Name: Elgene Menon Leo Anthony

Username: leoanelge

Student Id:300492604

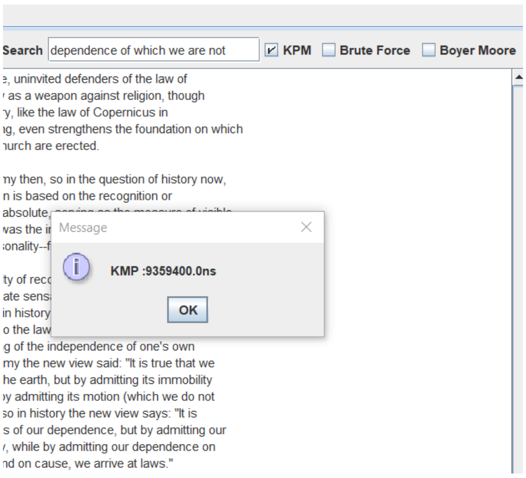
Comp261 Assignment 5

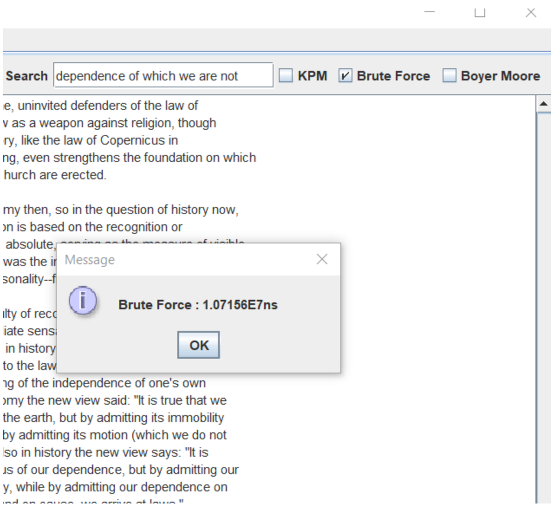
Question 1:

- ❖ Write a short summary of the performance you observed using the two search algorithms.

KMP algorithm preformed faster search than brute force algorithm. For example, below are some measures that compares both algorithms using war_and_peace.txt,

word search: dependence of which we are not





Question 2:

- ❖ Report the binary tree of codes your algorithm generates

-> 111010

 -> 111001

  -> 110

! -> 1110000111

" -> 11111010

' -> 111000010

(-> 011000111000

) -> 1111101111111

* -> 11111011010010

, -> 1111111

- -> 100101001

. -> 1110001

/ -> 0110001110010101111110

0 -> 111110110100001

1 -> 11111011010001

2 -> 111110110100000

3 -> 0110001110010111

4 -> 01100011100101010

5 -> 0110001110010100

6 -> 0110001110010110

7 -> 01100011100111110

8 -> 01100011100100

9 -> 01100011100111101

: -> 111000001001

; -> 111110110101

= -> 0110001110010101111111

? -> 1001010100

A -> 011000110

B -> 1110000001

C -> 01100010000

D -> 11111011000

E -> 01100010001

F -> 11100000101

G -> 111110111101

H -> 1110000011

I -> 100101011

J -> 11111011010011

K -> 111110111100

L -> 1111101111110

M -> 1001010101

N -> 1110000000

O -> 01100011101

P -> 011000101

Q -> 01100011100111111

R -> 11111011011

S -> 0110001111

T -> 100101000

U -> 01100011100110

V -> 111000001000

W -> 0110001001

X -> 01100011100111100

Y -> 111110111110

Z -> 011000111001110

à -> 0110001110010101110

a -> 1000

b -> 1111100

c -> 101111

d -> 10110

ä -> 01100011100101011100

e -> 000

f -> 100110

g -> 100100

h -> 0011

é -> 0110001110010101111010

i -> 0100

j -> 11111011001

ê -> 011000111001010110

k -> 0110000

l -> 01101

m -> 101110

n -> 0101

o -> 0111

p -> 1111110

q -> 11111011101

r -> 11110

s -> 0010

t -> 1010

u -> 111011

v -> 1001011

w -> 100111

x -> 1110000110

y -> 011001

z -> 11111011100

-> 0110001110010101111011

- ❖ The final size of War and Peace after Huffman coding.

