

UE19CS332 : Algorithms for Web and Information Retrieval
ASSIGNMENT - 3

Problem definition:

- Build a search engine for any 3 corpora of your choice,
- Your Code should be able to: Search for the terms in the query – Create Postings list
- Fill the Inverted Index
- Retrieve the data from the dictionary – Query response time.

TEAM MEMBERS :

| NAME | SRN | SECTION |
|--------------|---------------|---------|
| Priya Mohata | PES2UG19CS301 | E |
| R Sharmila | PES2UG19CS309 | E |
| Ritik | PES2UG19CS332 | E |

CORPUS – 3 : NEWS HEADLINES ANALYSIS CORPUS

DATASET LOCATION : https://drive.google.com/file/d/1dL-jz_hwksiA4uhTiBPgWj_fUQB_QdEJ/view?usp=sharing

NOTEBOOK NAME : A3_P3.ipynb

Link to the notebook :

https://colab.research.google.com/drive/1_Up7II6U7diJtqLkW7P3KYeRnIEoUsnd?usp=sharing

STEPS :

Importing all libraries

```
[1] # Assignment - 3

# PRIYA MOHATA - PES2UG19CS301
# R SHARMILA - PES2UG19CS309
# RITIK - PES2UG19CS332

# News Headline Analysis

import pandas as pd
import numpy as np
import nltk
from nltk.tokenize import word_tokenize
from nltk.tokenize import sent_tokenize
nltk.download('punkt')

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
True

[2] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

Case folding

```
# CASE FOLDING

train_data=pd.read_csv('/content/drive/MyDrive/DATASETS-AIWIR/Headlines_5000.csv')
train_data["headline"] = train_data["headline"].str.lower()
train_data.head()
```

| | date | time_12hr | time_24hr | headline | category |
|---|--------------|-----------|-----------|---|----------|
| 0 | FEB 26, 2022 | 12:13 AM | 00:13 | russia-ukraine war live updates: nato leaders ... | WORLD |
| 1 | FEB 25, 2022 | 10:25 PM | 22:25 | cnbc-tv18 classroom: what should be your optio... | MARKET |
| 2 | FEB 25, 2022 | 9:47 PM | 21:47 | ukraine-russia conflict: from sunflower oil to... | MARKET |
| 3 | FEB 25, 2022 | 9:41 PM | 21:41 | ioc to be dropped from nifty 50 from march 31;... | MARKET |
| 4 | FEB 25, 2022 | 9:23 PM | 21:23 | cbi says nse himalayan yogi none other than an... | MARKET |

```
[4] train_data.shape

(4999, 5)
```

Renaming Columns

```
[5] train_data.rename(columns = {'headline':'text'}, inplace = True)

[6] train_data.columns

Index(['date', 'time_12hr', 'time_24hr', 'text', 'category'], dtype='object')
```

Sentence Tokenization

```
[7] # SENTENCE TOKENIZATION
df=train_data['text']
l=list()
for line in df:
    token=sent_tokenize(line)
    l.append(token)

[8] df=train_data['text']
train_data['sent_token']=l

[9] train_data.head()
```

| | date | time_12hr | time_24hr | text | category | sent_token |
|---|--------------|-----------|-----------|---|----------|---|
| 0 | FEB 26, 2022 | 12:13 AM | 00:13 | russia-ukraine war live updates: nato leaders ... | WORLD | [russia-ukraine war live updates: nato leaders... |
| 1 | FEB 25, 2022 | 10:25 PM | 22:25 | cnbc-tv18 classroom: what should be your optio... | MARKET | [cnbc-tv18 classroom: what should be your opti... |
| 2 | FEB 25, 2022 | 9:47 PM | 21:47 | ukraine-russia conflict: from sunflower oil to... | MARKET | [ukraine-russia conflict: from sunflower oil t... |
| 3 | FEB 25, 2022 | 9:41 PM | 21:41 | ioc to be dropped from nifty 50 from march 31;... | MARKET | [ioc to be dropped from nifty 50 from march 31... |
| 4 | FEB 25, 2022 | 9:23 PM | 21:23 | cbi says nse himalayan yogi none other than an... | MARKET | [cbi says nse himalayan yogi none other than a... |

Word Tokenization

```
[10] # WORD TOKENIZATION
df=train_data['text']
l1=list()
for line in df:
    tokens=word_tokenize(line)
    l1.append(tokens)

[11] df=train_data['text']
train_data['word_token']=l1

[12] train_data.head()
```

| | date | time_12hr | time_24hr | text | category | sent_token | word_token |
|---|--------------|-----------|-----------|---|----------|---|---|
| 0 | FEB 26, 2022 | 12:13 AM | 00:13 | russia-ukraine war live updates: nato leaders ... | WORLD | [russia-ukraine war live updates: nato leaders... | [russia-ukraine, war, live, updates, :, nato, ... |
| 1 | FEB 25, 2022 | 10:25 PM | 22:25 | cnbc-tv18 classroom: what should be your optio... | MARKET | [cnbc-tv18 classroom: what should be your opti... | [cnbc-tv18, classroom, :, what, should, be, yo... |
| 2 | FEB 25, 2022 | 9:47 PM | 21:47 | ukraine-russia conflict: from sunflower oil to... | MARKET | [ukraine-russia conflict: from sunflower oil t... | [ukraine-russia, conflict, :, from, sunflower,... |
| 3 | FEB 25, 2022 | 9:41 PM | 21:41 | ioc to be dropped from nifty 50 from march 31;... | MARKET | [ioc to be dropped from nifty 50 from march 31... | [ioc, to, be, dropped, from, nifty, 50, from, ... |
| 4 | FEB 25, 2022 | 9:23 PM | 21:23 | cbi says nse himalayan yogi none other than an... | MARKET | [cbi says nse himalayan yogi none other than a... | [cbi, says, nse, himalayan, yogi, none, other,... |

Stop Words Removal

