# Inverted Index and Positional Index

Approach and Methodologies:-

In my approach to creating an inverted index and positional index, I have created eight files. In these eight files, I have distributed the task of preprocessing text files present in the 'text\_files' folder to the process of asking questions by the user. These are the names of the .py files I have made, along with their functionalities and the sequence in which they are used.

Python Files:-

Preprocess.py : In this file I have used NLTK, Beautiful Soup and String library the main working of this file is to perfrom the pre-processing steps of Tokenization, Lowering the letters Removing stop words , Removing the blank space tokens, and Removing punctuations.

AssignmentIR.py : In this file I have imported the above preprocess.py module and , This file loops through the files1.txt to file999.txt and calls the method in Preprocess to and stored the result back in the folder named as preprocess\_text\_files with same name.

Inverted\_Index.py : In this file I have created the inverted index by using the text file present in preprocess\_text\_files folder and this gives us the inverted index in the form of dictionary where the keys are the unique terms in the document(file) and the values are the posting list which contains the document(file) name where that term is present.

Positional\_Index.py : In this file I have created the Positional\_index using the text\_file present int the proprocess\_text\_files folder and this gives us the positional\_index in form of dictionary where the term is the key and the value is dictionary in itself , The dicationary which is present as a value contains the key as the document(file) name where that term is present and the value is the list which contains the positions where that words is present in the document(file).

QueryInputQ1.py : In this file what we do is we are basically taking the input than we are performing preprocessing on it by performing all the 5 steps as mentioned it starts with i) Lowering the Text, ii)Performing the Tokenization , iii)removing the stop words iv) removing the stopwords v)Removing the blank space tokens , This file has all function for all the above operation we only have to provid the file number and it will give us the preprocessed file.txt and store it in preprocessed\_files folder.

QueryInputQ2.py : It takes the input from the user as a sentence and it also takes the operations as well which can be AND, NOT,AND NOT,OR NOT, than it extracts the posting list of each term in the sentence and performs the above operations on them , Make sure the number of operation is one less than the number of terms in the list.

QueryInputQ3.py : It takes the input from the user as a sentence and it performs the phrase search on the sentence , when it first takes the input it preprocess it by performing all the operation’s on them like Tokenization, Lowering the letters Removing stop words , Removing the blank space tokens, and Removing punctuations, Then it finds all the document which contains all these terms then we return it.

printInvertedIndex.py: There is a python file separately which helps you in printing the inverted index.

## Library Used:-

The library which I have used for this assignment is NLTK and BeautifulSoup:-

NLTK : Provides tools for natural language processing tasks such as Tokenization

Stemming , Tagging , Parsing and Semantic Reasoning.

Beautiful Soup: A Library for parsing HTML and XML documents and extracting data from Them.

Sent\_Tokenize: (from nltk.tokenize) : Tokenizes text into sentences.

Word\_Tokenize: (from nltk.tokenize) : Tokenizes text into words.

stopwords: (from nltk.corpus) Provides a collection of common stop words for english languages.