input length: 3258246 bytes

output length: 1848598 bytes

Question 3:

war_and_peace.txt: Original length: 3258246 bytes
Output length: 1848598 bytes
Reduction in size: 43.26%

taisho.txt: Original length: 3649944 bytes
Output length: 1542656 bytes
Reduction in size: 57.73%

pi.txt: Original length: 1010003 bytes
Output length: 443632 bytes
Reduction in size: 56.08%

- ❖ Which of these achieves the best compression, i.e. the best reduction in size?

The best compression achieve is taisho.txt

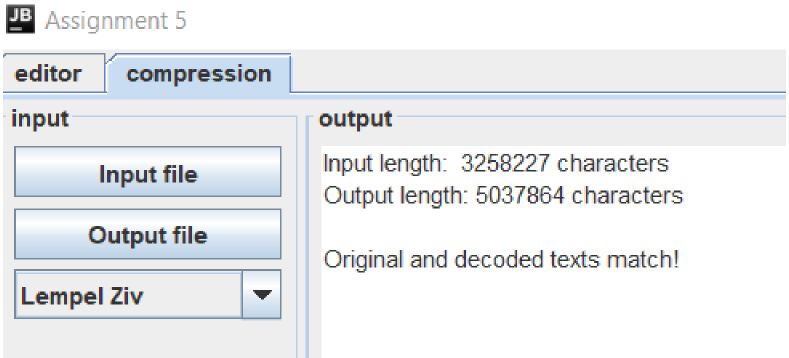
- ❖ What makes some of the encodings better than others?

This is because of the size of the character where some are small than the others such as taisho.txt has small character set.

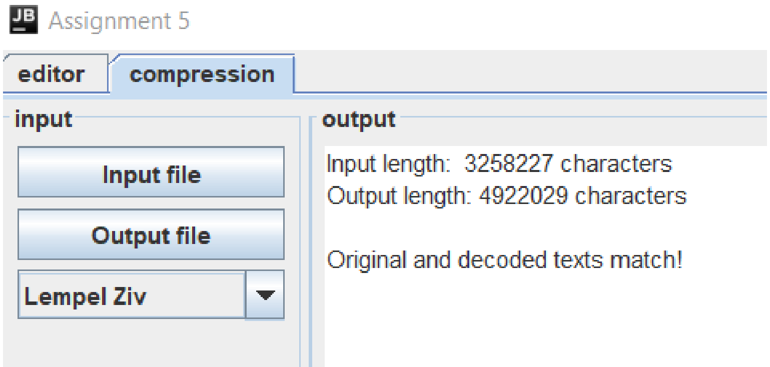
Question 4: The Lempel-Ziv algorithm has a parameter: the size of the sliding window.

- ❖ On a text of your choice
  Using war_and_peace.txt, the original size is 3258227 characters

