

```

"""
Rashad Khan
010713326
CS2520-01
Project 3
04/19/2023
"""

#part 1

import re

usrnme = ['lyang', 'kSimon', 'danny', 'tomatcpp', 'csDept', 'CoScpp', 'broncoWins', 'ponyExp',
'BldgAndRooms', 'helloKitty' ]
pswr = ['sheCodes#123', 'catchAllGood1%', '@my2Choices', '123abc;;', 'Hello2Monday$',
'GoodFriday@Cpp2', 'CS2520@Python', 'JavaIsHot2!', '2ManyRainingDays!', '1Startup@Starbucks']

user_dict = dict(zip(usrnme,pswr))
global usr
usr = ""

def check_password(password):
    if len(password) < 8:
        return False
    if not re.search(r"[A-Z]", password):
        return False
    if not re.search(r"[a-z]", password):
        return False
    if not re.search(r"\d", password):
        return False
    if not re.search(r"[!@#$%^&*(),.?\"':{}|<>]", password):
        return False
    return True

def user_login(user_dict):
    global usr
    for i in range (3):
        usr = input("Enter a username: ")
        if usr in user_dict:
            for i in range(3):
                pswd = input("Enter a password: ")
                if pswd == user_dict[usr]:
                    return True
            else:
                print("Invalid password, try again")
            return False
        else:
            print("Invalid username")
    return False

#user_login(user_dict)

def create_user(user_dict):
    global usr
    for i in range(3):
        usr = input("Enter a username: ")
        if usr in user_dict:
            print("username already in use", i+1, "of 3 attempts remaining")
            continue

```

```

    else:
        for i in range(3):
            pswd = input("Enter a password: ")
            if check_password(pswd):
                return {usr : pswd}
            else:
                print("Invalid password", i+1, "of 3 attempts remaining")
        return None
    return None

#user_dict.update(create_user(user_dict))
#print(user_dict)

def update_password(user_dict):
    for i in range (3):
        while user_login(user_dict):
            pswd = input("Enter new password: ")
            if check_password(pswd):
                user_dict[usr] = pswd
                return True
            else:
                print("Invalid password, try again")
                if i == 3:
                    break
                i+=1
                continue
    return False

#update_password(user_dict)

def main():
    while True:
        selection = input("1 : Login\n2 : New User\n3 : Change Password\n4 : Exit\nPlease Choose from
the options above : ")
        match selection:
            case "1":
                if user_login(user_dict):
                    print("Welcome")
                else:
                    print("Invalid")
            case "2":
                temp= create_user(user_dict)
                if temp != None:
                    user_dict.update(temp)
            case "3":
                update_password(user_dict)
            case "4":
                break

main()

"""
1 : Login
2 : New User
3 : Change Password
4 : Exit
Please Choose from the options above : 0
1 : Login
2 : New User
3 : Change Password
4 : Exit
Please Choose from the options above : 1
Enter a username: bad usr

```

```

Invalid username
Enter a username: lyang
Enter a password: bad pswd
Invalid password, try again
Enter a password: sheCodes#123
Welcome
1 : Login
2 : New User
3 : Change Password
4 : Exit
Please Choose from the options above : 2
Enter a username: lyang
username already in use 1 of 3 attempts remaining
Enter a username: rkhan
Enter a password: bad pswd
Invalid password 1 of 3 attempts remaining
Enter a password: Rkh4np4$$w0rd
1 : Login
2 : New User
3 : Change Password
4 : Exit
Please Choose from the options above : 3
Enter a username: bad usr
Invalid username
Enter a username: rkhan
Enter a password: bad pswd
Invalid password, try again
Enter a password: Rkh4np4$$w0rd
Enter new password: Pa$$word123
1 : Login
2 : New User
3 : Change Password
4 : Exit
Please Choose from the options above : 1
Enter a username: rkhan
Enter a password: Pa$$word123
Welcome
1 : Login
2 : New User
3 : Change Password
4 : Exit
Please Choose from the options above : 4

```

```

"""

```

```

#part 2

```

```

from collections import Counter
import re

def fileconversion():
    while True:
        filename = input("Enter a name of a file: ")
        try:
            with open(filename, "r") as file:
                bigList = [re.findall(r'\b[a-zA-Z]+\b', line.lower()) for line in file]
                break
        except FileNotFoundError:
            print("File does not exist")
            continue
    return bigList