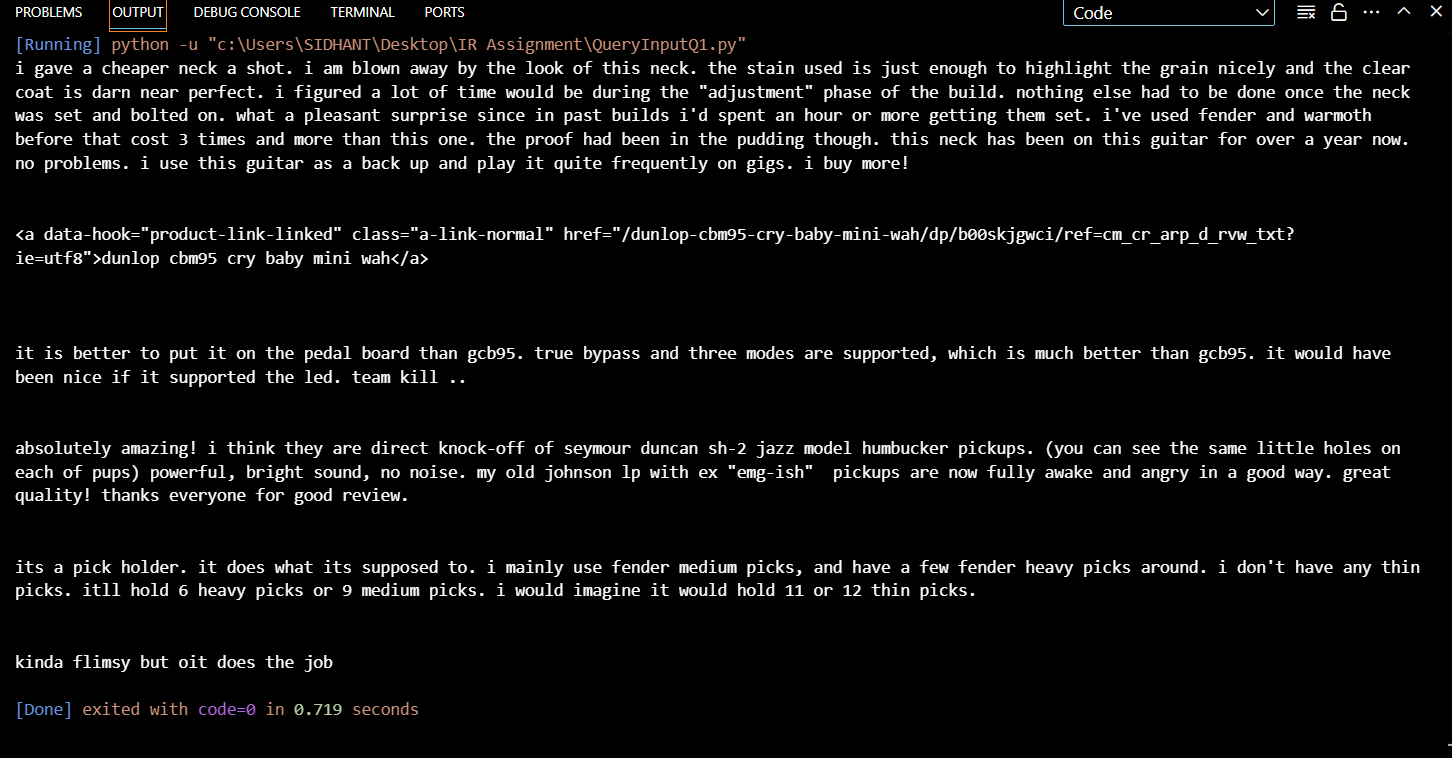
String: provides constant and function for string manipulation.

## WORKING RESULTS OF EACH PROBLEM (with screen