For window size: 40000



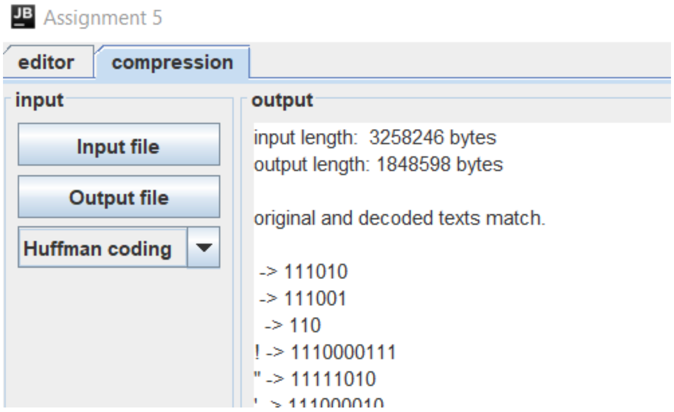
For window size: 50000

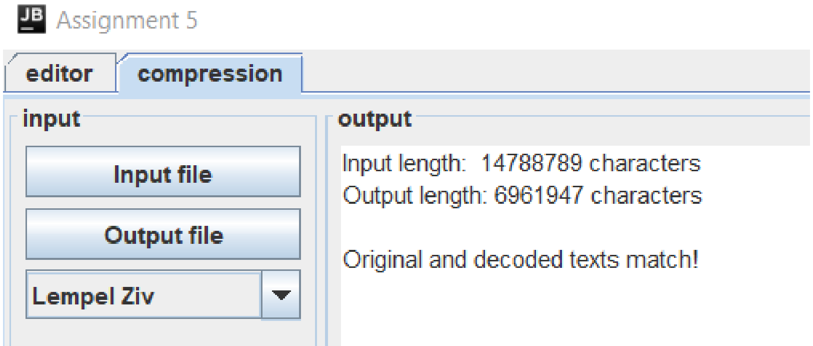


- ❖ How does changing the window size affect the quality of the compression?

Based on the output, it shows when the window size increased then the output of the compression size will be decreased in the file. This is because a larger compression file size could happen due to the way of formatting the output. The increased in size of the file because text is stored using the braces, comma, delimiters, the length and offset numbers. A better option of storing the data by using the four bytes where 2bytes for offset, the 3rd byte for length and 4th byte is for character this is for handling chucks of data. In addition, according to the data, allocate the maximum number of bits and record them by using the per block for compression.

Question 5: What happens if you Huffman encode War and Peace before applying Lempel-Ziv compression to it? Do you get a smaller file size (in characters) overall?





Yes, based on the result above, using the war_and_peace.txt with original size is 3258246 characters in length for Huffman encoding which outputted in binary which is 14788789 characters. This mean it has reduced by half in size and then used it to do Lempel-Ziv compression, the final output which is 6961947 characters showed the size of length has been increased which is due to some additional character such as the braces, comma, delimiters, the length and offset numbers.

Part4: Ngrams:

- Finds the ngrams probabilities for characters in the whakatauki

Prefix: tak|a (which mean tak followed by "a") n:3 Probability: 0.0010559662. Note that the result is obtained after applying "back-off" (n (5-2))

❖ Finds the log probability for the test that is in English

Using n-gram table the log probability of those two new string is as follows:

Turn your face to the sun and the shadows fall behind you, translation

NB: _(Underscore) *is used to represent white space and |(or sign) is used to indicate next character*

