

Homework 2 DDL: Oct.26 23:00

1. KMP(50%)

通过实现KMP算法，给定一个文本 `S` 和模式 `P`，得出 `P` 出现在 `S` 中多少次，以及出现的位置。

要求：

1. 计算出模式 `P` 的 `next` 数组。

实现 `computeNextArray` 接口

```
1 public static int[] computeNextArray(String pattern) {
2     /* TODO: YOUR CODE HERE */
3 }
```

样例输入：

```
1 int[] next;
2 next = computeNextArray("touristrealgod");
3 System.out.println(Arrays.toString(next));
4 next = computeNextArray("asardasd");
5 System.out.println(Arrays.toString(next));
```

样例输出：

```
1 [-1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
2 [-1, 0, 0, 1, 0, 0, 1, 2]
```

2. 计算 `P` 出现在 `S` 中多少次。

实现 `KMPSearchTimes` 接口：

```
1 public static int KMPSearchTimes(String text, String pattern) {
2     /*
3     TODO: YOUR CODE HERE
4     利用第一问实现的next数组
5     */
6 }
```

样例输入：

```
1 String P,S;
2 P = "wo";
3 S = "chenljbwowowoo";
4 System.out.println(KMPSearchTimes(S,P));
5 P = "tourist";
6 S = "touristrealgod";
7 System.out.println(KMPSearchTimes(S,P));
```

样例输出：

```
1 | 3
2 | 1
```

3. 计算 `P` 出现在 `S` 中的位置。

实现 `KMPFindLocations` 接口：

```
1 | public static LinkedList<Integer> KMPFindLocations(String text, String
2 | pattern) {
3 |     /*
4 |     TODO: YOUR CODE HERE
5 |     利用第一问实现的next数组
6 |     */
7 | }
```

样例输入：

```
1 | String P,S;
2 | P = "wo";
3 | S = "chenljnbwowowoo";
4 | System.out.println(KMPFindLocations(S,P));
5 | P = "tourist";
6 | S = "touristrealgod";
7 | System.out.println(KMPFindLocations(S,P));
```

样例输出：

```
1 | [8, 10, 12]
2 | [0]
```

2. 字符串运算(50%)

给定两个字符串形式的16进制整数 `num1` 和 `num2` 。

注意：

1. `num1` 和 `num2` 都只包含 0-9A-F，同时可包含+-符号（只考虑出现在首位，也可不出现）；
2. `num1` 和 `num2` 均不以零开头，除非是数字 0 本身；
3. 不能使用任何内建 **BigInteger** 库，也不能直接将输入的字符串转换为整数形式进行处理。
4. 输入的单个字符串长度不超过200位

1. 计算它们的和，结果也为字符串形式。

`addStrings` 接口：

```
1 public static String addStrings(String num1, String num2){
2     /*
3     TODO: YOUR CODE HERE
4     */
5 }
```

样例输入:

```
1 String num1, num2;
2 num1 = "2";
3 num2 = "3";
4 System.out.println(addStrings(num1,num2));
5 num1 = "123";
6 num2 = "456";
7 System.out.println(addStrings(num1,num2));
```

样例输出:

```
1 5
2 579
```

2. 计算乘积，结果也为字符串形式。

multiply 接口:

```
1 public static String multiply(String num1, String num2){
2     /*
3     TODO: YOUR CODE HERE
4     */
5 }
```

样例输入:

```
1 String num1, num2;
2 num1 = "2";
3 num2 = "3";
4 System.out.println(multiply(num1,num2));
5 num1 = "123";
6 num2 = "456";
7 System.out.println(multiply(num1,num2));
```

样例输出:

```
1 6
2 56088
```