

解压报文

题目描述：

为了提升数据传输的效率，会对传输的报文进行压缩处理。输入一个压缩后的报文，请返回它解压后的原始报文。

压缩规则： $n[\text{str}]$ ，表示方括号内部的 str 正好重复 n 次。注意 n 为正整数 ($0 < n \leq 100$)， str 只包含小写英文字母，不考虑异常情况。

输入描述：

输入压缩后的报文：

- 1) 不考虑无效的输入，报文没有额外的空格，方括号总是符合格式要求的；
- 2) 原始报文不包含数字，所有的数字只表示重复的次数 n ，例如不会出现像 $5b$ 或 $3[8]$ 的输入；

输出描述：

解压后的原始报文

注：

- 1) 原始报文长度不会超过 1000，不考虑异常的情况

示例 1

输入：

3[k]2[mn]

输出：

kkkmnmn

说明：

k 重复 3 次， mn 重复 2 次，最终得到 $kkkmnmn$

示例 2

输入：

3[m2[c]]

输出：

```
mccmccmcc
```

说明:

m2[c] 解压缩后为 mcc, 重复三次为 mccmccmcc

```
import re
```

```
if __name__ == '__main__':
    in_str = input()
    num_list = re.findall(r'\d{3}|\d{2}|\d{1}', in_str)
    #print(num_list)
    #exit()
    str_list = []
    index = 0
    index_stack = []
    t_str = ""
    for s in in_str:
        if ord(s) >= ord('a') and ord(s) <= ord('z'):
            t_str = ".join([t_str, s])
        if s == '[':
            str_list.append(t_str)
            t_str = ""
            index += 1
            index_stack.append(index)
        if s == ']':
            # print(str_list, index)
            str_list[-1] += t_str * int(num_list[index_stack[-1] - 1])
            t_str = str_list[-1]
            str_list.pop()
            index_stack.pop()
    print(t_str)
```