

P1125 [NOIP2008 提高组] 笨小猴

### 题目描述

笨小猴的词汇量很小，所以每次做英语选择题的时候都很头疼。但是他找到了一种方法，经试验证明，用这种方法去选择选项的时候选对的几率非常大！

这种方法的具体描述如下：假设 $\max n$ 是单词中出现次数最多的字母的出现次数， $\min n$ 是单词中出现次数最少的字母的出现次数，如果 $\max n - \min n$ 是一个质数，那么笨小猴就认为这是个Lucky Word，这样的单词很可能就是正确的答案。

### 输入格式

一个单词，其中只可能出现小写字母，并且长度小于100。

### 输出格式

共两行，第一行是一个字符串，假设输入的的单词是Lucky Word，那么输出“Lucky Word”，否则输出“No Answer”；

第二行是一个整数，如果输入单词是Lucky Word，输出 $\max n - \min n$ 的值，否则输出0。

### 输入样例

error

### 输出样例

Lucky Word  
2

### 解析

需要注意的是最少字母出现次数的求法。

### 编码

```
#include<bits/stdc++.h>

using namespace std;
//为26个字母建立桶
int m[26];

int main() {
```

```

//最多的单词数量
int maxNum = INT_MIN;
//最小的单词数量
int minNum = INT_MAX;

string a;
cin >> a;
int l = a.length();
for (int i = 0; i < l; i++) {
    int index = a[i] - 'a';
    //将小写字母转成数字
    m[index]++;
}
//选出数量最多和最少的单词
for (int i = 0; i < 26; ++i) {
    //注意指定的字母必须要有值才合理
    if (m[i] > 0) {
        maxNum = max(maxNum, m[i]);
        minNum = min(minNum, m[i]);
    }
}
maxNum -= minNum;
if (maxNum == 0 || maxNum == 1) {
    cout << "No Answer" << endl;
    cout << 0;
    return 0;
}
for (int i = 2; i <= sqrt(maxNum); i++) {
    if (maxNum % i == 0) {
        cout << "No Answer" << endl;
        cout << 0;
        return 0;
    }
}
cout << "Lucky Word" << endl;
cout << maxNum;
return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