```
[13] # STOP WORDS REMOVAL
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stoplist= stopwords.words('english')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.

stoplist=set(stoplist)
l2=list()
for i in l1:
    output = [w for w in i if not w in stoplist]
    l2.append(output)
train_data['stop_words_removed']=l2

[15] train_data.head()
```

| | date | time_12hr | time_24hr | text | category | sent_token | word_token | stop_words_removed |
|---|--------------|-----------|-----------|---|----------|---|---|---|
| 0 | FEB 26, 2022 | 12:13 AM | 00:13 | russia-ukraine war live updates: nato leaders ... | WORLD | [russia-ukraine war live updates: nato leaders... | [russia-ukraine, war, live, updates, :, nato, ... | [russia-ukraine, war, live, updates, :, nato, ... |
| 1 | FEB 25, 2022 | 10:25 PM | 22:25 | cnbc-tv18 classroom: what should be your optio... | MARKET | [cnbc-tv18 classroom: what should be your opti... | [cnbc-tv18, classroom, :, what, should, be, yo... | [cnbc-tv18, classroom, :, options, trading, ex... |
| 2 | FEB 25, 2022 | 9:47 PM | 21:47 | ukraine-russia conflict: from sunflower oil to... | MARKET | [ukraine-russia conflict: from sunflower oil t... | [ukraine-russia, conflict, :, from, sunflower... | [ukraine-russia, conflict, :, sunflower, oil, ... |
| 3 | FEB 25, 2022 | 9:41 PM | 21:41 | ioc to be dropped from nifty 50 from march 31... | MARKET | [ioc to be dropped from nifty 50 from march 31... | [ioc, to, be, dropped, from, nifty, 50, from, ... | [ioc, dropped, nifty, 50, march, 31, :, apollo... |
| 4 | FEB 25, 2022 | 9:23 PM | 21:23 | cbi says nse himalayan yogi none other than an... | MARKET | [cbi says nse himalayan yogi none other than a... | [cbi, says, nse, himalayan, yogi, none, other... | [cbi, says, nse, himalayan, yogi, none, anand... |

3s completed at 5:11 PM

Stemming

```
# STEMMING
from nltk.stem import WordNetLemmatizer
from nltk.stem import PorterStemmer

# Stemming :
final_train_stem_list=[]
ps = PorterStemmer()
for line in train_data['stop_words_removed']:
    Stem_words=[]
    for i in line:
        rootWord = ps.stem(i)
        Stem_words.append(rootWord)
    Stem_words= [word for word in Stem_words if word.isalnum()]
    final_train_stem_list.append(Stem_words)

print(final_train_stem_list[0:5])

[['war', 'live', 'updat', 'nato', 'leader', 'meet', 'reassur', 'alli', 'near', 'russia', 'ukrain', 'air', 'india', 'oper', '3', 'flight', 'bucharest', 'evac

[17] train_data['stemmed_words']=final_train_stem_list
train_data.head()
```

| | date | time_12hr | time_24hr | text | category | sent_token | word_token | stop_words_removed | stemmed_words |
|---|--------------|-----------|-----------|---|----------|---|---|---|---|
| 0 | FEB 26, 2022 | 12:13 AM | 00:13 | russia-ukraine war live updates: nato leaders ... | WORLD | [russia-ukraine war live updates: nato leaders... | [russia-ukraine, war, live, updates, :, nato, ... | [russia-ukraine, war, live, updates, :, nato, ... | [war, live, updat, nato, leader, meet, reassur... |
| 1 | FEB 25, 2022 | 10:25 PM | 22:25 | cnbc-tv18 classroom: what should be your optio... | MARKET | [cnbc-tv18 classroom: what should be your opti... | [cnbc-tv18, classroom, :, what, should, be, yo... | [cnbc-tv18, classroom, :, options, trading, ex... | [classroom, option, trade, exit, strategi, exp... |
| 2 | FEB 25, 2022 | 9:47 PM | 21:47 | ukraine-russia conflict: from sunflower oil to... | MARKET | [ukraine-russia conflict: from sunflower oil t... | [ukraine-russia, conflict, :, from, sunflower... | [ukraine-russia, conflict, :, sunflower, oil, ... | [conflict, sunflow, oil, beer, consum, good, c... |
| 3 | FEB 25, 2022 | 9:41 PM | 21:41 | ioc to be dropped from nifty 50 from march 31... | MARKET | [ioc to be dropped from nifty 50 from march 31... | [ioc, to, be, dropped, from, nifty, 50, from, ... | [ioc, dropped, nifty, 50, march, 31, :, apollo... | [ioc, drop, nifti, 50, march, 31, apollo, hosp... |
| 4 | FEB 25, 2022 | 9:23 PM | 21:23 | cbi says nse himalayan yogi none other than an... | MARKET | [cbi says nse himalayan yogi none other than a... | [cbi, says, nse, himalayan, yogi, none, other... | [cbi, says, nse, himalayan, yogi, none, anand... | [cbi, say, nse, himalayan, yogi, none, anand, ... |

Lemmatization

