Statistical Machine Translation
Lab Exercise
4: Language Modelling

Please use Java as your programming language for this lab
Refer to the lecture slides for extra information

1- Given an input sentence, please calculate the **frequency** (*p(w)*) of each word (*w*) in the sentence according to the formula:

$$p(w) = \frac{occurrences\ of\ word}{number\ of\ tokens} \qquad (1)$$

Input: "the cat sat on the mat with a cat"
Output:

```
The word "a" frequency is: 0.111111111111
The word "on" frequency is: 0.111111111111
The word "mat" frequency is: 0.111111111111
The word "cat" frequency is: 0.222222222222
The word "the" frequency is: 0.222222222222
The word "with" frequency is: 0.111111111111
The word "sat" frequency is: 0.111111111111
```

2- Given an input sentence (*s*), please calculate the **unigram language model** of the sentence according to the formula:

$$p(s = w_1, \ldots, w_n) = p(w_1) \times \ldots \times p(w_n) \qquad (2)$$

Hint: Interpolation of the P(w) function in Question 1 could be a good idea.

Input: "the cat sat on the mat with a cat"
Output: 8.36300632515e-07

3- Following Question 1&2, please write a program to compute **bigram probability of a sentence**. The input to your program is a file containing a number of sentences and the output is the probability of one sentence. To compute **bigram relative frequency** use this formula:

$$p(w_2|w_1) = \frac{count\ (w_1,w_2)}{\sum_w count(w_1,w)} \qquad (3)$$

To compute the bigram probability of a sentence use this formula:

$$p(s) = p(w_2|w_1) \times p(w_3|w_2) \ldots \times p(w_n|w_{n-1}) \qquad (4)$$

Hint:

1, Interpolation of the function in Question 1 of Lab-3 could be a good idea.
2, Creating functions based on Question 1 and 2 could be a good idea.

Input: file_name.txt
Please calculate the probability of the sentence "**<s> a cat sat on the mat </s>**"
Output: 0.00097615576843

4- First, try another sentence using your program of Question 3:
Please calculate the probability of the sentence "**<s> a cat sat on the car </s>**". What result do you get? Think about what the reason is and why we need smoothing technique in language modeling.

Second, modify your function of **bigram relative frequency** according to add-one smoothing fomula:

$$p(w_2|w_1) = \frac{count\ (w_1,w_2)+1}{\sum_w count(w_1,w)+v} \qquad (5)$$

where v is vocabulary size (how many unique words in your file). Please use your smoothed function to calculate **bigram probability of a sentence** of the two sentences.

Input: file_name.txt
Please calculate the probability of the sentence "**<s> a cat sat on the mat </s>**"
Output: 0.000140949604457
Please calculate the probability of the sentence "**<s> a cat sat on the car </s>**"
Output: 3.00170453936e-05

Optional- In order to adapt your **bigram probability** program to **n-gram probability** program. Please add one more input to your program of Question 4**.**

Input:
1, file_name.txt
2, gram_number

Please calculate the n-gram probability of the sentence "**<s> a cat sat on the mat </s>**".

Output:
1-gram: **2.28175851587e-08**
2-gram: **0.000140949604457**
3-gram: **0.000263061746438**
4-gram: **0.000423106305459**