

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[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

补充说明：

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

```
#include <iostream>
```

```
#include <stack>
```

```
using namespace std;
```

```
int main() {
```

```
    string str;
```

```
    cin >> str;
```

```
    string tmpStr = "";
```

```
    int tmpNum = 0;
```

```
    stack<int> left;
```

```
    stack<string> pre;
```

```
    stack<int> preNum;
```

```
    for (int i = 0; i < str.size(); ++i) {
```

```
        if (str[i] <= '9' && str[i] >= '0') {
```

```
            tmpNum = tmpNum * 10 + str[i] - '0';
```

```
        } else if (str[i] == '['){
```

```
            pre.push(tmpStr);
```

```
            tmpStr = "";
```

```
            left.push(i);
```

```
            preNum.push(tmpNum);
```

```
            tmpNum = 0;
```

```
        } else if (str[i] == ']') {
```

```
        string newTmpStr = pre.top();
        pre.pop();
        int num = preNum.top();
        preNum.pop();
        for (int j = 0; j < num; ++j) {
            newTmpStr.append(tmpStr);
        }
        tmpStr = newTmpStr;
    } else {
        tmpStr.push_back(str[i]);
    }
}

cout << tmpStr << endl;
return 0;

}
```