```
[18] # LEMMATIZATION
import nltk
nltk.download('wordnet')
final_train_lemma_word = []
wordnet_lemmatizer = WordNetLemmatizer()
for line in train_data['stop_words_removed']:
    lemma_word=[]
    for w in line:
        word1 = wordnet_lemmatizer.lemmatize(w, pos = "n")
        word2 = wordnet_lemmatizer.lemmatize(word1, pos = "v")
        word3 = wordnet_lemmatizer.lemmatize(word2, pos = ("a"))
        lemma_word.append(word1)
    lemma_word= [word for word in lemma_word if word.isalnum()]
    final_train_lemma_word.append(lemma_word)

print(final_train_lemma_word[0:5])

[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Unzipping corpora/wordnet.zip.
[['war', 'live', 'update', 'nato', 'leader', 'meet', 'reassure', 'ally', 'near', 'russia', 'ukraine', 'air', 'india', 'operate', '3', 'flight', 'bucharest',

[19] train_data['lemmatized_words']=final_train_lemma_word
train_data.head()
```

| | date | time_12hr | time_24hr | text | category | sent_token | word_token | stop_words_removed | stemmed_words | lemmatized_words |
|---|--------------------|-----------|-----------|---|----------|---|---|---|---|---|
| 0 | FEB 26 2022 | 12:13 AM | 00:13 | russia-ukraine war live updates: nato leaders ... | WORLD | [russia-ukraine war live updates: nato leaders... | [russia-ukraine, war, live, updates, :, nato, ... | [russia-ukraine, war, live, updates, :, nato, ... | [war, live, updat, nato, leader, meet, reassur... | [war, live, update, nato, leader, meet, reassu... |
| 1 | FEB 25, 2022 | 10:25 PM | 22:25 | cnbc-tv18 classroom: what should be your optio... | MARKET | [cnbc-tv18 classroom: what should be your opti... | [cnbc-tv18, classroom, :, what, should, be, yo... | [cnbc-tv18, classroom, :, options, trading, ex... | [classroom, option, trade, exit, strategi, exp... | [classroom, option, trading, exit, strategy, e... |
| 2 | FEB 25, 2022 | 9:47 PM | 21:47 | ukraine-russia conflict: from sunflower oil to... | MARKET | [ukraine-russia conflict: from sunflower oil t... | [ukraine-russia, conflict, :, from, sunflower,... | [ukraine-russia, conflict, :, sunflower, oil, ... | [conflict, sunflow, oil, beer, consum, good, c... | [conflict, sunflower, oil, beer, consumer, goo... |
| 3 | FEB 25, 2022 | 9:41 PM | 21:41 | ioc to be dropped from nifty 50 from march 31;... | MARKET | [ioc to be dropped from nifty 50 from march 31... | [ioc, to, be, dropped, from, nifty, 50, from, ... | [ioc, dropped, nifty, 50, march, 31, :, apollo... | [ioc, drop, nifti, 50, march, 31, apollo, hosp... | [ioc, dropped, nifty, 50, march, 31, apollo, h... |
| 4 | FEB 25, 2022 | 9:23 PM | 21:23 | cbi says nse himalayan yogi none other than an... | MARKET | [cbi says nse himalayan yogi none other than a... | [cbi, says, nse, himalayan, yogi, none, other,... | [cbi, says, nse, himalayan, yogi, none, anand,... | [cbi, say, nse, himalayan, yogi, none, anand, ... | [cbi, say, nse, himalayan, yogi, none, anand, ... |

Building Inverted Index

```
# GENERATING INVERTED INDEX
def generate_inverted_index(data: list):
    inv_idx_dict = {}
    for index, doc_text in enumerate(data):
        for word in doc_text:
            if word not in inv_idx_dict.keys():
                inv_idx_dict[word] = [index]
            elif index not in inv_idx_dict[word]:
                inv_idx_dict[word].append(index)
    return inv_idx_dict

final_train=generate_inverted_index(final_train_stem_list)

j=0
for i in final_train:
    print(i,":",final_train[i])
    if j==20:
        break;
    j=j+1
```

```
war : [0, 18, 47, 54, 579, 3141]
live : [0, 2183, 3765, 4736]
updat : [0, 367, 1058, 1610, 1679, 1774, 1873, 2465, 2803, 2813, 3059, 3575, 3629, 3670, 4183, 4276]
nato : [0, 54, 322]
leader : [0, 1918, 3857, 4561, 4986]
meet : [0, 217, 221, 262, 429, 492, 699, 806, 837, 894, 960, 1057, 1112, 1118, 1670, 1762, 1941, 2332, 2410, 2500, 2576, 2735, 2828, 2958, 3009, 3302, 3438,
reassur : [0]
alli : [0, 1727, 2889]
near : [0, 39, 137, 153, 160, 294, 303, 318, 398, 568, 581, 593, 604, 637, 649, 816, 864, 914, 976, 1010, 1249, 1286, 1312, 1675, 1958, 2008, 2093, 2130, 211
russia : [0, 48, 51, 60, 62, 65, 68, 102, 183, 310, 322, 355, 369, 395, 2692]
ukrain : [0, 6, 7, 11, 48, 51, 60, 62, 63, 65, 68, 78, 86, 99, 102, 123, 148, 160, 175, 177, 183, 215, 229, 262, 303, 326, 355, 362, 369, 395, 406, 423, 461,
air : [0, 7, 988, 989, 1003, 1640, 2258, 2599, 3835, 4484, 4724, 4845]
india : [0, 7, 11, 53, 59, 73, 94, 111, 228, 242, 255, 259, 267, 273, 301, 324, 326, 328, 348, 392, 427, 429, 441, 442, 455, 498, 576, 612, 613, 657, 684, 7
oper : [0, 51, 78, 1315, 1406, 1575, 1733, 2782, 3071, 3085, 3261, 3270, 3283, 3377, 3596, 3834, 4485, 4637]
3 : [0, 26, 81, 83, 219, 254, 343, 378, 420, 463, 466, 487, 590, 592, 614, 616, 627, 665, 682, 726, 762, 771, 774, 779, 800, 801, 814, 826, 847, 867, 881, 9
flight : [0, 7, 991, 1999]
bucharest : [0]
evacu : [0, 7]
```

3s completed at 5:11 PM