Turn|_	n:4	Prob: 0.2857143
urn_|y	n:4	Prob: 0.0056179776
rn_y|o	n:4	Prob: 0.8333333
n_yo|u	n:4	Prob: 1
_you|r	n:4	Prob: 0.18479016
your|_	n:4	Prob: 0.7655334
our_|f	n:4	Prob: 0.06315789
ur_f|a	n:4	Prob: 0.5263158
r_fa|c	n:4	Prob: 0.4293478
_fac|e	n:4	Prob: 0.8169827
face|_	n:4	Prob: 0.44354838
ace_|t	n:4	Prob: 0.099585064
ce_t|o	n:4	Prob: 0.35490605
e_to|_	n:4	Prob: 0.7316547
_to_|t	n:4	Prob: 0.19636564
to_t|h	n:4	Prob: 0.8855353
o_th|e	n:4	Prob: 0.87489265
the|	n:4	Prob: 0.08106648
the_|s	n:4	Prob: 0.10454246
he_s|u	n:4	Prob: 0.06284093
e_su|n	n:4	Prob: 0.12790698
sun|	n:4	Prob: 0.28695652
sun_|a	n:4	Prob: 0.22222222
un_a|n	n:4	Prob: 0.26086956
n_an|d	n:4	Prob: 0.7039877
and|	n:4	Prob: 0.890655
and_|s	n:4	Prob: 0.092595376
nd_s|h	n:4	Prob: 0.16666667
d_sh|a	n:4	Prob: 0.09608541
_sha|d	n:4	Prob: 0.092819616
shad|o	n:4	Prob: 0.4716981
hado|w	n:4	Prob: 1
adow|s	n:4	Prob: 0.42
dows|_	n:4	Prob: 0.51111114
ows_|f	n:4	Prob: 0.031496063
ws_f|a	n: 2	Prob: 0.29657292
s_fa|l	n:4	Prob: 0.03259259
_fal|l	n:4	Prob: 0.7477876
fall|_	n:4	Prob: 0.3068783
all_|b	n:4	Prob: 0.037077032
ll_b|e	n:4	Prob: 0.8068182
l_be|h	n:4	Prob: 0.003236246
_beh|i	n:4	Prob: 0.82894737
behi|n	n:4	Prob: 0.9152047
ehin|d	n:4	Prob: 1
hind|_	n:4	Prob: 0.72328764
ind_|y	n:4	Prob: 0.009731731
nd_y|o	n: 2	Prob: 0.7345133
d_yo|u	n:4	Prob: 1
_you|,	n:4	Prob: 0.062137723
you,|_	n:4	Prob: 0.6847826
ou,_|t	n:4	Prob: 0.014851485
u,_t|r	n: 2	Prob: 0.033236995
,_tr|a	n:4	Prob: 0.12173913
_tra|n	n:4	Prob: 0.3671875
tran|s	n:4	Prob: 0.70212764
rans|l	n:4	Prob: 0.065656565
ansl|a	n:4	Prob: 0.7692308
nsla|t	n:4	Prob: 0.8333333
slat|i	n:4	Prob: 0.3846154
lati|o	n:4	Prob: 0.7481203
atio|n	n:4	Prob: 0.99680513

$\log_{10}(0.2857143) + \log_{10}(0.0056179776) + \log_{10}(0.8333333) + \log_{10}(1) + \log_{10}(0.18479016) + \log_{10}(0.7655334)$	=	-3.72302629772 ...
$- 3.72302629772 + \log_{10}(0.06315789) + \log_{10}(0.5263158) + \log_{10}(0.4293478) + \log_{10}(0.8169827)$	=	-5.65633017533 ...
$- 5.65633017533 + \log_{10}(0.44354838) + \log_{10}(0.099585064) + \log_{10}(0.35490605) + \log_{10}(0.7316547) + \log_{10}(0.19636564)$	=	-8.30370990427 ...
$- 8.30370990427 + \log_{10}(0.8855353) + \log_{10}(0.87489265) + \log_{10}(0.08106648) + \log_{10}(0.10454246) + \log_{10}(0.06284093)$	=	-11.6881726233 ...
$- 11.6881726233 + \log_{10}(0.12790698) + \log_{10}(0.28695652) + \log_{10}(0.22222222) + \log_{10}(0.26086956) + \log_{10}(0.7039877)$	=	-14.5126863226 ...
$- 14.5126863226 + \log_{10}(0.890655) * \log_{10}(0.092595376) + \log_{10}(0.16666667) + \log_{10}(0.09608541)$	=	-16.2562093865 ...
$- 16.2562093865 + \log_{10}(0.092819616) + \log_{10}(0.4716981) + \log_{10}(1) + \log_{10}(0.42) + \log_{10}(0.51111114)$	=	-18.2831408550 ...
$- 18.2831408550 + \log_{10}(0.031496063) + \log_{10}(0.29657292) + \log_{10}(0.03259259) + \log_{10}(0.7477876) + \log_{10}(0.3068783)$	=	-22.4388897789 ...
$- 22.4388897789 + \log_{10}(0.037077032) + \log_{10}(0.8068182) + \log_{10}(0.003236246) + \log_{10}(0.82894737)$	=	-26.5344406461 ...
$- 26.5344406461 + \log_{10}(0.9152047) + \log_{10}(1) + \log_{10}(0.72328764) + \log_{10}(0.009731731)$	=	-28.7254212648 ...
$- 28.7254212648 + \log_{10}(0.7345133) + \log_{10}(1) + \log_{10}(0.062137723) + \log_{10}(0.6847826) + \log_{10}(0.014851485)$	=	-32.0587436685 ...
$- 32.0587436685 + \log_{10}(0.033236995) + \log_{10}(0.12173913) + \log_{10}(0.3671875) + \log_{10}(0.70212764)$	=	-35.0403877648 ...
$- 35.0403877648 + \log_{10}(0.065656565) + \log_{10}(0.7692308) + \log_{10}(0.8333333) + \log_{10}(0.3846154) + \log_{10}(0.7481203) + \log_{10}(0.99680513)$	=	-36.9586258323 ...

