they provide key hints.

```
import random

counter = 0 # tracks the number of rolls

# numTimes tracks the number of times exactly two
# of the three rolls are identical
numTimes = 0

# roll three six-sided dice million times
while counter < 1000000:
    roll1 = random.randint(1, 6)
    roll2 = random.randint(1, 6)
    roll3 = random.randint(1, 6)

    # First check if at least one pair of the dice rolls
    # have identical values; this allows either two or all three
    # dice rolls to have identical values.

    if ___________________________________________________________:

        # Then check to make sure that not all three rolls are
        # identical

        if _______________________________________________________:

            numTimes = numTimes + 1

    counter = counter + 1

print(numTimes)
```

2. Suppose that we have already defined a boolean function called `isPrime` that takes as argument a positive integer, say n, and returns `True` if n is a prime and returns False otherwise. Now define a function called `numPrimes` that takes two positive integer arguments, say m and n, such that $m \leq n$. The function `numPrimes` should return the number of prime numbers in the range $[m, n]$. For example, the function call `numPrimes(24, 28)` should return `0` but `numPrimes(24, 30)` should return `1`, since there is only one prime (i.e., 29) in the range $[24, 30]$. Your definition of `numPrimes` should repeatedly call the function `isPrime`.
   Note: You don't have to define `isPrime`, only `numPrimes`.