```

evacu : [0, 7]
indian : [0, 7, 45, 46, 47, 127, 148, 164, 232, 251, 285, 308, 442, 548, 663, 707, 886, 934, 938, 1006, 1090, 1286, 1546, 1548, 15, 15]
classroom : [1, 469, 517]
option : [1, 469, 473, 517, 1670, 1758, 2213, 2701, 2783, 3127, 3554, 3793, 4064, 4516, 4638]

[22] # SORTING THE INDEX BASED ON TERMS
final_train1=sorted(final_train.items())
final_train1

2988,
3006,
3387,
3589,
3969,
4034,
4038,
4058,
4113]],
('circl', [2233]),
('circuit',
 [119,
 1086,
 1115,
 1192,
 1362,
 1488,
 1620,
 1711,
 1848,
 2662,
 2908,
 2956,
 3055,
 3251,
 3927,
 4260,
 4322]),
('circul', [3073, 3684]),
('circular', [3125, 3646]),
('cite', [3399, 4244]),

```

Adding the module for timing the query response

```

!pip install ipython-autotime
%load_ext autotime

Collecting ipython-autotime
  Downloading ipython-autotime-0.3.1-py2.py3-none-any.whl (6.8 kB)
Requirement already satisfied: ipython in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (5.5.0)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (4.8.0)
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (2.6.1)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (57.4.0)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (0.7.5)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (5.1.1)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (4.4.2)
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.4 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (1.0.18)
Requirement already satisfied: simplegeneric<0.8 in /usr/local/lib/python3.7/dist-packages (from ipython->ipython-autotime) (0.8.1)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from prompt-toolkit<2.0.0,>=1.0.4->ipython->ipython-autotime) (1.15.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages (from prompt-toolkit<2.0.0,>=1.0.4->ipython->ipython-autotime) (0.2.5)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.7/dist-packages (from pexpect->ipython->ipython-autotime) (0.7.0)
Installing collected packages: ipython-autotime
Successfully installed ipython-autotime-0.3.1
time: 229 µs (started: 2022-03-27 11:38:06 +00:00)

```

Building Positional Index

```

# GENERATING POSITIONAL INDEX
pos_index = {}
file_map = {}
def generate_positional_index(data:list):
    fileno=0
    lineno=1
    for line in data:
        lineno+=1;
        for pos, term in enumerate(line):
            if term in pos_index:
                pos_index[term][0] = pos_index[term][0] + 1
                if fileno in pos_index[term][1]:
                    pos_index[term][1][fileno].append(pos)
                else:
                    pos_index[term][1][fileno] = [pos]
            else:
                pos_index[term] = []
                pos_index[term].append(1)
                pos_index[term].append({})
                pos_index[term][1][fileno] = [pos]
        fileno += 1
    return pos_index

final=generate_positional_index(final_train_stem_list)
count=0
for i in final:
    count=count+1;
    if count<=20:
        print(i,final[i])
    else:
        break;

war [6, {0: [0], 18: [0], 47: [0], 54: [5], 579: [2], 3141: [3]}]
live [4, {0: [1], 2103: [4], 3765: [7], 4736: [10]}]
updat [17, {0: [2], 367: [3], 1058: [8], 1610: [10], 1679: [9], 1736: [3], 1774: [8], 1873: [3], 2465: [4], 2803: [3], 2813: [1], 3059: [3], 3575: [1], 3629:
nato [3, {0: [3], 54: [3], 322: [7]}]

```

