

PART A:

```
public String decodeString(ArrayList<StringPart> parts)
{
    String expanded = "";
    for (StringPart nextPart : parts)
    {
        int ending = nextPart.getStart()+nextPart.getLength();
        expanded += masterString.substring(nextPart.getStart(), ending);
    }
    return expanded;
}
```

PART B:

```
public ArrayList<StringPart> encodeString(String word)
{
    ArrayList<StringPart> parts = new ArrayList<StringPart>();

    while (word.length() > 0)
    {
        StringPart nextPart = findPart(word);
        parts.add(nextPart);
        word = word.substring(nextPart.getLength());
    }
    return parts;
}
```

ALTERNATE SOLUTION:

```
public ArrayList<StringPart> encodeString(String word)
{
    ArrayList<StringPart> parts = new ArrayList<StringPart>();

    int index = 0;
    while (index < word.length())
    {
        StringPart nextPart = findPart(word.substring(index));
        parts.add(nextPart);
        index += nextPart.getLength();
    }
    return parts;
}
```

Part B:	encodeString	4 1/2 points
----------------	--------------	---------------------

- +1/2 construct an `ArrayList<StringPart>` (must assign to a variable, generic okay)
- +3 1/2 find, collect string parts, and build list (in context of loop)
 - +1 `findPart(X)`, where X is word or a substring of word
 - +1 calls to `findPart` involve progressively smaller suffixes of word
 - +1/2 add found string part to `ArrayList` of string parts
 - +1 build correct list of string parts (must have used `findPart`)
- +1/2 return `ArrayList` of string parts

Part A:	decodeString	4 1/2 points
----------------	--------------	---------------------

- +1 traverse `parts`
 - +1/2 correctly access an element of `parts` (in context of loop)
 - +1/2 access all elements of `parts` (lose this if index out-of-bounds)
- +2 retrieve substrings from `masterString`
 - +1/2 correctly call `getStart()` and `getLength()` on accessed part
 - +1 1/2 extract a substring from `masterString`
 - +1/2 `masterString.substring(X, Y)`
 - +1 extract correct substring
- +1 1/2 build and return decoded string
 - +1 correctly build string from substrings of `masterString`
 - +1/2 return built string