logPob (string) =
$\sum_j \boldsymbol{logProb(jth\ letter)}$
Therefore LogProb(English

The log probabilities for English version is **-36.9586258323**

❖ Finds the log probability for the test that is in Te Reo

For the Te Reo version:

Hurihia to aroaro ki te ra tukuna to atarangi kia taka ki muri i a koe

Hu|r	n:2	Prob: 0.69863015
ur|i	n:2	Prob: 0.076125
ri|h	n:0	Prob: 0.04988357
ih|i	n:2	Prob: 1
hi|a	n:2	Prob: 3.1535793E-4
ia|_	n:2	Prob: 0.047173083
a_|t	n:2	Prob: 0.059052452
_t|o	n:2	Prob: 0.23747851
to|_	n:2	Prob: 0.6661312
o_|a	n:2	Prob: 0.06780819
_a|r	n:2	Prob: 0.050021842
ar|o	n:2	Prob: 0.020478783
ro|a	n:2	Prob: 0.054253183
oa|r	n:2	Prob: 0.087936044
ar|o	n:2	Prob: 0.020478783
ro|_	n:2	Prob: 0.001786113
o_|k	n:2	Prob: 0.01033349
_k|i	n:2	Prob: 0.24792452
ki|_	n:2	Prob: 0.06934307
i_|t	n:2	Prob: 0.07883818
_t|e	n:2	Prob: 0.017018987
te|_	n:2	Prob: 0.11733703
e_|r	n:2	Prob: 0.03199047
_r|a	n:2	Prob: 0.095168374
ra|_	n:2	Prob: 0.008780488
a_|t	n:2	Prob: 0.059052452
_t|u	n:2	Prob: 0.009592043

tu|k	n:2	Prob: 8.1135903E-4
uk|u	n:2	Prob: 0.016260162
ku|n	n:2	Prob: 0.16666667
un|a	n:2	Prob: 0.024658175
na|_	n:2	Prob: 0.16777408
a_|t	n:2	Prob: 0.059052452
_t|o	n:2	Prob: 0.23747851

to|_	n:2	Prob: 0.6661312
o_|a	n:2	Prob: 0.06780819
_a|t	n:2	Prob: 0.07809927
at|a	n:2	Prob: 0.048302542
ta|r	n:2	Prob: 0.057535816
ar|a	n:2	Prob: 0.025470842
ra|n	n:2	Prob: 0.1909756
an|g	n:2	Prob: 0.029984267
ng|i	n:2	Prob: 0.011726437
gi|_	n:1	Prob: 0.00207
i_|k	n:2	Prob: 0.0062240665
_k|i	n:2	Prob: 0.24792452
ki|a	n:1	Prob: 0.01743764
ia|_	n:2	Prob: 0.047173083
a_|t	n:2	Prob: 0.059052452
_t|a	n:2	Prob: 0.022622
ta|k	n:2	Prob: 0.10853868
ak|a	n:2	Prob: 0.014849551
ka|_	n:2	Prob: 0.2808399
a_|k	n:2	Prob: 0.006006768
_k|i	n:2	Prob: 0.24792452
ki|_	n:2	Prob: 0.06934307