```

[34] nato [3, {0: [3], 54: [3], 322: [1]}]
leader [5, {0: [4], 1918: [3], 3857: [7], 4561: [9], 4986: [4]}]
meet [38, {0: [5], 217: [10], 221: [4], 262: [6], 429: [7], 492: [4], 699: [3], 806: [9], 837: [7], 894: [3], 960: [10], 1057: [6], 1112: [1], 1118: [13], 1120: [1]}]
reassur [1, {0: [6]}]
alli [3, {0: [7], 1727: [1], 2889: [6]}]
near [56, {0: [8], 39: [4], 137: [2], 153: [2], 160: [2], 294: [3], 303: [2], 318: [7], 398: [6], 568: [6], 581: [9], 593: [10], 604: [2], 637: [7], 649: [3], 650: [2], 651: [2], 652: [2], 653: [2], 654: [2], 655: [2], 656: [2], 657: [2], 658: [2], 659: [2], 660: [2], 661: [2], 662: [2], 663: [2], 664: [2], 665: [2], 666: [2], 667: [2], 668: [2], 669: [2], 670: [2], 671: [2], 672: [2], 673: [2], 674: [2], 675: [2], 676: [2], 677: [2], 678: [2], 679: [2], 680: [2], 681: [2], 682: [2], 683: [2], 684: [2], 685: [2], 686: [2], 687: [2], 688: [2], 689: [2], 690: [2], 691: [2], 692: [2], 693: [2], 694: [2], 695: [2], 696: [2], 697: [2], 698: [2], 699: [2], 700: [2], 701: [2], 702: [2], 703: [2], 704: [2], 705: [2], 706: [2], 707: [2], 708: [2], 709: [2], 710: [2], 711: [2], 712: [2], 713: [2], 714: [2], 715: [2], 716: [2], 717: [2], 718: [2], 719: [2], 720: [2], 721: [2], 722: [2], 723: [2], 724: [2], 725: [2], 726: [2], 727: [2], 728: [2], 729: [2], 730: [2], 731: [2], 732: [2], 733: [2], 734: [2], 735: [2], 736: [2], 737: [2], 738: [2], 739: [2], 740: [2], 741: [2], 742: [2], 743: [2], 744: [2], 745: [2], 746: [2], 747: [2], 748: [2], 749: [2], 750: [2], 751: [2], 752: [2], 753: [2], 754: [2], 755: [2], 756: [2], 757: [2], 758: [2], 759: [2], 760: [2], 761: [2], 762: [2], 763: [2], 764: [2], 765: [2], 766: [2], 767: [2], 768: [2], 769: [2], 770: [2], 771: [2], 772: [2], 773: [2], 774: [2], 775: [2], 776: [2], 777: [2], 778: [2], 779: [2], 780: [2], 781: [2], 782: [2], 783: [2], 784: [2], 785: [2], 786: [2], 787: [2], 788: [2], 789: [2], 790: [2], 791: [2], 792: [2], 793: [2], 794: [2], 795: [2], 796: [2], 797: [2], 798: [2], 799: [2], 800: [2], 801: [2], 802: [2], 803: [2], 804: [2], 805: [2], 806: [2], 807: [2], 808: [2], 809: [2], 810: [2], 811: [2], 812: [2], 813: [2], 814: [2], 815: [2], 816: [2], 817: [2], 818: [2], 819: [2], 820: [2], 821: [2], 822: [2], 823: [2], 824: [2], 825: [2], 826: [2], 827: [2], 828: [2], 829: [2], 830: [2], 831: [2], 832: [2], 833: [2], 834: [2], 835: [2], 836: [2], 837: [2], 838: [2], 839: [2], 840: [2], 841: [2], 842: [2], 843: [2], 844: [2], 845: [2], 846: [2], 847: [2], 848: [2], 849: [2], 850: [2], 851: [2], 852: [2], 853: [2], 854: [2], 855: [2], 856: [2], 857: [2], 858: [2], 859: [2], 860: [2], 861: [2], 862: [2], 863: [2], 864: [2], 865: [2], 866: [2], 867: [2], 868: [2], 869: [2], 870: [2], 871: [2], 872: [2], 873: [2], 874: [2], 875: [2], 876: [2], 877: [2], 878: [2], 879: [2], 880: [2], 881: [2], 882: [2], 883: [2], 884: [2], 885: [2], 886: [2], 887: [2], 888: [2], 889: [2], 890: [2], 891: [2], 892: [2], 893: [2], 894: [2], 895: [2], 896: [2], 897: [2], 898: [2], 899: [2], 900: [2], 901: [2], 902: [2], 903: [2], 904: [2], 905: [2], 906: [2], 907: [2], 908: [2], 909: [2], 910: [2], 911: [2], 912: [2], 913: [2], 914: [2], 915: [2], 916: [2], 917: [2], 918: [2], 919: [2], 920: [2], 921: [2], 922: [2], 923: [2], 924: [2], 925: [2], 926: [2], 927: [2], 928: [2], 929: [2], 930: [2], 931: [2], 932: [2], 933: [2], 934: [2], 935: [2], 936: [2], 937: [2], 938: [2], 939: [2], 940: [2], 941: [2], 942: [2], 943: [2], 944: [2], 945: [2], 946: [2], 947: [2], 948: [2], 949: [2], 950: [2], 951: [2], 952: [2], 953: [2], 954: [2], 955: [2], 956: [2], 957: [2], 958: [2], 959: [2], 960: [2], 961: [2], 962: [2], 963: [2], 964: [2], 965: [2], 966: [2], 967: [2], 968: [2], 969: [2], 970: [2], 971: [2], 972: [2], 973: [2], 974: [2], 975: [2], 976: [2], 977: [2], 978: [2], 979: [2], 980: [2], 981: [2], 982: [2], 983: [2], 984: [2], 985: [2], 986: [2], 987: [2], 988: [2], 989: [2], 990: [2], 991: [2], 992: [2], 993: [2], 994: [2], 995: [2], 996: [2], 997: [2], 998: [2], 999: [2], 1000: [2], 1001: [2], 1002: [2], 1003: [2], 1004: [2], 1005: [2], 1006: [2], 1007: [2], 1008: [2], 1009: [2], 1010: [2], 1011: [2], 1012: [2], 1013: [2], 1014: [2], 1015: [2], 1016: [2], 1017: [2], 1018: [2], 1019: [2], 1020: [2], 1021: [2], 1022: [2], 1023: [2], 1024: [2], 1025: [2], 1026: [2], 1027: [2], 1028: [2], 1029: [2], 1030: [2], 1031: [2], 1032: [2], 1033: [2], 1034: [2], 1035: [2], 1036: [2], 1037: [2], 1038: [2], 1039: [2], 1040: [2], 1041: [2], 1042: [2], 1043: [2], 1044: [2], 1045: [2], 1046: [2], 1047: [2], 1048: [2], 1049: [2], 1050: [2], 1051: [2], 1052: [2], 1053: [2], 1054: [2], 1055: [2], 1056: [2], 1057: [2], 1058: [2], 1059: [2], 1060: [2], 1061: [2], 1062: [2], 1063: [2], 1064: [2], 1065: [2], 1066: [2], 1067: [2], 1068: [2], 1069: [2], 1070: [2], 1071: [2], 1072: [2], 1073: [2], 1074: [2], 1075: [2], 1076: [2], 1077: [2], 1078: [2], 1079: [2], 1080: [2], 1081: [2], 1082: [2], 1083: [2], 1084: [2], 1085: [2], 1086: [2], 1087: [2], 1088: [2], 1089: [2], 1090: [2], 1091: [2], 1092: [2], 1093: [2], 1094: [2], 1095: [2], 1096: [2], 1097: [2], 1098: [2], 1099: [2], 1100: [2], 1101:
```

Performing Boolean Queries

