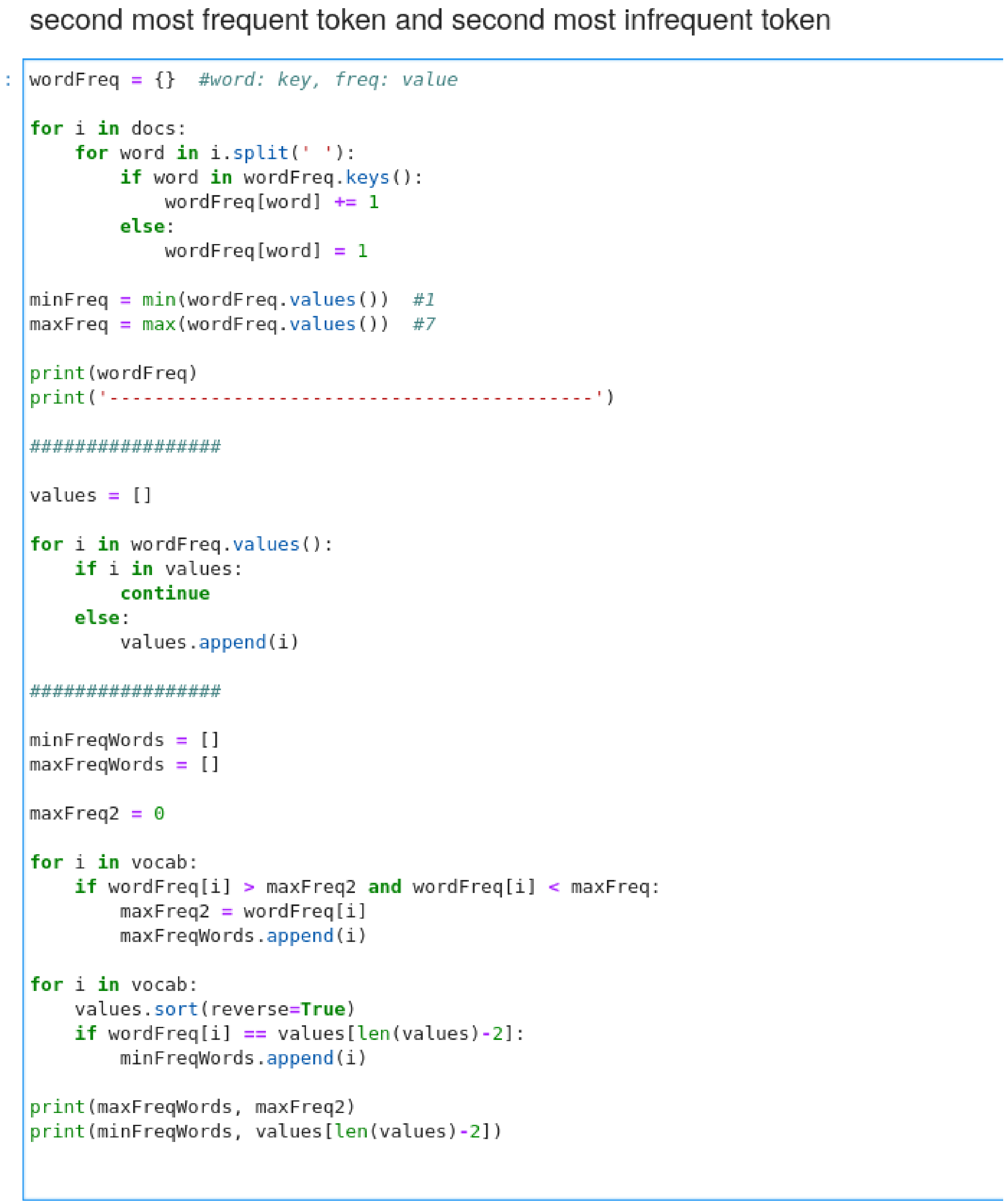
Natural Language Processing

Momina Atif Dar

P18-0030

(Tasks given during recorded lecture)