```

```

def displayTotalLinesAndWords(List):
    numLines = len(List)
    numWords = sum(len(line) for line in List)
    print(f"Total lines are {numLines}")
    print(f"Total number of words are {numWords}")

def createSetFromList(List):
    wordList = [word for line in List for word in line]
    wordSet = set(wordList)
    return wordSet

def main():
    list1 = fileconversion()
    print("Enter a second file to compare")
    list2 = fileconversion()

    print("Details for list 1")
    displayTotalLinesAndWords(list1)

    print("Details for list 2")
    displayTotalLinesAndWords(list2)

    set1 = createSetFromList(list1)
    set2 = createSetFromList(list2)

    totalWords = set1.union(set2)
    numTotalWords = len(totalWords)
    print(f"{numTotalWords} distinct words used in two articles, and they are: {totalWords}")

    uniqueSet1 = set1.difference(set2)
    numUniqueSet1 = len(uniqueSet1)
    print(f"{numUniqueSet1} distinct words appear in article 1 but not in article 2, and they are: {uniqueSet1}")

    uniqueSet2 = set2.difference(set1)
    numUniqueSet2 = len(uniqueSet2)
    print(f"{numUniqueSet2} distinct words appear in article 2 but not in article 1, and they are: {uniqueSet2}")

    commonWords = set1.intersection(set2)
    numCommonWords = len(commonWords)
    print(f"{numCommonWords} distinct words appear in both articles, and they are: {commonWords}")

    uniqueWords = uniqueSet1.union(uniqueSet2)
    numUniqueWords = len(uniqueWords)
    print(f"{numUniqueWords} distinct words appear in one of the articles only, and they are: {uniqueWords}")

    list3 = (list1+list2)
    dictf = {}
    for words in totalWords:
        dictf[words]=list3[0].count(words)+list3[1].count(words) + list3[2].count(words)

    sortedWords = sorted(dictf.items(), key=lambda x: x[1], reverse=True)

    print(sortedWords)

main()

"""
Enter a name of a file: Python.txt
Enter a second file to compare

```

Enter a name of a file: MachineLearning.txt

Details for list 1

Total lines are 2

Total number of words are 77

Details for list 2

Total lines are 1

Total number of words are 50

76 distinct words used in two articles, and they are: {'level', 'steps', 'statistical', 'get', 'you', 'and', 'as', 'readability', 'computer', 'philosophy', 'design', 'experiences', 'ranks', 'science', 'to', 'general', 'popular', 'machine', 'they', 'will', 'analyses', 'purpose', 'one', 'with', 'techniques', 'give', 'by', 'dive', 'past', 'statistics', 'create', 'program', 'specific', 'successful', 'learns', 'programs', 'learning', 'learn', 'programming', 'predict', 'languages', 'improve', 'that', 'making', 'is', 'studying', 'language', 'artificial', 'step', 'how', 'from', 'tasks', 'into', 'of', 'the', 'necessary', 'code', 'field', 'high', 'consistently', 'a', 'ability', 'data', 'library', 'scikit', 'outcome', 'ready', 'application', 'python', 'uses', 'direction', 'perform', 'most', 'using', 'intelligence', 'world'}

43 distinct words appear in article 1 but not in article 2, and they are: {'level', 'steps', 'statistical', 'you', 'as', 'readability', 'philosophy', 'design', 'experiences', 'language', 'ranks', 'how', 'science', 'tasks', 'general', 'popular', 'they', 'necessary', 'will', 'code', 'field', 'high', 'purpose', 'one', 'consistently', 'with', 'techniques', 'give', 'ability', 'library', 'scikit', 'past', 'application', 'uses', 'create', 'specific', 'successful', 'programs', 'perform', 'programming', 'most', 'languages', 'improve'}

20 distinct words appear in article 2 but not in article 1, and they are: {'get', 'making', 'studying', 'artificial', 'step', 'into', 'analyses', 'data', 'by', 'dive', 'outcome', 'ready', 'statistics', 'program', 'direction', 'learns', 'predict', 'using', 'intelligence', 'world'}