```
[24] # Boolean Query
# AND
def and_query(l1, l2):
    p1 = 0
    p2 = 0
    result = list()
    while p1 < len(l1) and p2 < len(l2):
        if l1[p1] == l2[p2]:
            result.append(l1[p1])
            p1 += 1
            p2 += 1
        elif l1[p1] > l2[p2]:
            p2 += 1
        else:
            p1 += 1
    return result

time: 10.8 ms (started: 2022-03-27 11:38:08 +00:00)
```

```
[25] def or_query(l1,l2):
    result=list();
    p1=0
    p2=0
    while p1 < len(l1) and p2 < len(l2):
        if l1[p1] == l2[p2]:
            result.append(l1[p1])
            p1 += 1
            p2 += 1
        elif l1[p1] > l2[p2]:
            result.append(l2[p2])
            p2 += 1
        else:
            result.append(l1[p1])
            p1 += 1
    while(p1 < len(l1)):
        result.append(l1[p1])
        p1 += 1
    while p2 < len(l2):
        result.append(l2[p2])
        p2 += 1
    return result

time: 13.9 ms (started: 2022-03-27 11:38:12 +00:00)
```

```
# PERFORMING THE BOOLEAN QUERY
print("Enter the first input word : ")
input1=input()
print("Enter the second input word : ")
input2=input()
```

```

[29] Enter the first input word :
    india
    Enter the second input word :
    russia
    time: 9.51 s (started: 2022-03-27 11:39:45 +00:00)

[30] l1=final_train[input1]
    l2=final_train[input2]
    print("posting list for",input1,l1)
    print("posting list for",input2,l2)
    print("Resultant list: ",and_query(l1,l2))

posting list for india [0, 7, 11, 53, 59, 73, 94, 111, 228, 242, 255, 259, 267, 273, 301, 324, 326, 328, 348, 392, 427, 429, 441, 442, 455, 498, 576, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000]
posting list for russia [0, 48, 51, 60, 62, 65, 68, 102, 183, 310, 322, 355, 369, 395, 2692]
Resultant list: [0, 2692]
time: 7.71 ms (started: 2022-03-27 11:39:57 +00:00)

[31] print("Resultant list: ",or_query(l1,l2))
    print("Length of posting list for",input1,len(l1))
    print("Length of posting list for",input2,len(l2))
    print("Length of and list: ",len(and_query(l1,l2)))
    print("Resultant list :",or_query(l1,l2))
    print("Length of the OR list:",len(or_query(l1,l2)))

Resultant list: [0, 7, 11, 48, 51, 53, 59, 60, 62, 65, 68, 73, 94, 102, 111, 183, 228, 242, 255, 259, 267, 273, 301, 310, 322, 324, 326, 328, 348, 355, 369, 392, 427, 429, 441, 442, 455, 498, 576, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000]
Length of posting list for india 248
Length of posting list for russia 15
Length of and list: 2
Resultant list : [0, 7, 11, 48, 51, 53, 59, 60, 62, 65, 68, 73, 94, 102, 111, 183, 228, 242, 255, 259, 267, 273, 301, 310, 322, 324, 326, 328, 348, 355, 369, 392, 427, 429, 441, 442, 455, 498, 576, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000]
Length of the OR list: 261
time: 6.7 ms (started: 2022-03-27 11:40:04 +00:00)

[32] print("Enter the third input word : ")
    input3=input()
    print("Enter the fourth input word : ")
    input4=input()
    l3=final_train[input3]
    l4=final_train[input4]
    resultant=or_query(or_query(and_query(l1,l4),l3),l2)
    print("Result:",resultant)
    print("Result length:",len(resultant))

Enter the third input word :
meet
Enter the fourth input word :
option
Result: [0, 48, 51, 60, 62, 65, 68, 102, 183, 217, 221, 262, 310, 322, 355, 369, 395, 429, 492, 699, 806, 837, 894, 960, 1057, 1112, 1118, 1670, 1762, 1941, 2692]
Result length: 52
time: 6.63 s (started: 2022-03-27 11:40:16 +00:00)

[33] resultant=and_query(or_query(l1,l3),or_query(l2,l4))
    print("Result:",resultant)
    print("Result length:",len(resultant))