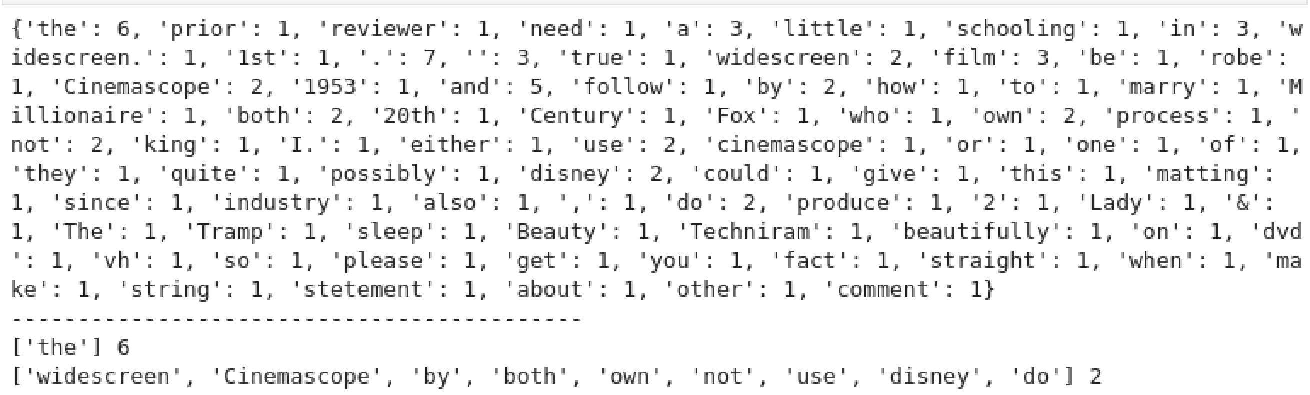
Second most frequent token and second most infrequent token:



second most frequent token and second most infrequent token

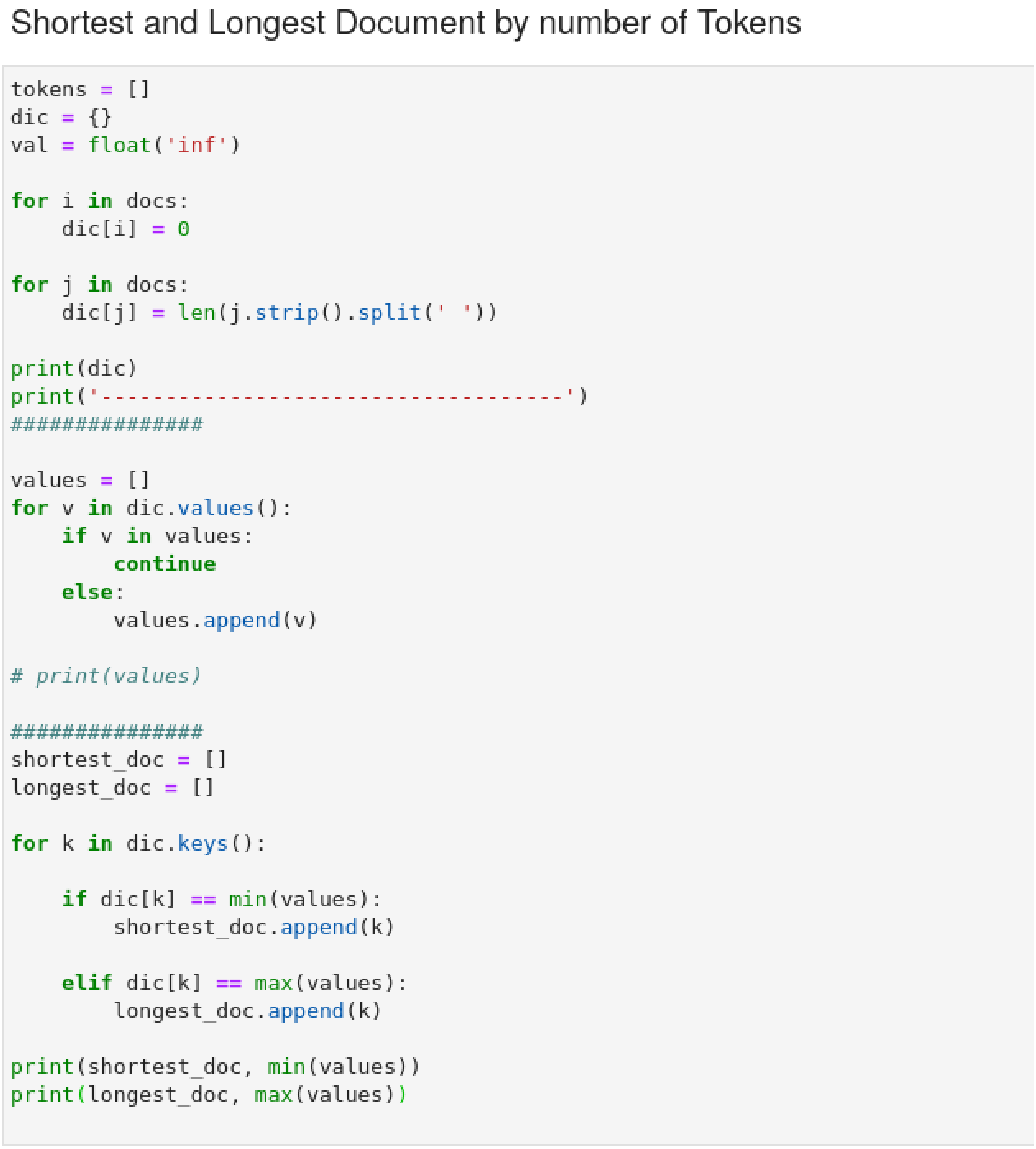
```
wordFreq = {}  #word: key, freq: value

for i in docs:
    for word in i.split(' '):
        if word in wordFreq.keys():
            wordFreq[word] += 1
        else:
            wordFreq[word] = 1

minFreq = min(wordFreq.values())  #1
maxFreq = max(wordFreq.values())  #7

print(wordFreq)
print('-------------------------------------------')

#################

values = []

for i in wordFreq.values():
    if i in values:
        continue
    else:
        values.append(i)

#################

minFreqWords = []
maxFreqWords = []

maxFreq2 = 0

for i in vocab:
    if wordFreq[i] > maxFreq2 and wordFreq[i] < maxFreq:
        maxFreq2 = wordFreq[i]
        maxFreqWords.append(i)

for i in vocab:
    values.sort(reverse=True)
    if wordFreq[i] == values[len(values)-2]:
        minFreqWords.append(i)

print(maxFreqWords, maxFreq2)
print(minFreqWords, values[len(values)-2])
```