13 distinct words appear in both articles, and they are: {'from', 'to', 'that', 'machine', 'of', 'and', 'the', 'learning', 'learn', 'python', 'computer', 'is', 'a'}

63 distinct words appear in one of the articles only, and they are: {'level', 'steps', 'statistical', 'get', 'you', 'as', 'readability', 'philosophy', 'making', 'design', 'studying', 'experiences', 'language', 'ranks', 'how', 'science', 'artificial', 'tasks', 'step', 'into', 'general', 'popular', 'they', 'world', 'necessary', 'will', 'code', 'field', 'analyses', 'high', 'purpose', 'one', 'consistently', 'with', 'techniques', 'give', 'ability', 'data', 'by', 'dive', 'library', 'scikit', 'outcome', 'past', 'application', 'ready', 'statistics', 'uses', 'create', 'program', 'specific', 'direction', 'successful', 'learns', 'programs', 'perform', 'programming', 'predict', 'most', 'using', 'languages', 'intelligence', 'improve'}

[('the', 9), ('machine', 6), ('learning', 6), ('to', 5), ('is', 5), ('of', 5), ('a', 5), ('python', 5), ('and', 4), ('learn', 4), ('computer', 3), ('programming', 2), ('that', 2), ('from', 2), ('into', 2), ('data', 2), ('level', 1), ('steps', 1), ('statistical', 1), ('get', 1), ('you', 1), ('as', 1), ('readability', 1), ('philosophy', 1), ('design', 1), ('experiences', 1), ('ranks', 1), ('science', 1), ('general', 1), ('popular', 1), ('they', 1), ('will', 1), ('analyses', 1), ('purpose', 1), ('one', 1), ('with', 1), ('techniques', 1), ('give', 1), ('by', 1), ('dive', 1), ('past', 1), ('statistics', 1), ('create', 1), ('program', 1), ('specific', 1), ('successful', 1), ('learns', 1), ('programs', 1), ('predict', 1), ('languages', 1), ('improve', 1), ('making', 1), ('studying', 1), ('language', 1), ('artificial', 1), ('step', 1), ('how', 1), ('tasks', 1), ('necessary', 1), ('code', 1), ('field', 1), ('high', 1), ('consistently', 1), ('ability', 1), ('library', 1), ('scikit', 1), ('outcome', 1), ('ready', 1), ('application', 1), ('uses', 1), ('direction', 1), ('perform', 1), ('most', 1), ('using', 1), ('intelligence', 1), ('world', 1)]

Enter a name of a file: Test1.txt

Enter a second file to compare

Enter a name of a file: Test2.txt

Details for list 1

Total lines are 9

Total number of words are 313

Details for list 2

Total lines are 22

Total number of words are 517

402 distinct words used in two articles, and they are: {'as', 'calls', 'his', 'common', 'instead', 'stable', 'out', 'will', 'van', 'him', 'fluency', 'stemmed', 'body', 'adding', 'when', 'inspired', 'cost', 'cycle', 'security', 'strive', 'possible', 'summarized', 'tools', 'beautiful', 'cython', 'level', 'release', 'cluttered', 'would', 'dynamic', 'its', 'poisoning', 'object', 'can', 'meanings', 'foundation', 'materials', 'functions', 'strives', 'tribute', 'reflect', 's', 'paradigm', 'simpler', 'variable', 'just', 'system', 'november', 'implementation', 'offer', 'aim', 'due', 'is', 'improved', 'sketch', 'announced', 'complex', 'code', 'then', 'rather', 'at', 'versions', 'interfacing',

