1. Implement a program to calculate the average occurrences of each word in a sentence in the attached article (Youvegottofindwhatyoulove.txt).

Program workflow:

讀取檔案→對文章資料轉成spark可讀取格式→計算文章句數→對資料平行化→計算字數→取前30個出現次數最高者並取平均

Execution commands:

```
hadoop fs -put /home/node/03/Downloads/Youvegottofindwhatyoulove.txt
hadoop fs -ls hdfs:///user/ubuntu
python3 /home/node03/Desktop/hw3.py
```

A. Show the top 30 most frequent occurring words and their average occurrences in a sentence.

words	counts	average counts
the	90	0.65
I	86	0.62
to	71	0.51
and	49	0.35
was	48	0.35
a	46	0.33
of	40	0.29
that	38	0.27
it	35	0.25
in	33	0.24
you	29	0.21
is	28	0.20
my	25	0.18
had	22	0.16
out	20	0.14
with	19	0.14
It	18	0.13
And	18	0.13

me	18	0.13
have	17	0.12
for	17	0.12
all	16	0.12
life	15	0.11
as	14	0.10
be	14	0.10
your	14	0.10
on	14	0.10
what	14	0.10
from	13	0.09
college	12	0.09

B. According to the result, what are the characteristics of these words?

大部分都是以介係詞為主，不太會有名詞，動詞出現的頻率也不高，有此可知，此篇文章經常使用相似的句子格式，利用不同的名詞與動詞表達不同的敘述。

在計算文字時，如果是合併後的文字，例如：can’t、don't等視為一個文字，且在計算文章句數時，首先會將雙引號內的句子連同一整個句子視為一個句子，所以在這篇文章我會計算出139句。

最後，在python中若對空字串做字串切割，則會產生一個空字串，所以在這篇文章中，段落與段落之間會皆多一行空字串，所以在做words count時，會再多算出24個空字串，我在計算時會將它們移除。

2. Implement a program to calculate the average amount in credit card trip for different number of passengers which are from one to four passengers in 2017.09 NYC Yellow Taxi trip data. In NYC Taxi data, the "Passenger_count" is a driver-entered value. Explain also how you deal with the data loss issue.

Program workflow:

讀取檔案→對文章資料轉成spark可讀取格式→只抓取部分欄位資料並對其平行化→只計算利用信用卡的消費金額→取平均

Execution commands:

```
hadoop fs -put /home/node/03/Downloads/yellow_tripdata_2017-09.csv
hadoop fs -ls hdfs:///user/ubuntu
python3 /home/node03/Desktop/hw3.py
```

numbers of passengers	average amount
1	17.9549
2	18.7733
3	18.1566
4	18.3178

data loss issue：在讀取檔案時，就先以str資料型態儲存，在python中這樣missing data會以字串”NA”表示，接著在計算數量時，直接找只用信用卡且1~4位乘客的資料做計算，會排除missing data或是error data。

3. For each of the above task 1 and 2, compare the execution time on local worker and yarn cluster. Also, give some discussions on your observation

Task1

	Execution time(seconds)
local worker	5.0560
yarn cluster	5.0970

Task2

	Execution time(seconds)
local worker	23.8610
yarn cluster	43.2634

在這次的作業中，可看出yarn cluster 的執行時間皆大於local worker的執行時間，我認為是因為這次的作業檔案皆不夠大的關係，所以yarn cluster 為了要串接其他node，而導致最後執行時間反而更大，且不同的檔案大小也會影響執行時間，即使兩個問題做的是相同的運算。