Output:



```
{'the': 6, 'prior': 1, 'reviewer': 1, 'need': 1, 'a': 3, 'little': 1, 'schooling': 1, 'in': 3, 'w
idescreen.': 1, '1st': 1, '.': 7, '': 3, 'true': 1, 'widescreen': 2, 'film': 3, 'be': 1, 'robe':
1, 'Cinemascope': 2, '1953': 1, 'and': 5, 'follow': 1, 'by': 2, 'how': 1, 'to': 1, 'marry': 1, 'M
illionaire': 1, 'both': 2, '20th': 1, 'Century': 1, 'Fox': 1, 'who': 1, 'own': 2, 'process': 1, '
not': 2, 'king': 1, 'I.': 1, 'either': 1, 'use': 2, 'cinemascope': 1, 'or': 1, 'one': 1, 'of': 1,
'they': 1, 'quite': 1, 'possibly': 1, 'disney': 2, 'could': 1, 'give': 1, 'this': 1, 'matting':
1, 'since': 1, 'industry': 1, 'also': 1, ',': 1, 'do': 2, 'produce': 1, '2': 1, 'Lady': 1, '&':
1, 'The': 1, 'Tramp': 1, 'sleep': 1, 'Beauty': 1, 'Techniram': 1, 'beautifully': 1, 'on': 1, 'dvd
': 1, 'vh': 1, 'so': 1, 'please': 1, 'get': 1, 'you': 1, 'fact': 1, 'straight': 1, 'when': 1, 'ma
ke': 1, 'string': 1, 'stetement': 1, 'about': 1, 'other': 1, 'comment': 1}
-------------------------------------------
['the'] 6
['widescreen', 'Cinemascope', 'by', 'both', 'own', 'not', 'use', 'disney', 'do'] 2
```

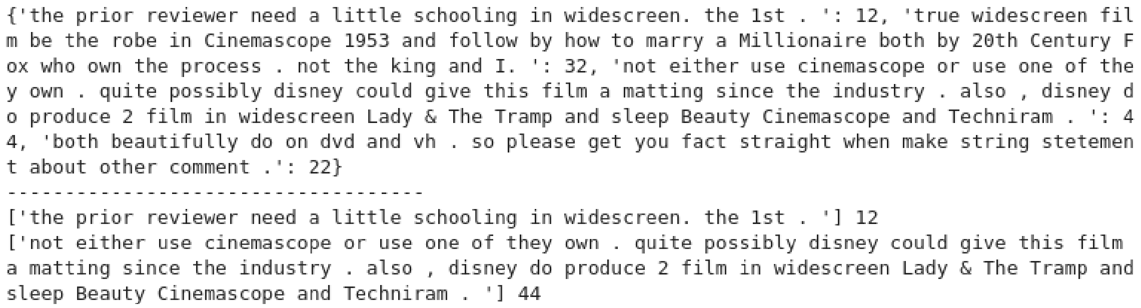
Shortest and Longest Document by number of tokens:



Shortest and Longest Document by number of Tokens

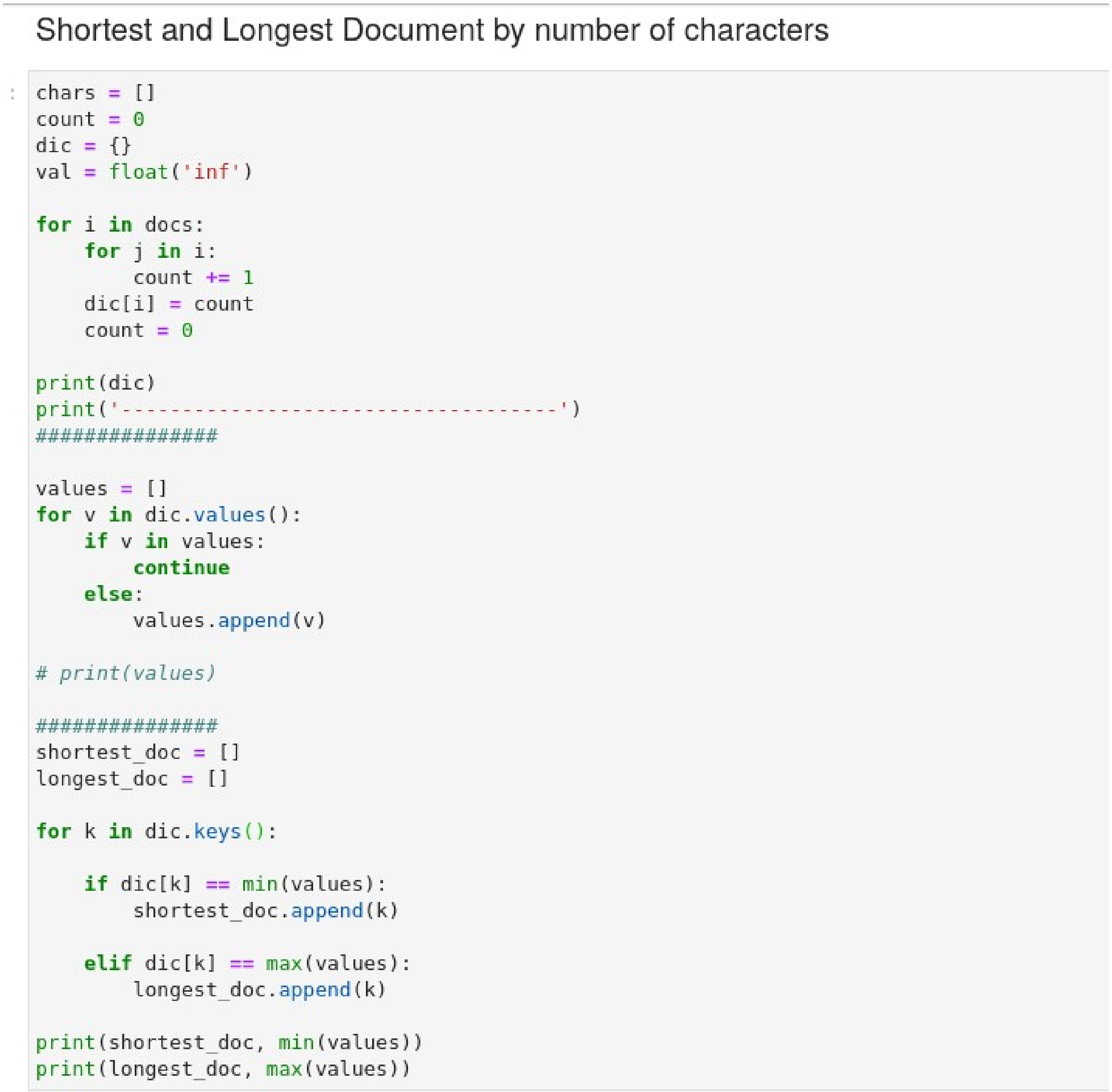
```
tokens = []
dic = {}
val = float('inf')

for i in docs:
    dic[i] = 0

for j in docs:
    dic[j] = len(j.strip().split(' '))

print(dic)
print('------------------------------------')
###############

values = []
for v in dic.values():
    if v in values:
        continue
    else:
        values.append(v)

# print(values)

###############
shortest_doc = []
longest_doc = []

for k in dic.keys():

    if dic[k] == min(values):
        shortest_doc.append(k)

    elif dic[k] == max(values):
        longest_doc.append(k)

print(shortest_doc, min(values))
print(longest_doc, max(values))
```