'potential', 'abc', 'initially', 'unicode', 'structured', 'series', 'alex', 'expressions', 'includes', 'programming', 'fun', 'filter', 'playful', 'elected', 'while', 'also', 'permanent', 'interfaces', 'reference', 'generator', 'preferably', 'project', 'binds', 'increased', 'during', 'five', 'on', 'lisp', 'cache', 'tutorials', 'including', 'ugly', 'amoeba', 'tradition', 'support', 'compact', 'cwi', 'vacation', 'coding', 'dictionaries', 'means', 'modularity', 'long', 'may', 'community', 'steering', 'new', 'binding', 'january', 'typing', 'rosum', 'difficult', 'it', 'later', 'perl', 'often', 'aspect', 'aphorisms', 'oriented', 'philosophy', 'via', 'like', 'memory', 'ported', 'book', 'group', 'functional', 'denial', 'implicit', 'older', 'logic', 'fully', 'resolution', 'do', 'backported', 'well', 'this', 'successor', 'programmable', 'made', 'releases', 'their', 'designed', 'until', 'minimalist', 'fellow', 'attack', 'offers', 'council', 'c', 'use', 'readability', 'methodology', 'are', 'extension', 'clarity', 'reads', 'service', 'show', 'less', 'describe', 'a', 'method', 'direct', 'complicated', 'improvements', 'was', 'exception', 'with', 'into', 'forward', 'software', 'critical', 'execution', 'borrowed', 'speed', 'building', 'there', 'two', 'range', 'bestowed', 'be', 'natural', 'programmer', 'utility', 'future', 'design', 'not', 'used', 'netherlands', 'because', 'clever', 'vision', 'embraces', 'include', 'should', 'to', 'wiskunde', 'decision', 'set', 'zen', 'handling', 'makes', 'which', 'time', 'another', 'end', 'written', 'frustrations', 'leading', 'standard', 'could', 'dictator', 'premature', 'issues', 'joining', 'better', 'features', 'centrum', 'espoused', 'responsibilities', 'collection', 'martelli', 'such', 'style', 'translation', 'developer', 'upon', 'member', 'obvious', 'further', 'functools', 'library', 'languages', 'haskell', 'explicit', 'increases', 'emphasis', 'combination', 'sole', 'existing', 'contract', 'opposite', 'reporting', 'lead', 'x', 'had', 'capable', 'core', 'extensible', 'by', 'name', 'shouldered', 'way', 'non', 'author', 'changes', 'conceived', 'collector', 'major', 'unpythonic', 'benevolent', 'document', 'of', 'one', 'all', 'term', 'avoid', 'pythonic', 'in', 'error', 'currently', 'reflected', 'implement', 'eggs', 'language', 'wide', 'monty', 'or', 'program', 'pypy', 'that', 'only', 'applications', 'title', 'approaches', 'functionality', 'occasionally', 'grammar', 'other', 'giving', 'remote', 'web', 'syntax', 'small', 'spam', 'compliment', 'examples', 'culture', 'particularly', 'commitment', 'large', 'modules', 'were', 'approach', 'simple', 'script', 'neologism', 'wrote', 'more', 'developers', 'reject', 'postponed', 'easily', 'popular', 'contrast', 'informatica', 'british', 'december', 'than', 'python', 'uses', 'comedy', 'multi', 'translates', 'responsibility', 'july', 'supported', 'chief', 'september', 'called', 'idioms', 'conform', 'bar', 'from', 'sets', 'api', 'compiler', 'late', 'metaobjects', 'guido', 'receive', 'counting', 'terms', 'began', 'maker', 'he', 'motto', 'patches', 'cpython', 'notable', 'automates', 'has', 'available', 'interpreter', 'related', 'detecting', 'some', 'the', 'no', 'for', 'pep', 'life', 'metaprogramming', 'four', 'management', 'itertools', 'released', 'active', 'parts', 'setl', 'marginal', 'fixes', 'mapandreduce', 'something', 'rough', 'foo', 'names', 'comprehensions', 'paradigms', 'optimization', 'understand', 'considered', 'highly', 'counts', 'october', 'important', 'transcription', 'list', 'choice', 'and', 'ml', 'extensions', 'garbage', 'operating', 'expedited', 'move', 'many', 'concern'}

