

7 编程题：幻方的判断。（2分）

**题目内容：**  
一个n度幻方共有n的平方个数字的排列，它们都是不同的整数，在一个方块中，n个数字在所有行、所有列和所有对角线中的和都相同。  
请编写is\_magicsquare函数判断一个填充好数字的方形是否是幻方。

```
#在此处编写is_magicsquare函数的定义代码

if __name__=='__main__':
    n = eval(input())
    ls = []
    for i in range(n):
        ls.append(list(eval(input())))
    #print(ls)
    if is_magicsquare(ls)==True:
        print('Yes')
    else:
        print('No')
```

**输入格式：**  
第一行输入一个整数n，表示该数字正方形的度数。然后下面是n行，每行包含n个用英文逗号分开的正整数。

**输出格式：**  
如果是幻方则输出Yes，否则输出No。

**输入样例：**  
2  
1,2  
3,4

**输出样例：**  
No

综合成绩：
1. 程序填空。（2: 未提交
2. 程序填空。（2: 未提交
3. 程序填空。（2: 未提交
4. 程序填空。（2: 未提交
5. 程序填空。（2: 未提交
▶ 6. 编写程序实现功能：使用选择排序数） 最好的成绩：2分数

输入样例:

2  
4,4  
4,4

输出样例:

No

输入样例:

3  
8,1,6  
3,5,7  
4,9,2

输出样例:

Yes

输入样例:

4  
16,9,6,3  
5,4,15,10  
11,14,1,8  
2,7,12,13

输出样例:

Yes

时间限制: 500ms 内存限制: 32000kb

Python3.4 ▼

def is\_magicsquare(lst):

综合成绩

1. 程序填空。(未提交)

2. 程序填空。(未提交)

3. 程序填空。(未提交)

4. 程序填空。(未提交)

5. 程序填空。(未提交)

6. 编写程序实现能: 使用选择排序)

最好的成绩: 2分

Python3.4 ▼

```
def is_magicsquare(lst):
    # 将lst转换成集合, 去掉重复元素, 如果集合长度与lst中元素数量相等, 则说明没有重复数据
    lst_set = set(lst[0])
    for i in range(1, len(lst)):
        lst_set.update(lst[i])

    if len(lst)**2 != len(lst_set):
        return False

    zero_row_sum = sum(lst[0])

    # 判断2个对角线上的数据和, 并且都与第0行的数据对比
```

用例测试结果	运行时间	占用内存	提示
用例1通过	54ms	4636kb	
用例2通过	29ms	4640kb	

提交答案

本次得分

```

def is_magicsquare(lst):
    lst_set = set(lst[0])
    for i in range(1, len(lst)):
        lst_set.update(lst[i])

    if len(lst)**2 != len(lst_set):
        return False

    zero_row_sum = sum(lst[0])

    sum1 = sum([lst[i][i] for i in range(len(lst))])
    sum2 = sum([lst[i][len(lst) - 1 - i] for i in range(len(lst))])
    if sum1 != sum2 or sum1 != zero_row_sum or sum2 != zero_row_sum:
        return False

    for i in range(len(lst)):
        if sum(lst[i]) != zero_row_sum or sum(lst[:,i]) != zero_row_sum or sum(lst[i]) !=
sum(lst[:,i]):
            return False
    return True

if __name__ == '__main__':
    n = eval(input())
    ls = []
    for i in range(n):
        ls.append(list(eval(input())))
    #print(ls)
    if is_magicsquare(ls) == True:
        print('Yes')
    else:
        print('No')

```

## 8 编程题：查字典。（2分）

### 题目内容：

假设有一种A语言，其单词也是由26个英文字母组成，但拼写与英文完全不同。请编写程序实现A语言单词翻译成英文单词的功能。

### 输入格式：

第1行输入单词的数量n。

从第2~2\*n+1行，每连续两行输入的单词，前一行单词是英文单词，后一行单词是对应的A语言单词。

第2\*n+2行输入待查单词的数量m。

第2\*n+3~2\*n+m+2行，每行输入一个A语言单词。

### 输出格式：

分m行输出翻译结果。如果能够找到输入的A语言单词，则输出对应的英文；如果找不到，则输出notfound。

### 输入样例：

```
4
cat
atcay
pig
igpay
froot
ootfray
loops
oopslay
3
atcay
ittenkay
oopslay
```

### 输出样例：

```
cat
notfound
loops
```

综合

1. 程序填

未提交

2. 程序填

未提交

3. 程序填

未提交

4. 程序填

未提交

5. 程序填

未提交

6. 编写程  
能：使用  
数)

最好的成绩

7. 编程题

输入样例：

4  
cat  
atcay  
pig  
igpay  
froot  
ootfray  
loops  
oopslay  
3  
atcay  
ittenkay  
oopslay

输出样例：

cat  
notfound  
loops

时间限制：500ms 内存限制：32000kb

Python3.4

```
words_number = int(input())
words_dict = {}
for i in range(words_number):
    Eng_word = input()
    A_word = input()
    words_dict[A_word] = Eng_word

translate_words_number = int(input())
for i in range(translate_words_number):
```

综

1. 程序

未提交

2. 程序

未提交

3. 程序

未提交

4. 程序

未提交

5. 程序

未提交

6. 编写

能：使

数)

最好的成

时间限制：500ms 内存限制：32000kb

Python3.4

```
words_number = int(input())
words_dict = {}
for i in range(words_number):
    Eng_word = input()
    A_word = input()
    words_dict[A_word] = Eng_word

translate_words_number = int(input())
for i in range(translate_words_number):
    translate_word = input()
    translate_info = words_dict.get(translate_word, None)
    if not translate_info:
```

综合成绩：6分

1. 程序填空。(2分数)

未提交

2. 程序填空。(2分数)

未提交

3. 程序填空。(2分数)

未提交

4. 程序填空。(2分数)

未提交

5. 程序填空。(2分数)

未提交

本次得分/总分：2.00/2.00分

6. 编写程序实现以下功能：使用选择排序... (2分)

提交答案

用例测试结果

运行时间

占用内存

提示

用例1通过

29ms

4612kb

```
words_number = int(input())
words_dict = {}
for i in range(words_number):
    Eng_word = input()
    A_word = input()
    words_dict[A_word] = Eng_word

translate_words_number = int(input())
for i in range(translate_words_number):
    translate_word = input()
    translate_info = words_dict.get(translate_word, None)
    if not translate_info:
        print('notfound')
    else:
        print(translate_info)
```