shots)

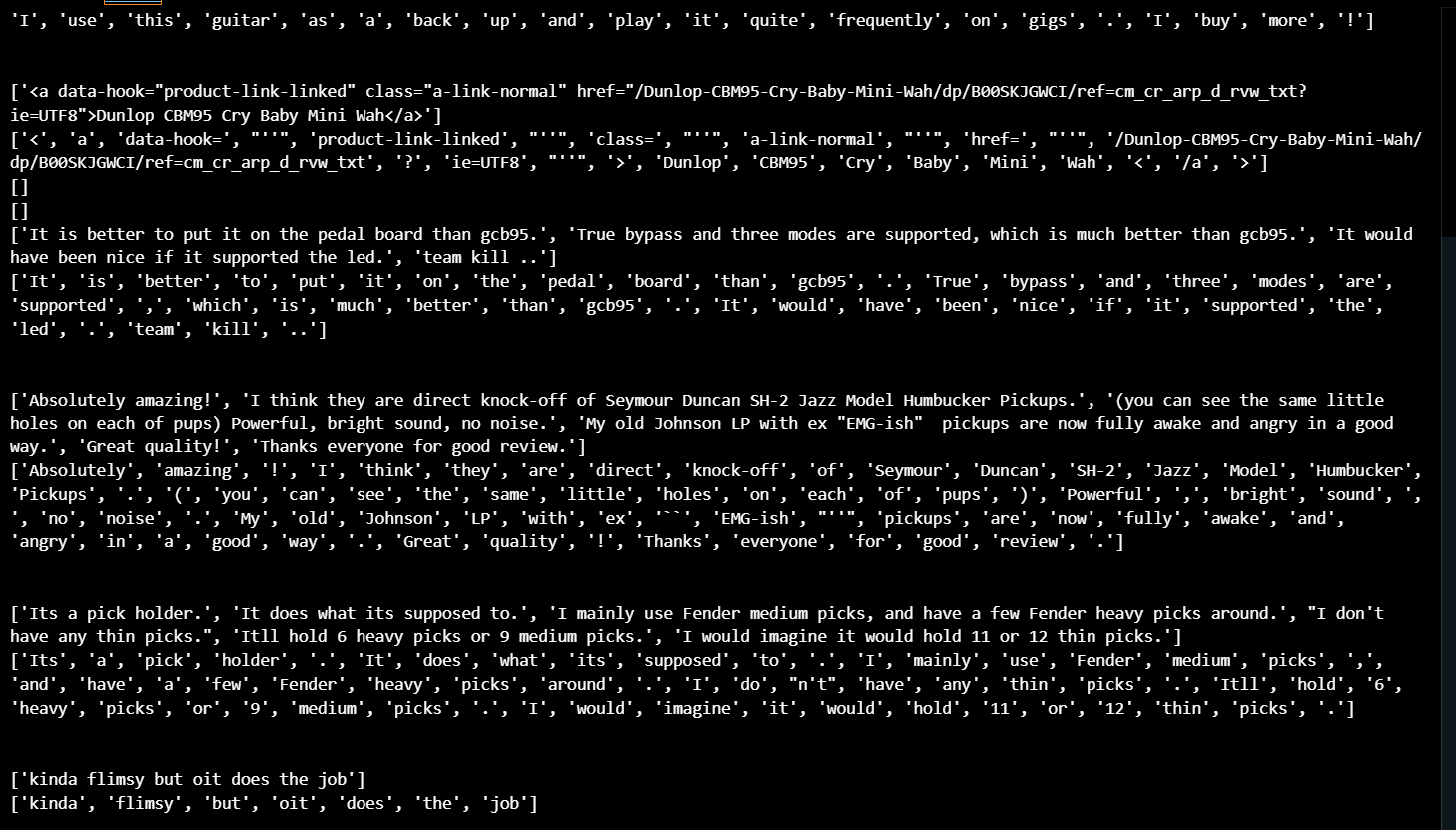
Q1. Here we have to print the contents of five sample file before and after performing each operation’s ?.