Result: [0, 1670, 2692]
Result length: 3
time: 6.42 ms (started: 2022-03-27 11:40:26 +00:00)

```

Performing Phrase Query on Inverted Index

```

nato [3, {0: [3], 54: [3], 322: [7]}]
[34] leader [5, {0: [4], 1918: [3], 3857: [7], 4561: [9], 4986: [4]}]
meet [38, {0: [5], 217: [10], 221: [4], 262: [6], 429: [7], 492: [4], 699: [3], 806: [9], 837: [7], 894: [3], 960: [10], 1057: [6], 1112: [1], 1118: [13], 1670: [1], 1762: [1], 1941: [1], 2692: [8]}]
reassur [1, {0: [6]}]
alli [3, {0: [7], 1727: [1], 2889: [6]}]
near [56, {0: [8], 39: [4], 137: [2], 153: [2], 160: [2], 294: [3], 303: [2], 318: [7], 398: [6], 568: [6], 581: [9], 593: [10], 604: [2], 637: [7], 649: [3], 650: [2], 651: [2], 652: [2], 653: [2], 654: [2], 655: [2], 656: [2], 657: [2], 658: [2], 659: [2], 660: [2], 661: [2], 662: [2], 663: [2], 664: [2], 665: [2], 666: [2], 667: [2], 668: [2], 669: [2], 670: [2], 671: [2], 672: [2], 673: [2], 674: [2], 675: [2], 676: [2], 677: [2], 678: [2], 679: [2], 680: [2], 681: [2], 682: [2], 683: [2], 684: [2], 685: [2], 686: [2], 687: [2], 688: [2], 689: [2], 690: [2], 691: [2], 692: [2], 693: [2], 694: [2], 695: [2], 696: [2], 697: [2], 698: [2], 699: [2], 700: [2], 701: [2], 702: [2], 703: [2], 704: [2], 705: [2], 706: [2], 707: [2], 708: [2], 709: [2], 710: [2], 711: [2], 712: [2], 713: [2], 714: [2], 715: [2], 716: [2], 717: [2], 718: [2], 719: [2], 720: [2], 721: [2], 722: [2], 723: [2], 724: [2], 725: [2], 726: [2], 727: [2], 728: [2], 729: [2], 730: [2], 731: [2], 732: [2], 733: [2], 734: [2], 735: [2], 736: [2], 737: [2], 738: [2], 739: [2], 740: [2], 741: [2], 742: [2], 743: [2], 744: [2], 745: [2], 746: [2], 747: [2], 748: [2], 749: [2], 750: [2], 751: [2], 752: [2], 753: [2], 754: [2], 755: [2], 756: [2], 757: [2], 758: [2], 759: [2], 760: [2], 761: [2], 762: [2], 763: [2], 764: [2], 765: [2], 766: [2], 767: [2], 768: [2], 769: [2], 770: [2], 771: [2], 772: [2], 773: [2], 774: [2], 775: [2], 776: [2], 777: [2], 778: [2], 779: [2], 780: [2], 781: [2], 782: [2], 783: [2], 784: [2], 785: [2], 786: [2], 787: [2], 788: [2], 789: [2], 790: [2], 791: [2], 792: [2], 793: [2], 794: [2], 795: [2], 796: [2], 797: [2], 798: [2], 799: [2], 800: [2], 801: [2], 802: [2], 803: [2], 804: [2], 805: [2], 806: [2], 807: [2], 808: [2], 809: [2], 810: [2], 811: [2], 812: [2], 813: [2], 814: [2], 815: [2], 816: [2], 817: [2], 818: [2], 819: [2], 820: [2], 821: [2], 822: [2], 823: [2], 824: [2], 825: [2], 826: [2], 827: [2], 828: [2], 829: [2], 830: [2], 831: [2], 832: [2], 833: [2], 834: [2], 835: [2], 836: [2], 837: [2], 838: [2], 839: [2], 840: [2], 841: [2], 842: [2], 843: [2], 844: [2], 845: [2], 846: [2], 847: [2], 848: [2], 849: [2], 850: [2], 851: [2], 852: [2], 853: [2], 854: [2], 855: [2], 856: [2], 857: [2], 858: [2], 859: [2], 860: [2], 861: [2], 862: [2], 863: [2], 864: [2], 865: [2], 866: [2], 867: [2], 868: [2], 869: [2], 870: [2], 871: [2], 872: [2], 873: [2], 874: [2], 875: [2], 876: [2], 877: [2], 878: [2], 879: [2], 880: [2], 881: [2], 882: [2], 883: [2], 884: [2], 885: [2], 886: [2], 887: [2], 888: [2], 889: [2], 890: [2], 891: [2], 892: [2], 893: [2], 894: [2], 895: [2], 896: [2], 897: [2], 898: [2], 899: [2], 900: [2], 901: [2], 902: [2], 903: [2], 904: [2], 905: [2], 906: [2], 907: [2], 908: [2], 909: [2], 910: [2], 911: [2], 912: [2], 913: [2], 914: [2], 915: [2], 916: [2], 917: [2], 918: [2], 919: [2], 920: [2], 921: [2], 922: [2], 923: [2], 924: [2], 925: [2], 926: [2], 927: [2], 928: [2], 929: [2], 930: [2], 931: [2], 932: [2], 933: [2], 934: [2], 935: [2], 936: [2], 937: [2], 938: [2], 939: [2], 940: [2], 941: [2], 942: [2], 943: [2], 944: [2], 945: [2], 946: [2], 947: [2], 948: [2], 949: [2], 950: [2], 951: [2], 952: [2], 953: [2], 954: [2], 955: [2], 956: [2], 957: [2], 958: [2], 959: [2], 960: [2], 961: [2], 962: [2], 963: [2], 964: [2], 965: [2], 966: [2], 967: [2], 968: [2], 969: [2], 970: [2], 971: [2], 972: [2], 973: [2], 974: [2], 975: [2], 976: [2], 977: [2], 978: [2], 979: [2], 980: [2], 981: [2], 982: [2], 983: [2], 984: [2], 985: [2], 986: [2], 987: [2], 988: [2], 989: [2], 990: [2], 991: [2], 992: [2], 993: [2], 994: [2], 995: [2], 996: [2], 997: [2], 998: [2], 999: [2], 1000: [2]}]
