AP Comp Sci, Quiz 04 Review

(1) How many String Objects have been created? two
How many Object Reference variables have been created? three
What EXACTLY would print below?

```
public static void main (String[] args)
{
        String Str1 = new String("first string");  } String Objects
Obj refs -->  String Str2 = "first string";
        String Str3 = Str1;

        System.out.println("Str1 == Str2?  " + (Str1 == Str2));
        System.out.println("Str1 == Str3?  " + (Str1 == Str3)); <-- Both have
        System.out.println("Str1.equals(Str2)?  " +                  same address...
        Str1.equals(Str2));
        System.out.println("Str1.equals(Str3)?  " +
        Str1.equals(Str3));

        //OTHER CODE NOT SHOWN
}
```

```
Process started >>>
Str1 == Str2?  false
Str1 == Str3?  true
Str1.equals(Str2)?  true
Str1.equals(Str3)?  true
<<< Process finished. (Exit code 0)
```

(2) Write a recursive method that calculates Powers.
For example $2^3 = 8$, $4^2 = 16$, etc.

Remember to ask yourself what is the simpler version of this problem. So 2^3 power is also 2 * 2^2 power.

```
//precondition:  base and exponent are both >= zero
//precondition:  base to the exponent power <= INT_MAX
//postcondition: base to the exponent power is returned
public int myPower(int base, int exponent)
{

    ONE POSSIBLE SOLUTION:
    ======================

    int answer = 1;

    if(exponent > 0)
    {
            answer = base * myPower(base, exponent - 1);
    }

    return answer;

}
```

(3) Carefully analyze the Java method provided in the box below (which uses recursion) and figure out the missing RUN OUTPUT. Do not change the Java code in any way. Fill in the missing RUN OUTPUT at the bottom of the page.

```java
public class RecursionTester
{

        public static void main(String args[])
        {
                System.out.println();
                System.out.print("When given \"racecar\", the method returns --> ");
                System.out.println(mystery("racecar"));
                System.out.println();
        }

        private static String mystery(String Test)
        {
                int length;
                String Result = "";
                String First;
                String Middle;
                String Last;

                length = Test.length();

                if (length > 1)
                {
                        First =  Test.substring(0, 1);
                        Middle = Test.substring(1, length - 1);
                        Last =   "" + Test.charAt(length - 1);
                        Result = First + Last + mystery(Middle);
                }

                return Result;
        }
}
```

Notice which String methods are being used, and how...

<-- notice the order that result is put back together, and notice the recursion!

```
/**
RUN OUTPUT:

Process started >>>

When given "racecar", the mystery method returns -->

<<< Process finished.

*/
```

rraacc

(4) The "end punctuation" for a sentence in English can be either a period [.], exclamation point [!], or question mark [?]. Complete the method properEndPunctuation below which determines if Sentence has proper end punctuation. When using String class methods, use only the String methods provided in the box below.

Standard Java 8 API String methods (https://docs.oracle.com/javase/8/docs/api/java/lang/String.html):

You must use ONLY these String methods!!

int length()
Returns the length of this string.
char charAt(int index)
Returns the char value at the specified index.
String trim()
Returns a string whose value is this string, with any leading and trailing whitespace removed.

```
//Preconditions:        Sentence is complete, may need spaces removed from both ends,
//                      and may or may not end with proper punctuation after trimming.
//Postconditions:       return true if Sentence ends with proper punctuation
//                      return false if Sentence doesn't end with proper punctuation
private boolean properEndPunctuation(String Sentence)
{
```

ONE POSSIBLE SOLUTION:
======================

```
    char punctuation;
    int length;

    Sentence = Sentence.trim();
    length = Sentence.length();
    punctuation = Sentence.charAt(length - 1);

    if(punctuation == '.' || punctuation == '!' || punctuation == '?')
    {
            return true;
    }
    else
    {
            return false;
    }

}
```