Output:



```
{'the prior reviewer need a little schooling in widescreen. the 1st . ': 12, 'true widescreen fil
m be the robe in Cinemascope 1953 and follow by how to marry a Millionaire both by 20th Century F
ox who own the process . not the king and I. ': 32, 'not either use cinemascope or use one of the
y own . quite possibly disney could give this film a matting since the industry . also , disney d
o produce 2 film in widescreen Lady & The Tramp and sleep Beauty Cinemascope and Techniram . ': 4
4, 'both beautifully do on dvd and vh . so please get you fact straight when make string stetemen
t about other comment .': 22}
------------------------------------
['the prior reviewer need a little schooling in widescreen. the 1st . '] 12
['not either use cinemascope or use one of they own . quite possibly disney could give this film
a matting since the industry . also , disney do produce 2 film in widescreen Lady & The Tramp and
sleep Beauty Cinemascope and Techniram . '] 44
```

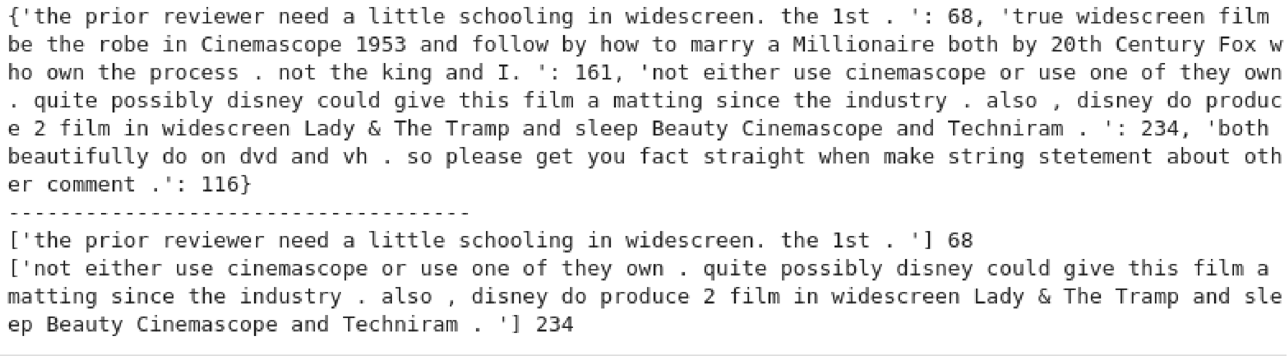
Shortest and Longest Document by number of characters:



Shortest and Longest Document by number of characters

```
chars = []
count = 0
dic = {}
val = float('inf')

for i in docs:
    for j in i:
        count += 1
    dic[i] = count
    count = 0

print(dic)
print('------------------------------------')
###############

values = []
for v in dic.values():
    if v in values:
        continue
    else:
        values.append(v)

# print(values)

###############
shortest_doc = []
longest_doc = []

for k in dic.keys():

    if dic[k] == min(values):
        shortest_doc.append(k)

    elif dic[k] == max(values):
        longest_doc.append(k)

print(shortest_doc, min(values))
print(longest_doc, max(values))
```

Output:



```
{'the prior reviewer need a little schooling in widescreen. the 1st . ': 68, 'true widescreen film
be the robe in Cinemascope 1953 and follow by how to marry a Millionaire both by 20th Century Fox w
ho own the process . not the king and I. ': 161, 'not either use cinemascope or use one of they own
. quite possibly disney could give this film a matting since the industry . also , disney do produc
e 2 film in widescreen Lady & The Tramp and sleep Beauty Cinemascope and Techniram . ': 234, 'both
beautifully do on dvd and vh . so please get you fact straight when make string stetement about oth
er comment .': 116}
------------------------------------
['the prior reviewer need a little schooling in widescreen. the 1st . '] 68
['not either use cinemascope or use one of they own . quite possibly disney could give this film a
matting since the industry . also , disney do produce 2 film in widescreen Lady & The Tramp and sle
ep Beauty Cinemascope and Techniram . '] 234
```