Sample questions

Recursion Quiz

1. Trace the following method's execution if the initial call is `mystery(30)`.

```java
int mystery(int n) {
    if(n==0)
        return 0;
    if(n%2==1)
        return n + mystery(n/2);
    return mystery(n/2);
}
```

2. Complete the following method that should return the sum of all even integers from 0 to `n` (inclusive on both sides). Assume `n` is greater than or equal to 0.

```java
int sumEven(int n) {
    if(n==0)
        return ________;
    if(n%2==1)
        return sumEven(n-1);
    return ________ + sumEven(__________);
```

3. How many times is the function `foo` called including the initial call, if the initial call is `foo(1)`?

```java
int foo(int n) {
    if(n > 100)
        return 0;
    if(n%2 == 1)
        return foo(2*n+1);
    return foo(n*4);
}
```

4. If the initial call is `foo(6, 0, 1)`, at some stage `foo(4, 1, 2)` is called. What is the function call made by `foo(4, 1, 2)`? As in `foo(____, _____, _____)`.

```java
int foo(int a, int b, int c) {
    if(a==0)
        return b;
    if(a==1)
        return c;
    return foo(a-1, c, b+c);
}
```

5. Complete the following function that returns the reverse of the String passed.

```java
String reverse(String s) {
    if(s==null || str.length() < 2)
        return s;
    char x = str.charAt(_______);
    char y = str.substring(_____________________);
    return reverse(y) + x;
}
```

6. Tail-optimize the following recursive function:

```java
/**
* @param n
* @return number of even digits in the integer n
* for example,
* countEvenDigits(1724983) = 3
* countEvenDigits(-483) = 2
* countEvenDigits(0) = 0 (by itself 0 has no significant digits)
*/
int countEvenDigits(int n) {
    if(n < 0) {
        return countEvenDigits(-n);
    }
    if(n == 0) {
        return 0;
    }
    if(n%2 == 0) {
        return 1 + countEvenDigits(n/10);
    }

    return countEvenDigits(n/10);
}
```

7. Complete the following function that returns a String with all occurrences of `source` replaced with `dest` in the String `str`:

```java
String replace(String str, String source, String dest) {
    if(str == ___________) {
        return str;
    }
    if(str.length() < ___________________) {
        return str;
    }
    String init = str.substring(source.length());
    if(init.equals(source)) {
        return __________ + replace(str.substring(source.length(), source, dest);
    }
    return __________ + replace(str.substring(1), source, dest);
}
```

8. Trace the execution of the function call `mystery(13)`.

```java
String mystery(int n) {
    if(n == 0) {
        return "";
    }
    return mystery(n/2) + n%2;
}
```

caller calls `mystery(13)`

calls `mystery(____)`

which calls `mystery(____)`

which calls `mystery(____)`

which calls `mystery(0)`

`mystery(0)` returns “” to `mystery(____)`

which returns "_____" to `mystery(____)`

which returns "_____" to `mystery(____)`

which returns "_____" to `mystery(____)`

which returns "_____" to `mystery(13)`

which returns "_____" to the caller