118 distinct words appear in article 1 but not in article 2, and they are: {'leading', 'could', 'dictator', 'issues', 'joining', 'centrum', 'responsibilities', 'collection', 'stable', 'out', 'will', 'him', 'body', 'translation', 'developer', 'upon', 'member', 'further', 'inspired', 'security', 'sole', 'possible', 'reporting', 'release', 'lead', 'x', 'had', 'capable', 'poisoning', 'shouldered', 'changes', 'reflect', 'conceived', 'major', 'benevolent', 'system', 'november', 'due', 'improved', 'announced', 'then', 'term', 'potential', 'versions', 'interfacing', 'error', 'initially', 'unicode', 'currently', 'series', 'elected', 'title', 'permanent', 'remote', 'web', 'project', 'increased', 'five', 'cache', 'commitment', 'were', 'amoeba', 'cwi', 'vacation', 'postponed', 'long', 'informatica', 'steering', 'december', 'new', 'january', 'later', 'ported', 'responsibility', 'july', 'chief', 'denial', 'older', 'september', 'backported', 'successor', 'releases', 'until', 'guido', 'receive', 'attack', 'council', 'began', 'maker', 'he', 'notable', 'automates', 'service', 'improvements', 'no', 'exception', 'life', 'forward', 'four', 'released', 'active', 'setl', 'fixes', 'bestowed', 'utility', 'future', 'netherlands', 'because', 'include', 'october', 'wiskunde', 'decision', 'set', 'handling', 'operating', 'expedited', 'end', 'concern'}

220 distinct words appear in article 2 but not in article 1, and they are: {'standard', 'premature', 'calls', 'better', 'espoused', 'common', 'instead', 'martelli', 'fluency', 'stemmed', 'style', 'adding', 'obvious', 'functools', 'cost', 'library', 'languages', 'haskell', 'explicit', 'increases', 'emphasis', 'combination', 'strive', 'summarized', 'contract', 'tools', 'beautiful', 'cython', 'level', 'cluttered', 'extensible', 'dynamic', 'object', 'name', 'can', 'way', 'meanings', 'foundation', 'materials', 'functions', 'strives', 'author', 'non', 'tribute', 'collector', 'paradigm', 'unpythonic', 'simpler', 'variable', 'document', 'just', 'sketch', 'offer', 'aim', 'complex', 'rather', 'one', 'avoid', 'pythonic', 'structured', 'reflected', 'implement', 'eggs', 'alex', 'expressions', 'includes', 'playful', 'fun', 'filter', 'monty', 'wide', 'pypy', 'approaches', 'applications', 'while', 'functionality', 'also', 'occasionally', 'grammar', 'giving', 'syntax', 'interfaces', 'small', 'generator', 'spam', 'preferably', 'compliment', 'binds', 'examples', 'during', 'culture', 'lisp', 'particularly', 'tutorials', 'modules', 'approach', 'simple', 'ugly', 'tradition', 'compact', 'script', 'neologism', 'wrote', 'more', 'reject', 'coding', 'dictionaries', 'means', 'modularity', 'contrast', 'popular', 'binding', 'british', 'typing', 'difficult', 'than',

'perl', 'uses', 'often', 'aspect', 'aphorisms', 'oriented', 'philosophy', 'via', 'multi', 'comedy', 'memory', 'translates', 'like', 'book', 'group', 'functional', 'implicit', 'logic', 'fully', 'resolution', 'do', 'called', 'idioms', 'conform', 'bar', 'well', 'this', 'programmable', 'sets', 'api', 'their', 'compiler', 'metaobjects', 'designed', 'minimalist', 'fellow', 'offers', 'terms', 'c', 'use', 'readability', 'methodology', 'motto', 'cpython', 'extension', 'clarity', 'reads', 'has', 'available', 'show', 'interpreter', 'related', 'less', 'describe', 'method', 'direct', 'some', 'complicated', 'pep', 'into', 'metaprogramming', 'software', 'critical', 'management', 'itertools', 'borrowed', 'building', 'parts', 'there', 'two', 'marginal', 'range', 'mapandreduce', 'something', 'rough', 'natural', 'programmer', 'design', 'foo', 'names', 'paradigms', 'used', 'optimization', 'clever', 'vision', 'considered', 'embraces', 'highly', 'understand', 'counts', 'should', 'important', 'transcription', 'zen', 'choice', 'makes', 'ml', 'extensions', 'time', 'move', 'another', 'opposite', 'written', 'frustrations']

64 distinct words appear in both articles, and they are: {'as', 'the', 'was', 'for', 'programming', 'language', 'with', 'features', 'or', 'program', 'his', 'that', 'only', 'van', 'such', 'execution', 'when', 'supported', 'other', 'speed', 'cycle', 'reference', 'existing', 'from', 'on', 'made', 'be', 'would', 'large', 'core', 'late', 'by', 'not', 'its', 'including', 'comprehensions', 'support', 'python', 'developers', 'counting', 's', 'to', 'easily', 'are', 'of', 'patches', 'a', 'implementation', 'list', 'and', 'is', 'may', 'which', 'community', 'garbage', 'code', 'all', 'at', 'rosum', 'many', 'detecting', 'in', 'abc', 'it'}

