

Recursion Workshop - Reverse a String

1. Create a static ReverseString class that reverses a string. The static class has a public method Reverse that accepts a string and an implementation option.

```
public static class ReverseString
{
    public static string Reverse(string str, string impl="recursive")
    {
        if (impl != "recursive" && impl != "iterative") {
            throw new Exception("Unknown implementation type.");
        }

        if (str.Length == 0) {
            return "";
        }
        return (impl == "recursive") ? RReverse(str) : IReverse(str);
    }

    private static string IReverse(string str)
    {
        return "";
    }

    private static string RReverse(string str)
    {
        return "";
    }
}
```

2. Provide implementations for the private methods IReverse and RReverse. IReverse reverse a string iteratively (using a for loop), while RReverse does it recursively.

3. Use the following Main program to test your code.

```
class Program
{
    static void Main(string[] args)
    {
        string impl = "iterative";

        Reverse("hello world");
        Reverse("hello world", impl);

        Reverse("apple");
        Reverse("apple", impl);

        Reverse("toy");
        Reverse("toy", impl);

        Reverse("");
        Reverse("", impl);
    }

    static void Reverse(string str, string impl = "recursive")
    {
        Console.WriteLine("'{0}' reversed is '{1}'",
            str, ReverseString.Reverse(str, impl));
    }
}
```

4. The output should look like this.

```
'hello world' reversed is 'dlrow olleh'
'hello world' reversed is 'dlrow olleh'
'apple' reversed is 'elppa'
'apple' reversed is 'elppa'
'toy' reversed is 'yot'
'toy' reversed is 'yot'
'' reversed is ''
'' reversed is ''
```