i_|m	n:2	Prob: 0.010373444
_m|u	n:2	Prob: 0.083333336
mu|r	n:2	Prob: 0.036121674
ur|i	n:2	Prob: 0.076125
ri|_	n:2	Prob: 4.5300112E-4
i_|a	n:2	Prob: 0.15145229
_a|_	n:2	Prob: 0.14838526
a_|k	n:2	Prob: 0.006006768
_k|o	n:2	Prob: 3.773585E-4
ko|e	n:1	Prob: 0.0026217978

logPob (string) =

$\sum_j \boldsymbol{logProb(jth\ letter)}$

Therefore, LogProb(whakatauki) =

$\log_{10}(0.69863015) + \log_{10}(0.076125) + \log_{10}(0.04988357) + \log_{10}(1) + \log_{10}(3.1535793E\text{-}4)$	= -6.07746408712 ...
$-6.07746408712 + \log_{10}(0.047173083) + \log_{10}(0.059052452) + \log_{10}(0.23747851) + \log_{10}(0.6661312)$	= -9.43334780048 ...
$-9.43334780048 + \log_{10}(0.06780819) + \log_{10}(0.050021842) + \log_{10}(0.020478783) + \log_{10}(0.054253183)$	= -14.8571766013 ...
$-14.8571766013 + \log_{10}(0.087936044) + \log_{10}(0.020478783) + \log_{10}(0.001786113) + \log_{10}(0.01033349)$	= -22.3355495782 ...
$-22.3355495782 + \log_{10}(0.24792452) + \log_{10}(0.06934307) + \log_{10}(0.07883818) + \log_{10}(0.017018987)$	= -26.9725567355 ...
$-26.9725567355 + \log_{10}(0.11733703) + \log_{10}(0.03199047) + \log_{10}(0.095168374) + \log_{10}(0.008780488)$	= -32.4760897206 ...
$-32.4760897206 + \log_{10}(0.059052452) + \log_{10}(0.009592043) + \log_{10}(8.1135903E\text{-}4) + \log_{10}(0.016260162)$	= -40.6026027262 ...
$-40.6026027262 + \log_{10}(0.16666667) + \log_{10}(0.024658175) + \log_{10}(0.16777408) + \log_{10}(0.059052452)$	= -44.9928302361 ...
$-44.9928302361 + \log_{10}(0.23747851) + \log_{10}(0.6661312) + \log_{10}(0.06780819) + \log_{10}(0.07809927) + \log_{10}(0.048302542)$	= -49.3857470320 ...
$-49.3857470320 + \log_{10}(0.057535816) + \log_{10}(0.025470842) + \log_{10}(0.1909756) + \log_{10}(0.029984267)$	= -54.4618941337 ...
$-54.4618941337 + \log_{10}(0.011726437) + \log_{10}(0.00207) + \log_{10}(0.0062240665) + \log_{10}(0.24792452) + \log_{10}(0.01743764)$	= -63.6468763009 ...
$-63.6468763009 + \log_{10}(0.047173083) + \log_{10}(0.059052452) + \log_{10}(0.022622) + \log_{10}(0.10853868)$	= -68.8118285715 ...
$-68.8118285715 + \log_{10}(0.014849551) + \log_{10}(0.2808399) + \log_{10}(0.006006768)$	= -73.4130155810 ...
$-73.4130155810 + \log_{10}(0.24792452) + \log_{10}(0.06934307) + \log_{10}(0.010373444) + \log_{10}(0.083333336) + \log_{10}(0.036121674) + \log_{10}(0.076125)$	= -80.8016561258 ...
$-80.8016561258 + \log_{10}(4.5300112E\text{-}4) + \log_{10}(0.15145229) + \log_{10}(0.14838526) + \log_{10}(0.006006768) + \log_{10}(3.773585E\text{-}4) + \log_{10}(0.0026217978)$	= -94.0198960536 ...

the log probabilities for Te Reo version is **-94.0198960536**

Question 6:

- Explain (1 paragraph) why the two log probabilities are so different.