338 distinct words appear in one of the articles only, and they are: {'calls', 'common', 'instead', 'stable', 'out', 'will', 'fluency', 'him', 'stemmed', 'body', 'adding', 'inspired', 'cost', 'security', 'strive', 'possible', 'summarized', 'tools', 'beautiful', 'cython', 'level', 'release', 'cluttered', 'dynamic', 'poisoning', 'object', 'can', 'meanings', 'foundation', 'materials', 'functions', 'strives', 'tribute', 'reflect', 'paradigm', 'simpler', 'variable', 'just', 'system', 'november', 'sketch', 'offer', 'aim', 'due', 'improved', 'complex', 'announced', 'then', 'rather', 'versions', 'potential', 'interfacing', 'initially', 'unicode', 'structured', 'series', 'alex', 'expressions', 'includes', 'playful', 'fun', 'filter', 'elected', 'while', 'also', 'permanent', 'interfaces', 'generator', 'preferably', 'project', 'binds', 'increased', 'during', 'five', 'lisp', 'cache', 'tutorials', 'ugly', 'amoeba', 'tradition', 'compact', 'cwi', 'vacation', 'coding', 'dictionaries', 'means', 'modularity', 'long', 'steering', 'new', 'binding', 'january', 'typing', 'difficult', 'later', 'perl', 'often', 'aspect', 'aphorisms', 'oriented', 'philosophy', 'via', 'like', 'memory', 'ported', 'book', 'group', 'functional', 'denial', 'implicit', 'older', 'logic', 'fully', 'resolution', 'do', 'backported', 'well', 'this', 'successor', 'programmable', 'releases', 'their', 'designed', 'until', 'minimalist', 'fellow', 'attack', 'offers', 'council', 'c', 'use', 'readability', 'methodology', 'extension', 'clarity', 'reads', 'service', 'show', 'less', 'describe', 'method', 'direct', 'complicated', 'improvements', 'exception', 'into', 'forward', 'software', 'critical', 'borrowed', 'building', 'there', 'two', 'range', 'bestowed', 'natural', 'programmer', 'utility', 'future', 'design', 'used', 'netherlands', 'because', 'clever', 'vision', 'embraces', 'include', 'should', 'wiskunde', 'decision', 'set', 'zen', 'handling', 'makes', 'time', 'another', 'end', 'written', 'frustrations', 'leading', 'standard', 'could', 'dictator', 'premature', 'issues', 'joining', 'better', 'centrum', 'espoused', 'responsibilities', 'collection', 'martelli', 'style', 'translation', 'developer', 'upon', 'member', 'obvious', 'further', 'functools', 'library', 'languages', 'haskell', 'explicit', 'increases', 'emphasis', 'combination', 'sole', 'contract', 'reporting', 'lead', 'x', 'had', 'capable', 'extensible', 'name', 'shouldered', 'way', 'non', 'author', 'changes', 'conceived', 'collector', 'major', 'unpythonic', 'benevolent', 'document', 'one', 'term', 'avoid', 'pythonic', 'error', 'currently', 'reflected', 'implement', 'eggs', 'wide', 'monty', 'pypy', 'approaches', 'applications', 'title', 'functionality', 'occasionally', 'grammar', 'giving', 'remote', 'web', 'syntax', 'small', 'spam', 'compliment', 'examples', 'culture', 'particularly', 'commitment', 'modules', 'were', 'approach', 'simple', 'script', 'neologism', 'wrote', 'more', 'reject', 'postponed', 'contrast', 'popular', 'informatica', 'british', 'december', 'than', 'uses', 'comedy', 'multi', 'translates', 'responsibility', 'july', 'chief', 'september', 'called', 'idioms', 'conform', 'bar', 'sets', 'api', 'compiler', 'metaobjects', 'guido', 'receive', 'terms', 'began', 'maker', 'he', 'motto', 'cpython', 'notable', 'automates', 'has', 'available', 'interpreter', 'related', 'some', 'no', 'pep', 'life', 'metaprogramming', 'four', 'management', 'itertools', 'released', 'active', 'parts', 'setl', 'marginal', 'fixes', 'mapandreduce', 'something', 'rough', 'foo', 'names', 'paradigms', 'optimization', 'understand', 'considered', 'highly', 'counts', 'october', 'important', 'transcription', 'choice', 'ml', 'extensions', 'operating', 'expedited', 'move', 'opposite', 'concern'}

