

[编程|40分] 拆礼盒

时间限制：C/C++ 1秒，其他语言 2秒

空间限制：C/C++ 32768K，其他语言 65536K

64bit IO Format: %lld

语言限定：C++11(clang++ 3.9), Java(javac 1.8), C(clang 3.9),

Python(2.7.3), Python3(3.5.2), JavaScript (V8 6.0.0)

本题可使用本地IDE编码，不做跳出限制，编码后请点击“保存并调试”按钮进行代码提交。

■ 题目描述

小v所在的公司即将举行年会，年会方案设计过程中必不可少的一项就是抽奖活动。小v在本次活动中被委以重任，负责抽奖活动的策划；为了让中奖的礼物更加精美且富有神秘感，打算采用礼品盒来包装奖品，此时小v发挥了自己的创意想捉弄一下获奖的同事，便采取了多重包装来包装奖品。

现给出一个字符串，并假定用一对圆括号()表示一个**礼品盒**，0表示**奖品**，你能据此帮获奖者算出**最少**要拆多少个礼品盒才能拿到奖品吗？

输入描述：

一行字符串，仅有 '('、')'、'0' 组成，其中一对 '('、')' 表示一个礼品盒，'0' 表示奖品；输入的字符串一定是有效的，即 '('、')' 一定是成对出现的。

输出描述：

输出结果为一个数字，表示小v要拆的最少礼品盒数量

示例1 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

```
((()((() (0)))))
```

输出

复制

```
5
```

示例2 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

(((0)))

输出

复制

3

```
#include<stdio.h>
#include<algorithm>
#include<string>
#include<iostream>
#include<string.h>
using namespace std;

// ((()((()((0))))))

int main(){
    string str;
    cin >> str;

    int num = 0;

    for(int i=0; i<str.size(); i++)
    {
        if(str[i] == '(')
            num++;
        else if(str[i] == ')')
            num--;
        else if(str[i] == '0')
            break;
    }

    cout << num << endl;

    return 0;
}
```



```
        }
    }
}

for(int i=0;i<=disk;i++){
    for(int j=0;j<=mem;j++){
        if(dp[countOfApp][i][j]!=-1){
            res=max(res,dp[countOfApp][i][j]);
        }
    }
}
```



```

* Welcome to vivo !
*/

#define MAX_NUM 100

int solution(int boxs[], int N)
{
    // TODO Write your code here
    int res = 0;
    vector<int> arr(boxs, boxs + N);
    sort(arr.begin(), arr.end());
    for (int i = 0; i < N-1; ++i) {
        int tmp = 1;
        while (i+1<N&&arr[i] == arr[i + 1]) {
            tmp++;
            ++i;
        }
        res += tmp*tmp;
    }
    return res;
}

int main()
{
    string str("");
    getline(cin, str);
    int boxs[MAX_NUM];
    int i = 0;
    char *p;
    int count = 0;

    const char* strc = str.c_str();
    p = strtok((char *)strc, " ");
    while (p)
    {
        boxs[i] = atoi(p);
        count++;
        p = strtok(NULL, " ");
        i++;
        if (i >= MAX_NUM)
            break;
    }

    int num = solution(boxs, count);
    cout << num << endl;
    return 0;
}

```