It is because the ngrams table that was built is based on English text, the chance of finding next character after the given prefix is higher in English compared to in Te Reo text. In addition, finding the following character/next character after the given prefix in Te Reo text will result in small probability because it is less likely to be the next character in English version which means the Te Reo log probability differ quite a lot compared to the English.

Question 7:

using this whakatauki: Titiro whakamuri kia haere whakamua

(a) War_and_peace.txt

```
Ti|t     n:2   Prob: 0.052287582
it|i     n:2   Prob: 0.06509021
ti|r     n:2   Prob: 0.015421722
ir|o     n:2   Prob: 0.01270971
ro|_     n:2   Prob: 0.001786113
o_|w     n:2   Prob: 0.050514538
===
_w|h     n:2   Prob: 2.8677125E-5
wh|a     n:2   Prob: 0.17335945
ha|k     n:2   Prob: 0.0024766407
ak|a     n:2   Prob: 0.014849551
ka|m     n:2   Prob: 0.002624672
===
am|u     n:2   Prob: 0.014922098
mu|r     n:2   Prob: 0.036121674
ur|i     n:2   Prob: 0.076125
ri|_     n:2   Prob: 4.5300112E-4
i_|k     n:2   Prob: 0.0062240665
===
_k|i     n:2   Prob: 0.24792452
ki|a     n:1   Prob: 0.01743764
ia|_     n:2   Prob: 0.047173083
a_|h     n:2   Prob: 0.045262266
_h|a     n:2   Prob: 0.24563798
====
ha|e     n:2   Prob: 0.0020638674
ae|r     n:2   Prob: 0.005988024
er|e     n:2   Prob: 0.18708989
re|_     n:2   Prob: 0.2677065
e_|w     n:2   Prob: 0.08050567
=====
_w|h     n:2   Prob: 0.2607324
wh|a     n:2   Prob: 0.17335945
ha|k     n:2   Prob: 0.0024766407
ak|a     n:2   Prob: 0.014849551
====
ka|m     n:2   Prob: 0.002624672
am|u     n:2   Prob: 0.014922098
mu|a     n:1   Prob: 0.020305352
```

$$\log_{10}(0.052287582) + \log_{10}(0.06509021) + \log_{10}(0.015421722) + \log_{10}(0.01270971) + \log_{10}(0.001786113) + \log_{10}(0.050514538) = -10.2204919405 \ldots$$

$$-10.2204919405 + \log_{10}(2.8677125E-5) + \log_{10}(0.17335945) + \log_{10}(0.0024766407) + \log_{10}(0.014849551) + \log_{10}(0.002624672) = -22.5393574445 \ldots$$

$$-22.5393574445 + \log_{10}(0.014922098) + \log_{10}(0.036121674) + \log_{10}(0.076125) + \log_{10}(4.5300112E-4) + \log_{10}(0.0062240665) = -32.4760588823 \ldots$$

$$-32.4760588823 + \log_{10}(0.24792452) + \log_{10}(0.01743764) + \log_{10}(0.047173083) + \log_{10}(0.045262266) + \log_{10}(0.24563798) = -38.1205256234 \ldots$$

$$-38.1205256234 + \log_{10}(0.0020638674) + \log_{10}(0.005988024) + \log_{10}(0.18708989) + \log_{10}(0.2677065) + \log_{10}(0.08050567) = -45.4230245954 \ldots$$

$$-45.4230245954 + \log_{10}(0.2607324) + \log_{10}(0.17335945) + \log_{10}(0.0024766407) + \log_{10}(0.014849551) = -51.2023057447 \ldots$$

$$-51.2023057447 + \log_{10}(0.002624672) + \log_{10}(0.014922098) + \log_{10}(0.020305352) = -57.3017902961 \ldots$$

the log probabilities for whakatauki for war_and_peace.txt (n) is -57.3017902961 bits

and then convert to bit-string $=1/2^{n} = 1/2^{-57.3017902961}$ = **1.77646920891E+17**