[('the', 11), ('python', 10), ('as', 5), ('to', 5), ('of', 4), ('in', 4), ('his', 3), ('project', 3), ('a', 3), ('was', 3), ('with', 3), ('and', 3), ('van', 2), ('its', 2), ('s', 2), ('on', 2), ('rosum', 2), ('which', 2), ('features', 2), ('lead', 2), ('x', 2), ('by', 2), ('major', 2), ('december', 2), ('for', 2), ('released', 2), ('many', 2), ('him', 1), ('when', 1), ('inspired', 1), ('cycle', 1), ('reflect', 1), ('system', 1), ('implementation', 1), ('announced', 1), ('code', 1), ('at', 1), ('interfacing', 1), ('abc', 1), ('unicode', 1), ('programming', 1), ('elected', 1), ('permanent', 1), ('reference', 1), ('five', 1), ('amoeba', 1), ('support', 1), ('cwi', 1),

('vacation', 1), ('long', 1), ('community', 1), ('steering', 1), ('new', 1), ('january', 1), ('backported', 1), ('successor', 1), ('releases', 1), ('until', 1), ('council', 1), ('exception', 1), ('bestowed', 1), ('utility', 1), ('netherlands', 1), ('include', 1), ('wiskunde', 1), ('decision', 1), ('handling', 1), ('dictator', 1), ('centrum', 1), ('responsibilities', 1), ('collection', 1), ('such', 1), ('translation', 1), ('developer', 1), ('upon', 1), ('member', 1), ('sole', 1), ('capable', 1), ('core', 1), ('shouldered', 1), ('conceived', 1), ('benevolent', 1), ('term', 1), ('language', 1), ('title', 1), ('commitment', 1), ('developers', 1), ('informatica', 1), ('responsibility', 1), ('july', 1), ('chief', 1), ('from', 1), ('late', 1), ('guido', 1), ('counting', 1), ('began', 1), ('maker', 1), ('he', 1), ('automates', 1), ('detecting', 1), ('life', 1), ('active', 1), ('setl', 1), ('comprehensions', 1), ('october', 1), ('list', 1), ('garbage', 1), ('operating', 1), ('calls', 0), ('common', 0), ('instead', 0), ('stable', 0), ('out', 0), ('will', 0), ('fluency', 0), ('stemmed', 0), ('body', 0), ('adding', 0), ('cost', 0), ('security', 0), ('strive', 0), ('possible', 0), ('summarized', 0), ('tools', 0), ('beautiful', 0), ('cython', 0), ('level', 0), ('release', 0), ('cluttered', 0), ('would', 0), ('dynamic', 0), ('poisoning', 0), ('object', 0), ('can', 0), ('meanings', 0), ('foundation', 0), ('materials', 0), ('functions', 0), ('strives', 0), ('tribute', 0), ('paradigm', 0), ('simpler', 0), ('variable', 0), ('just', 0), ('november', 0), ('offer', 0), ('aim', 0), ('due', 0), ('is', 0), ('improved', 0), ('sketch', 0), ('complex', 0), ('then', 0), ('rather', 0), ('versions', 0), ('potential', 0), ('initially', 0), ('structured', 0), ('series', 0), ('alex', 0), ('expressions', 0), ('includes', 0), ('fun', 0), ('filter', 0), ('playful', 0), ('while', 0), ('also', 0), ('interfaces', 0), ('generator', 0), ('preferably', 0), ('binds', 0), ('increased', 0), ('during', 0), ('lisp', 0), ('cache', 0), ('tutorials', 0), ('including', 0), ('ugly', 0), ('tradition', 0), ('compact', 0), ('coding', 0), ('dictionaries', 0), ('means', 0), ('modularity', 0), ('may', 0), ('binding', 0), ('typing', 0), ('difficult', 0), ('it', 0), ('later', 0), ('perl', 0), ('often', 0), ('aspect', 0), ('aphorisms', 0), ('oriented', 0), ('philosophy', 0), ('via', 0), ('like', 0), ('memory', 0), ('ported', 0), ('book', 0), ('group', 0), ('functional', 