

## 敏感字段加密

### 题目描述：

【敏感字段加密】给定一个由多个命令字组成的命令字符串：

- 1、字符串长度小于等于 127 字节，只包含大小写字母，数字，下划线和偶数个双引号；
  - 2、命令字之间以一个或多个下划线\_进行分割；
  - 3、可以通过两个双引号""来标识包含下划线\_的命令字或空命令字（仅包含两个双引号的命令字），双引号不会在命令字内部出现；
- 请对指定索引的敏感字段进行加密，替换为\*\*\*\*\*（6 个\*），并删除命令字前后多余的下划线\_。如果无法找到指定索引的命令字，输出字符串 ERROR。

### 输入描述：

输入为两行，第一行为命令字索引 K（从 0 开始），第二行为命令字符串 S。

### 输出描述：

输出处理后的命令字符串，如果无法找到指定索引的命令字，输出字符串 ERROR

### 补充说明：

### 示例

#### 示例 1

##### 输入：

```
1
password__a12345678_timeout_100
```

##### 输出：

```
password_*****_timeout_100
```

##### 说明：

#### 示例 2

##### 输入：

```
2
aaa_password_"a12_45678"_timeout__100_"_"
```

##### 输出：

```
aaa_password_*****_timeout_100_"_"
```

说明:

```
#include <iostream>
#include <vector>

using namespace std;

int main()
{
    int K;
    string str;
    while(cin >> K >> str)
    {
        if(K < 0)
        {
            cout << "ERROR" << endl;
            continue;
        }
        vector<string> keys;
        int i = 0;
        for(; i < str.size();i++)//跳过连续 '_'
        {
            if(str[i] != '_')
                break;
        }
        int start = i;//记录关键字开始位置
        for(; i < str.size(); i++)
        {
            if(str[i] == '_')
            {
                keys.push_back(str.substr(start, i - start));
                for(i++; i < str.size();i++)//跳过连续 '_'
                {
                    if(str[i] != '_')
                        break;
                }
                start = i;
                i--;
            }
            else if(str[i] == '"')
            {
                //cout << i << ", ";
                for(i++; i < str.size(); i++)
                {
                    if(str[i] == '"')
```

```

        {
            //cout << i << endl;
            keys.push_back(str.substr(start, i - start + 1));
            break;
        }
    }
    for(i++; i < str.size(); i++) // 跳过连续 '_'
    {
        if(str[i] != '_')
            break;
    }
    start = i;
    i--;
}

}
if(start < str.size())
{
    keys.push_back(str.substr(start));
}
if(K >= keys.size())
{
    cout << "ERROR" << endl;
    continue;
}
keys[K] = "*****";
string res = keys[0];
for(int i = 1; i < keys.size(); i++)
{
    res += "_" + keys[i];
}
cout << res << endl;
}
}

```