In [4]:

```
comments = ['这是一本非常好的书，作者用心了',
'作者大大辛苦了',
'好书，感谢作者提供了这么多的好案例',
'书在运输的路上破损了，我好悲伤。。。',
'为啥我买的书上有菜汤。。。。',
'啊啊啊啊啊啊，我怎么才发现这么好的书啊，相见恨晚',
'书的质量有问题啊，怎么会开胶呢？？？？？？',
'好好好好好好好好好好好',
'好难啊看不懂好难啊看不懂好难啊看不懂',
'书的内容很充实',
'你的书上好多代码啊，不过想想也是，编程的书嘛，肯定代码多一些',
'书很不错!!一级棒!!买书就上当当，正版，价格又实惠，让人放心!!!',
'无意中来到你小铺就淘到心意的宝贝，心情不错!',
'送给朋友的、很不错',
'这是一本好书，讲解内容深入浅出又清晰明了，推荐给所有喜欢阅读的朋友同好们。',
'好好好，下一个看看',
'渣渣渣渣渣渣渣渣',
'没用没用没用没用',
'很好的书，五颗星']
def comm(s):
    if(len(s)-len(set(s))>=3):
        return False
    else:
        return True
newcomments=list(filter(comm,comments))
print(newcomments)
```

['这是一本非常好的书，作者用心了', '作者大大辛苦了', '好书，感谢作者提供了这么多的好案例', '书在运输的路上破损了，我好悲伤。。。', '书的内容很充实', '送给朋友的、很不错', '这是一本好书，讲解内容深入浅出又清晰明了，推荐给所有喜欢阅读的朋友同好们。', '很好的书，五颗星']

In [18]:

```
text="""Python Success Stories

Background

Industrial Light & Magic (ILM) was started in 1975 by filmmaker George Lucas, in order to create the

While much of ILM's early work was done with miniature models and motion controlled cameras, ILM has

In the early days, ILM was involved with the creation of custom computer graphics hardware and softw

Naturally, as time went by many of the early innovations at ILM made it into the commercial realm, b

Today, ILM runs a batch processing environment capable of modeling, rendering and compositing tens o

Making all this work, and keeping it working, requires a certain degree of technical wizardry, as we

Enter Python
Back in 1996, in the 101 Dalmation days, ILM was exclusively an SGI IRIX shop, and the production pi

Looking ahead towards more CG-intensive films, ILM staff began to search for ways to control an incr

It was around this time that Python version 1.4 came out, and Python was coming into its own as a po

At ILM, speed of development is key, and Python was a faster way to code (and re-code) the programs

Python Streamlines Production
But Python was not designed just as a replacement for shell scripting and, as it turns out, Python e

Unlike Unix shell scripting, Python can be embedded whole as a scripting language within a larger so

Using this capability, ILM integrated Python into custom applications written in C or C++, such as I

At the same time, more and more components, such as those responsible for ILM's many custom file for

As Python was used more widely, extending and customizing in-house software became a lot easier. By

As it turned out, even some of ILM's non-technical users were able to learn enough Python to develop

Python Unifies the Toolset
Encouraged by its successes in batch process control and in scripting applications and software comp

Python is now used for tracking and auditing functionality within the production pipeline, where an

Python is also used to develop the CG artist's interface to ILM's asset management system. Designed

As Python was applied in more ways, it slowly crowded out a plethora of competing technologies for s

Hardware Costs Reduced
Although chosen initially for its ease of use and integration capabilities, Python's portability to

Once Python was in use, it made the production control system portable. This gave ILM additional fre

Source Code Access Important
After having used Python intensively for six years, ILM has yet to run into significant bugs or port

However, availability of source code for the language acts as an important insurance policy should p

One case where access to source has already been beneficial was in ILM's continued use of Python 1.4
```

```
Instead, ILM installs new systems with newer versions of Python but maintains older systems only so

Python Tested by Time
The visual effects industry is intensely competitive. To stay on top of the heap, ILM continuously r

Since its adoption in 1996, the use of Python has also been reviewed numerous times. Each time, ILM

About the Author
 Tim Fortenberry joined Industrial Light & Magic in 1999 as an intern. Later that same year he began

As an engineer, Fortenberry is responsible for developing and maintaining the myriad of applications

Originally from Southern California, Fortenberry received his Bachelor of Arts degree from the Unive
list_text = list(text.split(" "))
#print(list_text)
count_list=[]
for i in list_text:
    #print(list_text.count(i))
    count_list.append(list_text.count(i))
#print(count_list)
dictionary = dict(zip(count_list,list_text))
#print(dictionary)
max_count=max(count_list)
print("出现次数最多的单词是：")
print(dictionary.get(max_count))
print("出现次数为：")
print(max_count)
```

```
出现次数最多的单词是：
and
出现次数为：
59
```

In [ ]: