测试题（含答案）

测试题 1

问题：文本的数字化表示，每一个词对应唯一的ID

输入：一篇文章中存在大量的单词，词与词之间空格分隔

输出：取出高频词，且每个词对应其数字ID

初级版本(参考)：

```
# 代码：
words=["我","北京","天安门"]
word2id={word:index for index,word in enumerate(words)}
id2word={index:word for index,word in enumerate(words)}
print(word2id)
print(id2word)

# 结果
{'我': 0, '北京': 1, '天安门': 2}
{0: '我', 1: '北京', 2: '天安门'}
```

高级版本：

```
# 数据：一段文本
# 格式：word1 word2 word3 ... wordn
institutions anarchists advocate social relations ...
```

```
# 代码
from collections import Counter

MAX_VOCAB_SIZE = 30000  # the vocabulary size

# tokenize函数，把一篇文本转化成一个个单词
def word_tokenize(text):
    return text.split()

with open("./data/text8", "r") as fin:
    text = fin.read()

text = [w for w in word_tokenize(text.lower())]
vocab = dict(Counter(text).most_common(MAX_VOCAB_SIZE - 1))


idx_to_word = [word for word in vocab.keys()]
word_to_idx = {word: i for i, word in enumerate(idx_to_word)}

print(word_to_idx["good"])
print(idx_to_word[384])
print(len(word_to_idx))

# 结果
384
good
29999
```

测试题 2

问题：根据用户的ip地址，判断该用户所处地域

输入：由两部分组成：ip地址库 + 用户ip日志

输出：用户和其所处的地域

```
# 数据1：用户地域日志
# 格式：cookie, info, ip
56DB40DC2A  直播吧 49.80.167.142
6AB72748BC  直播吧 117.90.226.247
C4C3CFF809  直播吧 114.106.206.65
8D490EFFA8  直播吧 111.19.78.131
B5C7A787B6  直播吧 117.173.171.211
A913F82ADF  直播吧 123.130.93.249
A054229993  直播吧 171.119.171.152
5F158DD09E  直播吧 60.9.44.189
09F24D041D  直播吧 36.149.19.53
6E7320F4D8  直播吧 120.135.24.218

# 数据2：ip库：
# 格式：start ip, end ip, area, country, province
202.98.29.0 202.98.29.255 亚洲 中国 吉林
202.98.30.0 202.98.30.255 亚洲 中国 吉林
202.98.31.0 202.98.31.255 亚洲 中国 吉林
202.98.32.0 202.98.63.255 亚洲 中国 重庆
202.98.64.0 202.98.79.255 亚洲 中国 云南
202.98.80.0 202.98.87.255 亚洲 中国 云南
```

```
# 代码：
import sys

#ch2 = lambda x: '.'.join([str(x/(256**i)%256) for i in range(3,-1,-1)])
ip_convert = lambda x:sum([256**j*int(i) for j,i in enumerate(x.split('.')[::-1])]
)

def load_ip_lib_func(ip_lib_fd):
    ip_lib_list = []
    file_in = open(ip_lib_fd, 'r')
    for line in file_in:
        ss = line.strip().split(' ')
        if len(ss) != 5:
            continue
        start_ip = ss[0].strip()
        end_ip = ss[1].strip()
        area = ss[2].strip()
        country = ss[3].strip()
        province = ss[4].strip()

        ip_lib_list.append((ip_convert(start_ip), ip_convert(end_ip), area, countr
y, province))

    return ip_lib_list
```

```
def get_addr(ip_lib_list, ip_str):
    ip_num = ip_convert(ip_str)

    low_index = 0
    mid_index = 0
    high_index = len(ip_lib_list) - 1

    while (low_index < high_index):
        mid_index = (low_index + high_index) / 2
        sss = ip_lib_list[mid_index]
        start_ip = sss[0]
        end_ip = sss[1]
        provice = sss[4].strip()

        if ip_num < start_ip:
            high_index = mid_index - 1
        elif ip_num > start_ip:
            low_index = mid_index + 1

    if ip_num < start_ip:
        provice = ip_lib_list[mid_index-1][4]
    else:
        provice = ip_lib_list[mid_index][4]
    return provice


ip_lib_list = load_ip_lib_func(ip_lib_fd)

for line in sys.stdin:
    ss = line.strip().split('\t')
    if len(ss) != 3:
        continue

    cookie = ss[0].strip()
    query = ss[1].strip()
    ip_str = ss[2].strip()

    user_addr = get_addr(ip_lib_list, ip_str)
    print '\t'.join([cookie, query, ip_str, user_addr])

# 结果:
B766525A02  碰碰车 125.95.123.186  北京
4920BB0B0F  碳化木 59.173.176.173  湖北
10E22007B7  磁力泵 60.179.255.254  浙江
1CCDC85A91  磁悬浮 117.135.239.18  贵州
5C7EE0E62C  磨刀机 121.238.214.57  江苏
D9DFA0B3A9  磨刀机 220.191.231.222 浙江
B4B0FAC142  磨刀机 218.62.250.19   云南
```

注：

#1: 如果你对Python不是很熟悉也没有关系，我们为你挑选了免费的Python预修资料。 你可以联系周帆拿到课程资料，并且可以根据现在的情况制定一个学习建议。

#2：测试题是用于同学自己检查一下基础情况，你可以先根据问题通过自己编程实现获得结果，并与老师提供的答案进行比对。

所写代码不需和老师提供的答案完全一致，殊途同归，只要能够获得正确（或者相似）的结果即可。

如果能够顺利完成测试题，说明你具备学习该课程的能力。

预祝同学顺利学完课程，拿到期望的offer！