Ans:-

## Lowering and Printing it :-

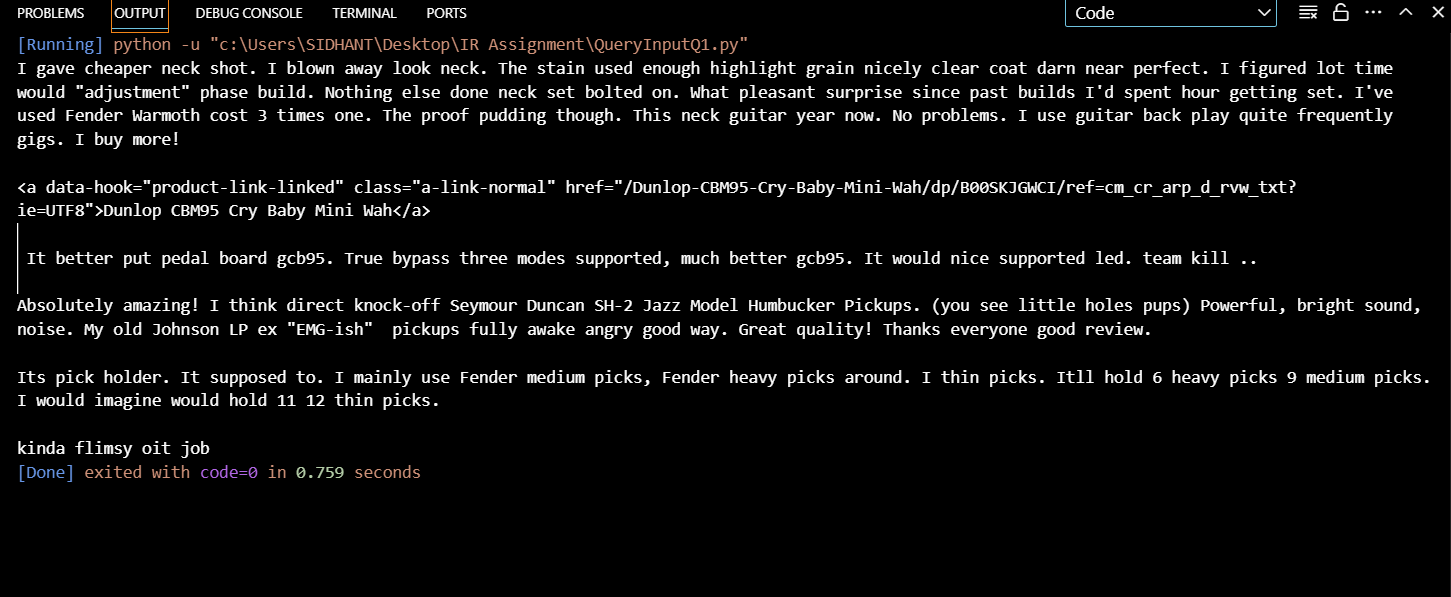


## Perform Tokenization and printing it :-

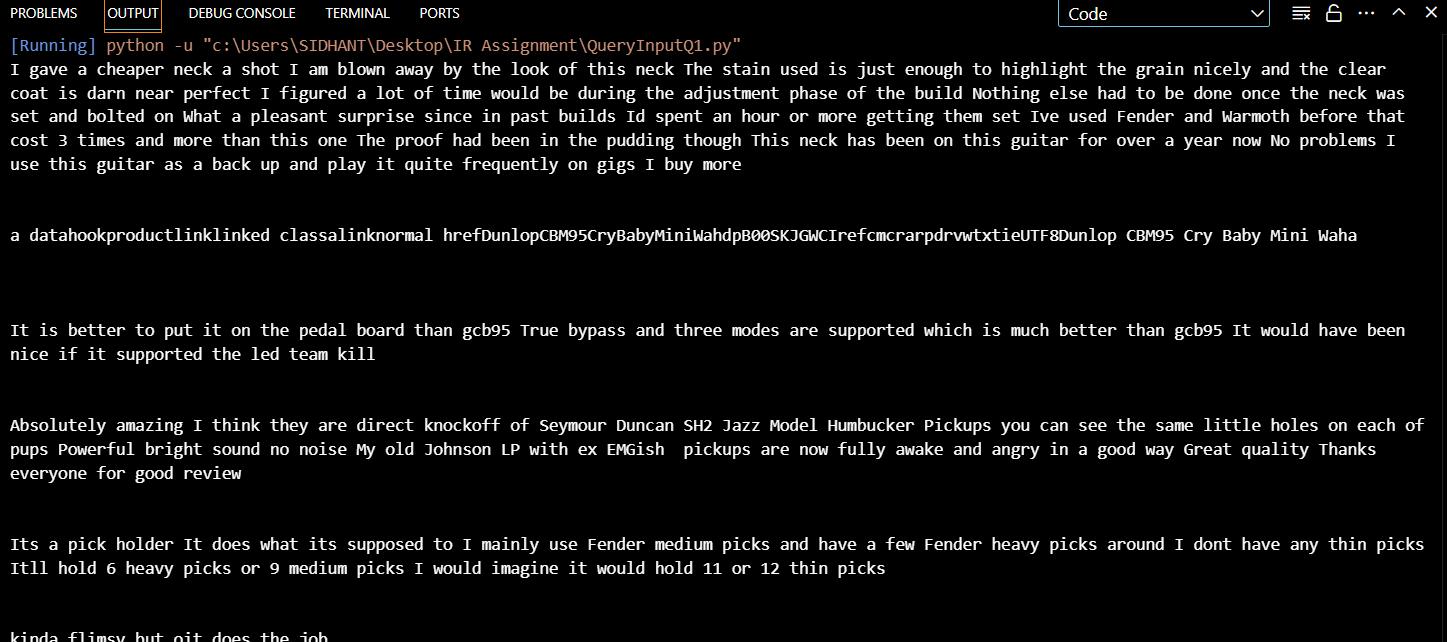


## RemoveStopwords and printing it :-

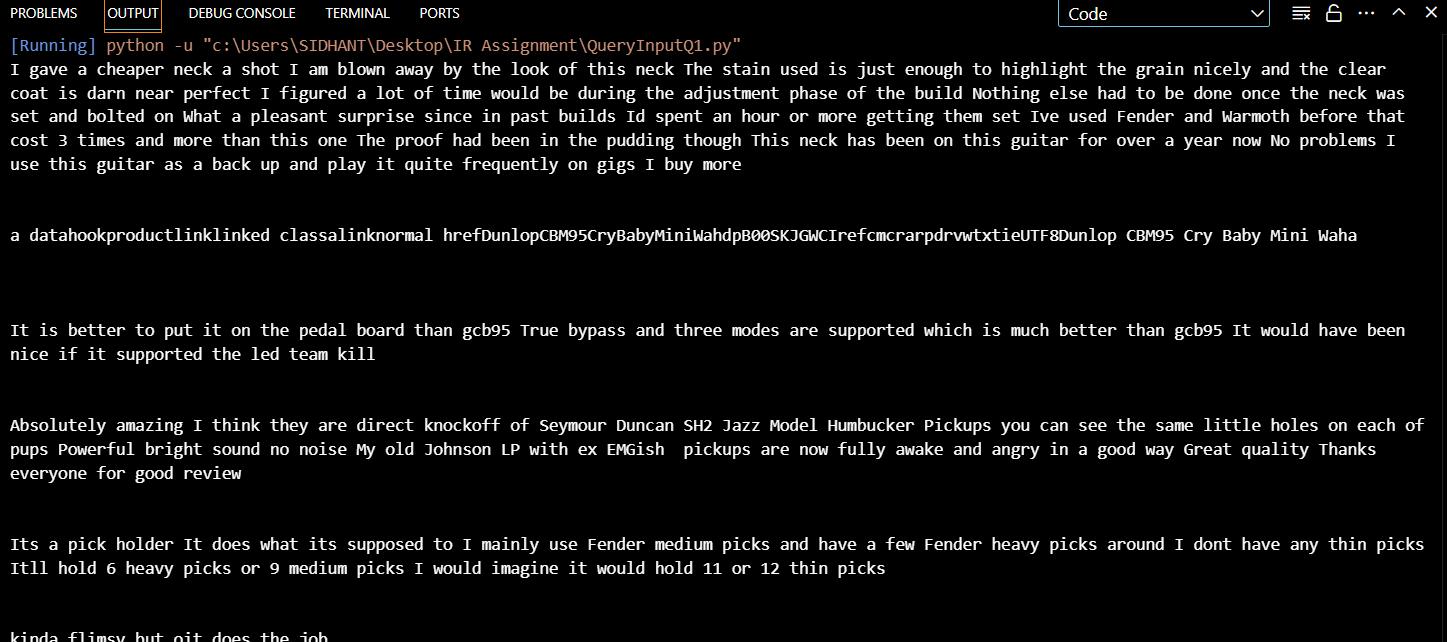
We might find that some stop words are present because the stop words which nltk corpus gives are all in lower case letter so we shoul perform this step after lowering only, just for the stake of question I am doing this.



## Remove Punctuations and printing it :-



## Remove blank space tokens and printing it :-



## Q2. Unigram Inverted Index and Boolean Queries

Input format:-