0), ('denial', 0), ('implicit', 0), ('older', 0), ('logic', 0), ('fully', 0), ('resolution', 0), ('do', 0), ('well', 0), ('this', 0), ('programmable', 0), ('made', 0), ('their', 0), ('designed', 0), ('minimalist', 0), ('fellow', 0), ('attack', 0), ('offers', 0), ('c', 0), ('use', 0), ('readability', 0), ('methodology', 0), ('are', 0), ('extension', 0), ('clarity', 0), ('reads', 0), ('service', 0), ('show', 0), ('less', 0), ('describe', 0), ('method', 0), ('direct', 0), ('complicated', 0), ('improvements', 0), ('into', 0), ('forward', 0), ('software', 0), ('critical', 0), ('execution', 0), ('borrowed', 0), ('speed', 0), ('building', 0), ('there', 0), ('two', 0), ('range', 0), ('be', 0), ('natural', 0), ('programmer', 0), ('future', 0), ('design', 0), ('not', 0), ('used', 0), ('because', 0), ('clever', 0), ('vision', 0), ('embraces', 0), ('should', 0), ('set', 0), ('zen', 0), ('makes', 0), ('time', 0), ('another', 0), ('end', 0), ('written', 0), ('frustrations', 0), ('leading', 0), ('standard', 0), ('could', 0), ('premature', 0), ('issues', 0), ('joining', 0), ('better', 0), ('espoused', 0), ('martelli', 0), ('style', 0), ('obvious', 0), ('further', 0), ('functools', 0), ('library', 0), ('languages', 0), ('haskell', 0), ('explicit', 0), ('increases', 0), ('emphasis', 0), ('combination', 0), ('existing', 0), ('contract', 0), ('opposite', 0), ('reporting', 0), ('had', 0), ('extensible', 0), ('name', 0), ('way', 0), ('non', 0), ('author', 0), ('changes', 0), ('collector', 0), ('unpythonic', 0), ('document', 0), ('one', 0), ('all', 0), ('avoid', 0), ('pythonic', 0), ('error', 0), ('currently', 0), ('reflected', 0), ('implement', 0), ('eggs', 0), ('wide', 0), ('monty', 0), ('or', 0), ('program', 0), ('pypy', 0), ('that', 0), ('only', 0), ('applications', 0), ('approaches', 0), ('functionality', 0), ('occasionally', 0), ('grammar', 0), ('other', 0), ('giving', 0), ('remote', 0), ('web', 0), ('syntax', 0), ('small', 0), ('spam', 0), ('compliment', 0), ('examples', 0), ('culture', 0), ('particularly', 0), ('large', 0), ('modules', 0), ('were', 0), ('approach', 0), ('simple', 0), ('script', 0), ('neologism', 0), ('wrote', 0), ('more', 0), ('reject', 0), ('postponed', 0), ('easily', 0), ('popular', 0), ('contrast', 0), ('british', 0), ('than', 0), ('uses', 0), ('comedy', 0), ('multi', 0), ('translates', 0), ('supported', 0), ('september', 0), ('called', 0), ('idioms', 0), ('conform', 0), ('bar', 0), ('sets', 0), ('api', 0), ('compiler', 0), ('metaobjects', 0), ('receive', 0), ('terms', 0), ('motto', 0), ('patches', 0), ('cpython', 0), ('notable', 0), ('has', 0), ('available', 0), ('interpreter', 0), ('related', 0), ('some', 0), ('no', 0), ('pep', 0), ('metaprogramming', 0), ('four', 0), ('management', 0), ('itertools', 0), ('parts', 0), ('marginal', 0), ('fixes', 0), ('mapandreduce', 0), ('something', 0), ('rough', 0), ('foo', 0), ('names', 0), ('paradigms', 0), ('optimization', 0), ('understand', 0), ('considered', 0), ('highly', 0), ('counts', 0), ('important', 0), ('transcription', 0), ('choice', 0), ('ml', 0), ('extensions', 0), ('expedited', 0), ('move', 0), ('concern', 0)]

""