

0. Given a string, return a version where the last 3 characters are moved to the beginning. The string length will be at least 3.

```
public String right3(String str) {  
    if (str.length() == 3)  
    {  
        return str;  
    }  
    String end = str.substring(str.length() - 3);  
    String beg = str.substring(0, str.length() - 3);  
    return end + beg;  
  
}
```

1. Given a string, return true if it ends in "ous". The string may be of any length, including zero.

```
public boolean endsOus(String str) {  
    if (str.length() < 3)  
    {  
        return false;  
    }  
    if (str.substring(str.length() - 3).equals("ous"))  
    {  
        return true;  
    }  
    return false;  
  
}
```

2. Given a string, return a string length 3 made of its first 3 chars. If the string length is less than 3, use '!' for the missing chars.

```
public String atFirst(String str) {  
    if(str.length()==0)  
        return "!!!";  
    if(str.length()==1)  
        return str+"!!";  
    if(str.length()==2)  
        return str+"!";  
    if(str.length()==3)  
        return str;  
    return str.substring(0,3);  
}
```

3. Given a string, return a new string made of 3 copies of the first 3 chars of the original string. The string may be any length. If there are fewer than 3 chars, use whatever is there.

extraFront("Hello") → "HelHelHel"

extraFront("ab") → "ababab"

extraFront("H") → "HHH"

```
public String thriceFront(String str) {  
    String f;  
    if (str.length()>=3)  
        f = str.substring(0,3);  
    else  
        f = str;  
    return f + f + f;  
}
```