

# 特工年度大考





## 1. 探索新知

### 特工大考

各位优秀的特工们，这一年辛苦了，你们都是工会的骄傲！

本次年度考核，为大家准备了语、数、英三个方向的题目，请任选其一完成。最后，预祝大家顺利通过考核！

#### 语文

#### 诗仙本仙

利用Python编程，制作一款“诗仙本仙”的诗词填空游戏

要求如下：

- (1) 需要有一个诗句库，诗句库包括：“白日依山尽，黄河入海流。”等（诗句库可以自己继续增加）
- (2) 通过编程在这个诗句库中随机抽取一句诗，随机选择这句诗的前半句或后半句作为空白题目
- (3) 将生成的题目打印输出，格式为“\_\_\_\_\_，黄河入海流。”或者“白日依山尽，\_\_\_\_\_”

提示：【`random.choice(seq)` 从非空序列 `seq` 返回一个随机元素】



## 1. 探索新知

数学

### 自幂数

自幂数，它指的是一个n位数，其每位上的数字的n次幂之和等于它本身。例如，153是3位数，其每位数的3次方之和 $1^3+5^3+3^3=153$ ，所以153是自幂数。

请编写一段代码，用于检测输入的数字是否为自幂数。是的话输出“True”，否则输出“False”。

英语

### 词频统计器

统计字符串中的单词和单词出现的次数

要求如下：

- (1) 统计的语句为：to be or not to be that is a question
- (2) 要求统计该句中出现的的所有单词和其出现次数
- (3) 使用字典进行输出，格式如下（注意：排列顺序可以不一致，但是统计的次数要正确）：

```
{'to': 2, 'be': 2, 'or': 1, 'not': 1, 'is': 1, 'a': 1, 'question': 1, 'that': 1}
```



## 1. 探索新知

### 1.1

### 诗仙本仙

#### 1. 创建诗句库

【poem】：  
诗

```
poems = [  
    "白日依山尽，黄河入海流。",  
    "采菊东篱下，悠然见南山。",  
    "山重水复疑无路，柳暗花明又一村。",  
    "天生我材必有用，千金散尽还复来。"  
]
```

#### 2. 随机选择一个诗句

```
# 随机选择一个诗句  
poem = random.choice(poems)
```

别忘了导入  
random库

#### 3. 将诗句分割成前后两部分

```
# 将诗句按逗号分割成前后两部分  
part = poem.split(',')
```

【思考】：此时  
part 的值是多少？



#### 4. 随机选择前半句或后半句替换为空白题目

```
# 随机决定是替换前半句还是后半句  
if random.choice([True, False]):  
    question = "_____", " + part[1] # 替换前半句  
else:  
    question = part[0] + ", _____" # 替换后半句
```

# 1.1

## 诗仙本仙

### 5. 打印输出

```
# 生成题目并打印出来  
print(question)
```

### 完整代码

```
import random  
# 诗句列表  
poems = [  
    "白日依山尽，黄河入海流。",  
    "采菊东篱下，悠然见南山。",  
    "山重水复疑无路，柳暗花明又一村。",  
    "天生我材必有用，千金散尽还复来。"  
]  
# 随机选择一个诗句  
poem = random.choice(poems)  
# 将诗句按逗号分割成前后两部分  
part = poem.split(',')  
# 随机决定是替换前半句还是后半句  
if random.choice([True, False]):  
    question = "_____", part[1] # 替换前半句  
else:  
    question = part[0] + ", _____" # 替换后半句  
# 生成题目并打印出来  
print(question)
```

## 1.2

## 自幂数

### 1. 输入数字

```
n = int(input("请输入要确认的数字: "))
```

### 2. 定义check函数，判断是否为自幂数

#### ① 定义check函数

```
def check(num):
```

#### ② 计算各位数字的n次幂之和

```
# 将数字转换为字符串以方便处理每一位数字
```

```
num_str = str(num)
```

```
# 获取数字的位数
```

```
n = len(num_str)
```

```
# 计算每位数字的n次幂之和
```

```
num_sum = 0
```

```
for i in num_str:
```

```
    num_sum += int(i) ** n
```