ukrain [38, {0: [10], 6: [4], 7: [9], 11: [7], 48: [9], 51: [7], 60: [10], 62: [8], 63: [6], 65: [8], 68: [9], 78: [8], 86: [6], 99: [3], 102: [4], 123: [8], 124: [8], 125: [8], 126: [8], 127: [8], 128: [8], 129: [8], 130: [8], 131: [8], 132: [8], 133: [8], 134: [8], 135: [8], 136: [8], 137: [8], 138: [8], 139: [8], 140: [8], 141: [8], 142: [8], 143: [8], 144: [8], 145: [8], 146: [8], 147: [8], 148: [8], 149: [8], 150: [8], 151: [8], 152: [8], 153: [8], 154: [8], 155: [8], 156: [8], 157: [8], 158: [8], 159: [8], 160: [8], 161: [8], 162: [8], 163: [8], 164: [8], 165: [8], 166: [8], 167: [8], 168: [8], 169: [8], 170: [8], 171: [8], 172: [8], 173: [8], 174: [8], 175: [8], 176: [8], 177: [8], 178: [8], 179: [8], 180: [8], 181: [8], 182: [8], 183: [8], 184: [8], 185: [8], 186: [8], 187: [8], 188: [8], 189: [8], 190: [8], 191: [8], 192: [8], 193: [8], 194: [8], 195: [8], 196: [8], 197: [8], 198: [8], 199: [8], 200: [8], 201: [8], 202: [8], 203: [8], 204: [8], 205: [8], 206: [8], 207: [8], 208: [8], 209: [8], 210: [8], 211: [8], 212: [8], 213: [8], 214: [8], 215: [8], 216: [8], 217: [8], 218: [8], 219: [8], 220: [8], 221: [8], 222: [8], 223: [8], 224: [8], 225: [8], 226: [8], 227: [8], 228: [8], 229: [8], 230: [8], 231: [8], 232: [8], 233: [8], 234: [8], 235: [8], 236: [8], 237: [8], 238: [8], 239: [8], 240: [8], 241: [8], 242: [8], 243: [8], 244: [8], 245: [8], 246: [8], 247: [8], 248: [8], 249: [8], 250: [8], 251: [8], 252: [8], 253: [8], 254: [8], 255: [8], 256: [8], 257: [8], 258: [8], 259: [8], 260: [8], 261: [8], 262: [8], 263: [8], 264: [8], 265: [8], 266: [8], 267: [8], 268: [8], 269: [8], 270: [8], 271: [8], 272: [8], 273: [8], 274: [8], 275: [8], 276: [8], 277: [8], 278: [8], 279: [8], 280: [8], 281: [8], 282: [8], 283: [8], 284: [8], 285: [8], 286: [8], 287: [8], 288: [8], 289: [8], 290: [8], 291: [8], 292: [8], 293: [8], 294: [8], 295: [8], 296: [8], 297: [8], 298: [8], 299: [8], 300: [8], 301: [8], 302: [8], 303: [8], 304: [8], 305: [8], 306: [8], 307: [8], 308: [8], 309: [8], 310: [8], 311: [8], 312: [8], 313: [8], 314: [8], 315: [8], 316: [8], 317: [8], 318: [8], 319: [8], 320: [8], 321: [8], 322: [8], 323: [8], 324: [8], 325: [8], 326: [8], 327: [8], 328: [8], 329: [8], 330: [8], 331: [8], 332: [8], 333: [8], 334: [8], 335: [
```


Performing Phrase Query on Positional Index

```

✓ [38] # Phrase query on positional index :
0s
def fetch_list(d):
    l=list();
    d1=d[1];
    for i in d1:
        l.append(i)
    return l;
def post_phrase_query(phr):
    query=phr.split()
    for i in range(0,len(query)-1,2):
        l1=fetch_list(final[query[i]])
        l2=fetch_list(final[query[i+1]])
        result=and_query(l1,l2)
    print(result)

```

time: 8.94 ms (started: 2022-03-27 11:41:11 +00:00)

```

✓ 12s
▶ print("Enter your query")
q=input()
post_phrase_query(q)

```

```

↳ Enter your query
air india russia
[0, 7, 988, 1003, 1640, 4724, 4845]
time: 11.9 s (started: 2022-03-27 11:41:14 +00:00)

```

```

✓ [40] print("Enter your query")
4s
q=input()
post_phrase_query(q)

```

```

Enter your query
india indian
[0, 7, 442, 2251]
time: 4.42 s (started: 2022-03-27 11:41:30 +00:00)

```