Final Project

Group 02

Students Name:

1. Abdullah Alhoraibi

2. Mohamad Meaari

3. Ran Arino

In [1]:
```
1 import os
```

In [2]:
```
1  #This function displays the menu as follows
2  #    1. Indexing
3  #    3. Exit
4
5  def printMenu():
6      print('Menu:')
7      print('Please enter 1 for indexing and 3 to exit')
8      result_menu = int(input('1. Indexzing\n3. Exit\n'))
9      return result_menu
10 printMenu()
```

```
Menu:
Please enter 1 for indexing and 3 to exit
1. Indexzing
3. Exit
1
```

Out[2]: 1

In [3]:

```python
# This function takes a text file as input and replaces all punctuations int
# Input: text
# Output: text with no punctuations

def punctuationsRemoval(text: str) -> str:
    """
    (str) -> str
    Return a text after replacing any punctuations to blank, " ".

    >>>punctuationsRemoval('abcdefg##@@higklmn,*()%opqr""stu')
    'abcdefg    higklmn     opqr  stu'
    """
    PUNCTUATIONS = ["!", "(", ")", "—", "–", "-", "[", "]", "{", "}", ";", "

    clear_text = ""
    for word in text:
        clear_text += word if word not in PUNCTUATIONS else " "
            # if a variable "word" is not punctuation, then append "word" it

    return clear_text

punctuationsRemoval('abcdefg##@@higklmn,*()%opqr""stu')
```

Out[3]: 'abcdefg   higklmn   opqr stu'

In [4]:

```python
# This function takes a text as input and removes all stopwords.
# Input: text
# Output: text with nostop words
def stopWordRemoval(text: str) -> list:
    """
    (str) -> list
    Return word list based on the input data(txt format), which are excluded
    Before running this function, should be removed punctuations from the te

    >>>stopWordRemoval("The monkeys jump on the bed.")
    ['monkeys', 'jump', 'bed.']
    """
    word_list = list(filter(lambda word: word != "", text.split(" ")))
        # excluding "" from the text after splitting by the blnk

    with open('Stop_Words.txt', 'r') as f: # load stopwords file
        Stop_Words = f.read()

    remove_list = [i.strip("'").strip('"') for i in Stop_Words.split(", ")
    clean_list = list(filter(lambda word: word.lower() not in remove_list, w

    return clean_list

stopWordRemoval("The monkeys jump on the bed.")
```

Out[4]: ['monkeys', 'jump', 'bed.']

The following functions takes a clean text as argument, and appends TermDocFreqFile with the list of terms, the document in which they appear and their frequencies. termDocFreqFile format: 3 columns, values separated with space Term doc# freq Ontario 1 5 years 1 1 sugar 1 2

Input: cleanText, and document id output: termDocFreqFile format: 3 columns, values seperated with space

In [5]:

```
def appendTermDocFreq(docid: int, cleanText: str, termDocFreqFile):
    """
    Appends TermDocFreqFile with the term(lowercase), the document number, a
    The format is like below.

    ontario 1 2\n
    government 1 3\n
    """
    term_freq = {} # format -> {term: freqency}
    for word in cleanText:
        word = word.lower() # every word changes lower case
        if word not in term_freq:
            term_freq[word] = 1
        else:
            term_freq[word] += 1

    append_text = ''
    for k, v in term_freq.items():
        append_text += '{} {} {}\n'.format(k, docid, v)

    termDocFreqFile.write(append_text)
```

The following function reads termDocFreqFile line by line and append the global index that go from terms as keys to list of documents that contain them with their frequencies as val. Appending the index works as follows: - for line in termDocFreqFile, - if the term does not exist in index, add the term as key, the value will be a dictionary containg docid:freq as key:val in index - if the term already exists in index, append the val (which is a dictionary) with docid:freq input: termDocFreqFile output: fill in the index structure global variable defined in top of the module.

Note:
In order to read or write the laten alphabets (like é), we set the following statement in open() function: **encoding='utf-8'**.

In [6]:

```
def genIndex(termDocFreqFile):
    index_file = {}
    # Format -> {term_01: {doc#: freq, doc#: freq,...}, term_02:{doc#: freq,

    termDocFreqFile = open("TermDocFreq.txt", 'r', encoding='utf-8')
    for line in termDocFreqFile: # read text document line by line
        read = line[:-1].split(" ") # apply split method after removing the
        if read[0] not in index_file: # read[0], read[1], read[2] = term, do
            index_file[read[0]] = {read[1]: read[2]}
        else:
            index_file[read[0]][read[1]] = read[2]

    return index_file
```

The following function reads all the text files in the folder 'dataset', appends them to a list, and returns the list. Input: None Output: a list if texts

In [7]:

```
def readFolderContent():
    files = []
    file_list = os.listdir('dataset')
    for filename in sorted(file_list):
        with open('dataset' + '/' + filename, 'r', encoding='utf-8') as infi
            files.append(infile.read())
    return files
```

The following function creates necessary files needed in this project. For more information about this function review the flowchart given in the instructions.

In [9]:

```
def indexing():
    termDocFreqFile = open("TermDocFreq.txt", 'w', encoding='utf-8')

    # readFolderContent is called to create a list of files.
    files = readFolderContent()
    id=1
    for file in files:
        puncRemoved = punctuationsRemoval(file) # remove all punctuations
        stopWordsRemoved = stopWordRemoval(puncRemoved) # remove all stop wo
        appendTermDocFreq(id, stopWordsRemoved, termDocFreqFile)  # Call app
        id += 1

    global global_index_file
    global_index_file = genIndex(termDocFreqFile) # Call genIndex function t
    termDocFreqFile.close()

def main():
    option=printMenu()
    if option == 1:
        indexing()


if __name__ == "__main__":
       main()
```

```
Menu:
Please enter 1 for indexing and 3 to exit
1. Indexzing
3. Exit
1
```

In [10]:

```
global_index_file
```

Out[10]:

```
{'ontario': {'1': '2',
  '11': '1',
  '12': '1',
  '14': '1',
  '21': '1',
  '22': '1',
  '25': '1',
  '30': '2',
  '32': '1',
  '35': '2',
  '42': '4',
  '56': '2',
  '65': '6',
  '94': '8',
  '121': '1',
  '130': '4',
  '137': '2',
  '146': '1',
  '168': '2',
```