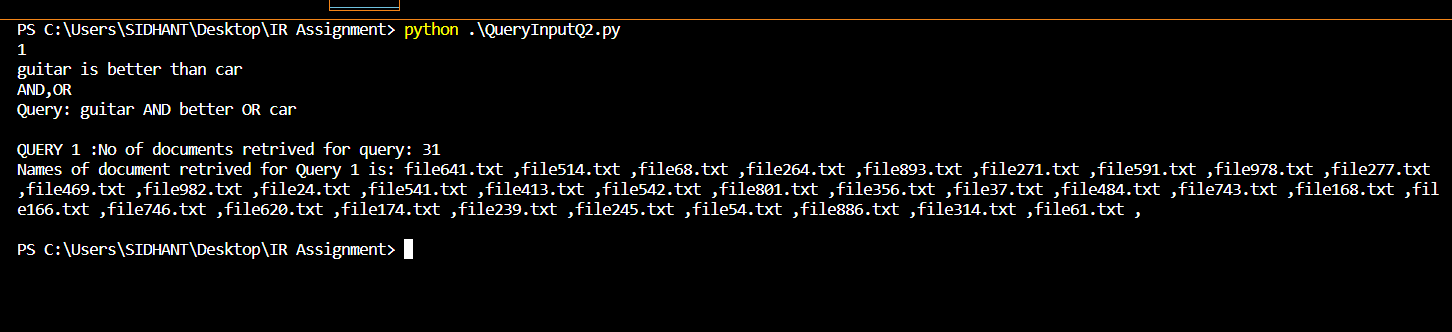
Input format:

a. The first line contains N denoting the number of queries to execute

b. The next 2N lines contain queries in the following format:

i. Input sequence

ii. Operations separated by comma



# Q3. Positional index and queries

a. The first line contains N denoting the number of queries to execute

b. The next N lines contain phrase queries

4. Output Format:

a. 2N lines consisting of the results in the following format:

i. Number of documents retrieved for query X using positional index

ii. Names of documents retrieved for query X using positional index