(b) the text at http://www.gutenberg.org/files/44897/44897.txt?

```
Tit|i   n:3   Prob: 0.25
iti|r   n:3   Prob: 0.017605634
tir|o   n:3   Prob: 0.41935483
iro|_   n:3   Prob: 0.5135135
ro_|w   n:2   Prob: 0.039887376
o_w|h   n:3   Prob: 0.31764707
_wh|a   n:3   Prob: 0.25353926
wha|k   n:3   Prob:0.39318886
hak|a   n:3   Prob:0.88505745
aka|m   n:3   Prob:0.045016076
kam|u   n:3   Prob:0.14285715
amu|r   n:3   Prob:0.18125
mur|i   n:3   Prob:0.6551724
uri|_   n:3   Prob:0.23287672
ri_|k   n:3   Prob:0.03311258
i_k|i   n:3   Prob:0.3617021
_ki|a   n:3   Prob:0.110738255
kia|_   n:3   Prob:0.45454547
ia_|h   n:3   Prob:0.06635071
a_h|a   n:3   Prob:0.50877196
_ha|e   n:3   Prob:0.15370706
hae|r   n:3   Prob:0.90654206

=====
aer|e   n:3   Prob:0.1903501
ere|_   n:3   Prob:0.47826087
re_|w   n:3   Prob:0.03451582
e_w|h   n:3   Prob:0.26107225
_wh|a   n:3   Prob:0.25353926

====
wha|k   n:3   Prob:0.39318886
hak|a   n:3   Prob:0.88505745
aka|m   n:3   Prob:0.045016076
kam|u   n:3   Prob:0.14285715
amu|a   n:3   Prob:0.33333334
```

$\log_{10}(0.25) + \log_{10}(0.017605634) + \log_{10}(0.41935483) + \log_{10}(0.5135135) + \log_{10}(0.039887376) + \log_{10}(0.31764707)$	=	-4.92049448645 …
$- 4.92049448645 + \log_{10}(0.25353926) + \log_{10}(0.39318886) + \log_{10}(0.88505745) + \log_{10}(0.045016076) + \log_{10}(0.14285715)$	=	-8.16660698413 …
$- 8.16660698413 + \log_{10}(0.18125) + \log_{10}(0.6551724) + \log_{10}(0.23287672) + \log_{10}(0.03311258) + \log_{10}(0.3617021)$	=	-11.6465032486 …
$- 11.6465032486 + \log_{10}(0.110738255) + \log_{10}(0.45454547) + \log_{10}(0.06635071) + \log_{10}(0.50877196) + \log_{10}(0.15370706) + \log_{10}(0.90654206)$	=	-15.2721777223 …
$- 15.2721777223 + \log_{10}(0.1903501) + \log_{10}(0.47826087) + \log_{10}(0.03451582) + \log_{10}(0.26107225) + \log_{10}(0.25353926)$	=	-18.9541356375 …
$- 18.9541356375 + \log_{10}(0.39318886) + \log_{10}(0.88505745) + \log_{10}(0.045016076) + \log_{10}(0.14285715) + \log_{10}(0.33333334)$	=	-22.0814145996 …

the log probabilities for whakatauki for given text (n) is -22.0814145996 bits

and then convert to bit-string = $1/2^{\ n} = 1/2^{\ -22.0814145996}$ = **4437804.24420**

<u>Question 8:</u>

please refer to Question8.java