# Java字符串专题练习（一）

在线测试地址：

<http://222.18.75.38/JudgeOnline/contest.php?cid=1137>

请结合第六章例6-1~例6-6进行练习。

## 替换空格

**题目描述：**

请实现一个函数，将一个字符串中的空格替换成“%20”。例如，当字符串为We Are Happy.时则经过替换之后的字符串为We%20Are%20Happy。

**输入：**

每个输入文件仅包含一组测试样例。  
对于每组测试案例，输入一行代表要处理的字符串。

**输出：**

对应每个测试案例，输出经过处理后的字符串。

**样例输入：**

We Are Happy

**样例输出：**

We%20Are%20Happy

【分析】

可以直接使用Java的String类的replace方法或replaceAll方法替换。这2个方法的声明如下：

public [String](mk:@MSITStore:D:\java\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **replace**([CharSequence](mk:@MSITStore:D:\\java\\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html" \o "java.lang 中的接口) target,

[CharSequence](mk:@MSITStore:D:\java\JDK_API_1_6_zh_CN.CHM::/java/lang/CharSequence.html) replacement)

使用指定的字面值替换序列替换此字符串所有匹配字面值目标序列的子字符串。该替换从字符串的开头朝末尾执行，例如，用 "b" 替换字符串 "aaa" 中的 "aa" 将生成 "ba" 而不是 "ab"。

**参数：**

target - 要被替换的 char 值序列

replacement - char 值的替换序列

**返回：**

所得 String

public [String](mk:@MSITStore:D:\java\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **replaceAll**([String](mk:@MSITStore:D:\java\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) regex,

[String](mk:@MSITStore:D:\java\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) replacement)

使用给定的 replacement 替换此字符串所有匹配给定的[正则表达式](mk:@MSITStore:D:\\java\\JDK_API_1_6_zh_CN.CHM::/java/util/regex/Pattern.html" \l "sum)的子字符串。

**参数：**

regex - 用来匹配此字符串的正则表达式

replacement - 用来替换每个匹配项的字符串

**返回：**

所得 String

## 左旋转字符串

**题目描述：**

汇编语言中有一种移位指令叫做循环左移（ROL），现在有个简单的任务，就是用字符串模拟这个指令的运算结果。对于一个给定的字符序列S，请你把其循环左移K位后的序列输出。例如，字符序列S=”abcXYZdef”,要求输出循环左移3位后的结果，即“XYZdefabc”。是不是很简单？OK，搞定它！

**输入：**

多组测试数据，每个测试数据包含一个字符序列S和非负整数K。其中S的长度不超过1000。

**输出：**

对应每个测试案例，输出新序列。

**样例输入：**

UDBOJ 4

abba 1

**样例输出：**

JUDBO

bbaa

【分析】：此题可以利用Java中的String类的substring方法实现。

对于输入UDBOJ 4来说，先截取左边4个字符UDBO，再截取剩下的那个字符J，再将2部分利用concat方法连接起来即可。

public [String](mk:@MSITStore:D:\java\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **substring**(int beginIndex)

返回一个新的字符串，它是此字符串的一个子字符串。该子字符串从指定索引处的字符开始，直到此字符串末尾。

示例：

"unhappy".substring(2) 将返回 "happy"

"Harbison".substring(3) 将返回"bison"

"emptiness".substring(9) 将返回"" (an empty string)

**参数：**

beginIndex - 起始索引（包括）。

**返回：**

指定的子字符串。

public [String](mk:@MSITStore:D:\java\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **substring**(int beginIndex,int endIndex)

返回一个新字符串，它是此字符串的一个子字符串。该子字符串从指定的 beginIndex 处开始，直到索引 endIndex - 1 处的字符。因此，该子字符串的长度为 endIndex-beginIndex。

示例：

"hamburger".substring(4, 8) 将返回 "urge"

"smiles".substring(1, 5) 将返回"mile"

**参数：**

beginIndex - 起始索引（包括）。

endIndex - 结束索引（不包括）。

**返回：**

指定的子字符串。

public [String](mk:@MSITStore:D:\java\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) **concat**([String](mk:@MSITStore:D:\java\JDK_API_1_6_zh_CN.CHM::/java/lang/String.html) str)

将指定字符串连接到此字符串的结尾。

如果参数字符串的长度为 0，则返回此 String 对象。否则，创建一个新的 String 对象，用来表示由此 String 对象表示的字符序列和参数字符串表示的字符序列连接而成的字符序列。

示例：

"cares".concat("s") 将返回 "caress"

"to".concat("get").concat("her") 将返回 "together"

## 翻转字符串

给定一个字符串，逐个翻转字符串中的每个单词。

**样例**

给出s = **"the sky is blue"**，返回**"blue is sky the"**

**说明**

* 单词的构成：无空格字母构成一个单词
* 输入字符串是否包括前导或者尾随空格？可以包括，但是反转后的字符不能包括
* 如何处理两个单词间的多个空格？在反转字符串中间空格减少到只含一个

【分析】：首先使用String类的trim方法将首尾空格去掉。然后利用split方法使用**正则规则[\\s]+**拆分字符串到一个String数组，最后再从尾到头遍历String数组的每一个元素，将每个元素连接起来，连接时保证单词之间以一个空格分隔即可。连接时可以利用String类的concat方法。

说明：正则规则**[\\s]+**表示**一个**或**多个空格**。使用示例：split(“[\\s]+”)

## 比较字符串

比较两个字符串A和B，确定A中是否包含B中所有的字符。字符串A和B中的字符都是 **大写字母**

**样例**

给出 A = "ABCD" B = "ACD"，返回 true

给出 A = "ABCD" B = "AABC"， 返回 false

**注意**

在 A 中出现的 B 字符串里的字符不需要连续或者有序。

【分析】：

由于都是大写字母，因此可以定义2个int型数组fa和fb，长度都是26，每个元素初始值都为0。扫描字符串A，统计每个字符出现的频率，存入数组fa。再扫描字符串B，统计每个字符出现的频率，存入数组fb。然后扫描数组fb，对于fb数组的每个不为0的元素值，其值都应该小于等于fa数组对应位置的元素值，此时可以确定A中包含B中所有的字符。否则不是。