

罗马数字包含以下七种字符:I, V, X, L, C, D 和 M。

| 字符 | 数值 |
|----|------|
| I | 1 |
| V | 5 |
| X | 10 |
| L | 50 |
| C | 100 |
| D | 500 |
| M | 1000 |

例如，罗马数字 2 写做 II，即为两个并列的 1。12 写做 XII，即为 X + II。27 写做 XXVII，即为 XX + V + II。

通常情况下，罗马数字中小的数字在大的数字的右边。但也存在特例，例如 4 不写做 IIII，而是 IV。数字 1 在数字 5 的左边，所表示的数等于大数 5 减小数 1 得到的数值 4。同样地，数字 9 表示为 IX。这个特殊的规则只适用于以下六种情况：

I 可以放在 V (5) 和 X (10) 的左边，来表示 4 和 9。

X 可以放在 L (50) 和 C (100) 的左边，来表示 40 和 90。

C 可以放在 D (500) 和 M (1000) 的左边，来表示 400 和 900。

给定一个罗马数字，将其转换成整数。输入确保在 1 到 3999 的范围内。

```
public int romanToInt (String s) {
    char[] chars = s.toCharArray();
    int sum = 0;
    Map<String,Integer> map = new HashMap<>();
    map.put("I",1);
    map.put("V",5);
    map.put("X",10);
    map.put("L",50);
    map.put("C",100);
    map.put("D",500);
    map.put("M",1000);

    for (int i = 0; i < chars.length; i++) {
        if(i<chars.length-1 && map.get(chars[i]+"") < map.get(chars[i+1]+"")){
            sum -= map.get(chars[i]+"");
        }else {
            sum += map.get(chars[i]+"");
        }
    }

    return sum;
}
```

孔乙己说“回”字有四种写法，编程语言中常见的命名风格有如下四种：

- 全部首字母大写
- 第一个单词首字母小写，其余单词首字母大写
- 单词全部小写，由下划线连接
- 单词全部小写，由减号连接

请设计并实现一个 `caseTransform` 函数，使得一个字符串 `str` 可以被方便地转成四种形式，并且将四种形式通过空格拼接成一个字符串返回

为方便起见，这里假设输入字符串全部符合以上四种形式的英文字母组合

```
public class stringTransfer {

    public static StringBuffer strTransfer(String Str) {

        // TODO Auto-generated constructor stub
        if (Str.length() == 0 || Str.equals(null)) {
            return null;
        }
        StringBuffer result = new StringBuffer();
        String aString = Str;// 全部首字母大写
        String bString = "";// 第一个单词首字母小写，其余单词首字母大写
        String cString = "";// 单词全部小写，由下划线连接
        String dString = "";// 单词全部小写，由减号连接
        int count = 0;
        for (int i = 0; i < Str.length(); i++) {
            char charTemp = Str.charAt(i);
            String strTemp = Character.toString(charTemp);
            int intTemp = charTemp;
            if (intTemp >= 65 && intTemp <= 90) {
                if (count == 0) {
                    strTemp = strTemp.toLowerCase();
                    bString = bString + strTemp;
                    cString = cString + strTemp;
                    dString = dString + strTemp;
                } else {
                    bString = bString + strTemp;
                    strTemp = strTemp.toLowerCase();
                    cString = cString + "_" + strTemp;
                    dString = dString + "-" + strTemp;
                }
            }
            count++;
        } else {
            bString = bString + strTemp;
            cString = cString + strTemp;
        }
    }
}
```

```

        dString = dString + strTemp;
    }

}

result.append(aString);
result.append(" ");
result.append(bString);
result.append(" ");
result.append(cString);
result.append(" ");
result.append(dString);
return result;
}

public static void main(String[] args) {
    // TODO Auto-generated method stub
    // String aString="azAZ";
    // for (int i = 0; i < aString.length(); i++) {
    // int a=aString.charAt(i);
    // System.out.println(a);
    // }
    // aString=aString+"d";
    // System.out.println(aString.toLowerCase());
    String teString = "PascalCaseTest";
    System.out.println(strTransfer(teString));
}

}

```