Extract\_words\_from\_file:

reading the whole book into a string.

Split the string into a sequence of words.

Find\_element:

Repeat:

Compare every word in the word\_list with item:

If there is a word as same as item

Then

Return true, item was found in the list

Until all the words are compared

Return False, item is not found in the list

Text\_to\_words:

Repeat:

Convert the upper cased letter of each text in a text\_ list into lower cased letter

Split the text into a sequence of words

Until all the words in the list are checked

Returen the word\_list

Uniquelize:

Join a list with space into a string

Split the string, in order to remove the possible space in the list of words

Turn the list into a set, and turn the set back to a list, in order to remove the possible duplicate words in the list

Return a word list

Word\_not\_in\_count:

Extract\_words\_from\_file，to obtain a text\_list from the book file

Text\_to\_words，to obtain a word\_list

Uniquelize, to remove the duplicate words in word\_list

Extract\_words\_from\_file，obtain a vocabulary\_list from the vocabulary file

Repeat:

Try to compare each word in the word\_list with the words in vocabulary\_list

If the vocabulary\_list does not contain the word

Then

Add this word into a list of words\_not\_in

Until all the words in the word\_list are checked

Return the number of the words that are not in the vocabulary

Main program:

Set book file name to file

Set vocabulary file name to vocab

Extract\_words\_from\_file，to obtain a text\_list from the book file

Text\_to\_words，to obtain a word\_list

To obtain the words count of the book by measure the length of the word\_list

Extract\_words\_from\_file，to obtain a text\_list from the book file

Text\_to\_words，to obtain a word\_list

Univquelize, to remove duplicate words in the word\_list

To obtain the exact number of words appeared in the book by measure the length of the word\_list after removing duplication

Word\_not\_in\_count, to obtain the number of words that are not in the vocabulary.

Def Extract\_words\_from\_file(filename)

File 🡨 open(filename)

File\_content 🡨 read（File）

Close File

Words 🡨 File\_content split

return words

Def Find\_element(alist, item)

Repeat until len(alist) <=1

If the No. int(len(alist)/2) word = item

Then

Return true

If the No. int(len(alist)/2) word < item

Then

Alist 🡨 the second half of Alist

Continue

If the No. int(len(alist)/2) word > item

Then

Alist 🡨 the first half of Alist

Continue

Def Text\_to\_words(text\_list)

Words 🡨 []

Repeat, Until all the words in the list are checked

i 🡨 the i\_th element of text\_list

My\_substitutions 🡨 i.maketrans(“ABCD…”\\”abcd…”

Cleaned\_text 🡨 i.translate(my\_substtutions)

Wds 🡨 split cleaned\_text

Words 🡨 words + wds

Returen the word\_list

Def Uniquelize(alist)

Alist 🡨 split (Join alist with space)

Alist 🡨 list(set(alist)

Return alist

Def Word\_not\_in\_count(file, vocab)

Words 🡨 uniquelize(texts\_to\_words(extract\_words\_from\_file(file)))

Words\_not\_in 🡨 []

Vocab\_list 🡨 (extract\_words\_from\_file(vocab)

Repeat Until all the words in the word\_list are checked

i 🡨 the i\_th element of words

if not find\_element(vocab, i)

then

append i to words\_not\_in

Return len(words\_not\_in)

Main program:

file 🡨 file name

vocab 🡨 vocabulary file name

the words count of the book 🡨 len(text\_to\_words(extract\_words\_from\_file(file)))

the exact number of words appeared in the book 🡨

len(uniquelie(text\_to\_words(extract\_words\_from\_file(file))))

number of words that are not in the vocabulary 🡨 words\_not\_in\_count(file, vocab).