- Your program takes two arguments: filename and topN
- You should read the given text file and preprocess the text according to following order: Tokenize the text by whitespace(not just space character, e.g. more than one space, tab, newline etc.), remove punctuations, and apply the lowercase.
- You are asked to calculate followings:
    - **Average Term Length By Initial Character:** For example, If your tokens are [”apple”,”banana”,”avocado”,”blueberry”], then your output should be like

      a = 6 b

      = 7.5
    - **Total Minimum Distance:** For each term pair, calculate the following formula

$$\frac{f(t_1) * f(t_2)}{1+ln \sum d(t_1,t_2)}$$

where *f(t)* is the count of the term *t* in the text and $d(t_1,t_2)$ gives the minimum distance between t_1 and t_2 where t_1 is followed by t_2. For example, If the text is

”aa bb cc aa cc dd bb” and t_1 = aa and t_2 = bb, then $\sum d(t_1,t_2) = 1+3 = 4$. You

should print only topN pairs according to the score.

Sample Output

InitialCharacter AverageLength
1 3.5 2
2.0 3 5.0
5 1.0
7 4.0
a 6.285714285714286
b 7.0
d 5.333333333333333
e 7.0 f 6.0 g 7.125 h 5.375 i 6.0
k 9.266666666666667 m
5.857142857142857
o 8.0
p 8.5
r 6.0
s 7.214285714285714 t
6.363636363636363
u 7.0
v 2.4285714285714284 y 10.0 z 7.5
ç 11.666666666666666 ö
11.090909090909092 ü
12.666666666666666

Pair{t1='yerleşkesindeki', t2='ve', factor=26.0}
Pair{t1='ve', t2='sayılı', factor=15.356018837890671}
Pair{t1='tarih', t2='ve', factor=13.0}
Pair{t1='donanımlı', t2='ve', factor=13.0}
Pair{t1='öğrencileri', t2='ve', factor=13.0}
Pair{t1='söyleşilere', t2='ve', factor=13.0}
Pair{t1='yaratıcı', t2='ve', factor=13.0}
Pair{t1='eden', t2='ve', factor=13.0}
Pair{t1='ve', t2='30425', factor=13.0}
Pair{t1='kültürel', t2='ve', factor=13.0}