#### ③ 判断是否为自幂数，并返回结果

```
# 判断是否等于原数字
```

```
return num_sum == num
```

### 3. 输出结果

```
print(check(n))
```

## 完整代码

```
# 判断数字是否是自幂数
def check(num):
    # 将数字转换为字符串以方便处理每一位数字
    num_str = str(num)
    # 获取数字的位数
    n = len(num_str)
    # 计算每位数字的n次幂之和
    num_sum = 0
    for i in num_str:
        num_sum += int(i) ** n
    # 判断是否等于原数字
    return num_sum == num

# 测试函数
n = int(input("请输入要确认的数字: "))
print(check(n))
```

### 1. 设置变量string存储字符串

```
# 定义变量存储字符串  
string = "to be or not to be that is a question"
```

### 2. 将字符串分割为列表

```
lis = string.split(" ") # 使用分割函数将字符串转为列表
```

### 3. 定义空字典，用来存储单词和单词出现的次数

```
count = {} # 定义一个空字典
```

### 4. 挨个判断lis中的元素（单词）是否在字典中

```
# 遍历列表  
for x in lis:  
    # 判断列表元素是否在字典中  
    if x not in count: # 如果元素还不在字典中，那么就添加键值对  
        count[x] = 1  
    else: # 如果元素已经在字典中，那么就将键对应的值加1  
        count[x] += 1
```

### 5. 打印字典，词频统计完毕

```
print(count)
```

## 完整代码

```
# 定义变量存储字符串
string = "to be or not to be that is a question"
# 使用分割函数将字符串转为列表
lis = string.split(" ")
count = {} # 定义一个空字典
# 遍历列表
for x in lis:
    # 判断列表元素是否在字典中
    if x not in count: # 如果元素还不在于字典中，就添加键值对
        count[x] = 1
    else: # 如果元素已经在字典中，那么就将键对应的值加1
        count[x] += 1
print(count) # 打印字典
```



## 2. 强化练习

1. 运行以下代码后，程序打印的最大结果可能是？（ ）

- A. 12
- B. 13
- C. 16
- D. 17

```
import random

nums = [5, 6, 12, 1, 3, 7, 2, 8]
num1 = random.choice(nums)
num2 = random.randint(2, 5)

print(num1 + num2)
```

2. 下面Python代码执行后的输出是（ ）

- A. 100
- B. 95
- C. 55
- D. 9

```
N = 100
while N > 0:
    if N % 2:
        break
    elif N % 3 == 0:
        N -= 5
    else:
        N -= 20
print(N)
```

3. 下面可以正确输出 They're planning a party for their friend's birthday. 的Python语句是？（ ）

- A. print('They\'re planning a party for their friend\'s birthday.')
- B. print("They\'re planning a party for their friend\'s birthday.')
- C. print('They're planning a party for their friend's birthday.')
- D. print('They\'re planning a party for their friend\'s birthday.')



## 2. 强化练习

4. 执行下面python代码后，输出的结果是？（ ）

- A. [10, 20, 30] [10, 20, 30]
- B. [10, 20, 30, 40] [10, 20, 30, 40]
- C. [10, 20, 30] [10, 20, 30, 40]
- D. [10, 20, 30, 40] None

```
def add(c):  
    d = c.append(40)  
    return d  
  
a = [10, 20, 30]  
b = add(a)  
print(a, b)
```

5. 执行下列Python程序，最后的输出是（ ）？

- A. olleh dlrow
- B. hello world
- C. world hello
- D. dlrow olleh

```
s = 'hello world'  
words = s.split()  
words = [x[::-1] for x in words]  
print(" ".join(words))
```

**提示：**

[x[::-1] for x in words]意思是 对words中的每个元素进行倒序处理

### 3. 术语箱

poem 诗  
part 部分  
count 计数, 总数

### 4. 课后挑战

有一个“/data/书籍存单.csv”文件，内容如下图所示。小明编写代码，想查询小说《封神榜》的价格，请补充代码中缺失的部分。

- ① \_\_\_\_\_  
② \_\_\_\_\_

*书籍存单.csv		
文件(E)	编辑(E)	搜索(S)
1	书名,价格,库存量	
2	西游记,40,253	
3	红楼梦,80,452	
4	三国演义,60,164	
5	聊斋志异,50,172	
6	水浒传,70,561	
7	老残游记,55,492	
8	封神榜,74,497	
9	镜花缘,64,573	
10	儒林外史,43,319	

```
with open('/data/书籍存单.csv', 'r', encoding='utf-8') as f:  
    for line in f.readlines():  
        a = _____①_____.split(",")  
        if a[0] == "封神榜":  
            print( _____②_____ )
```