

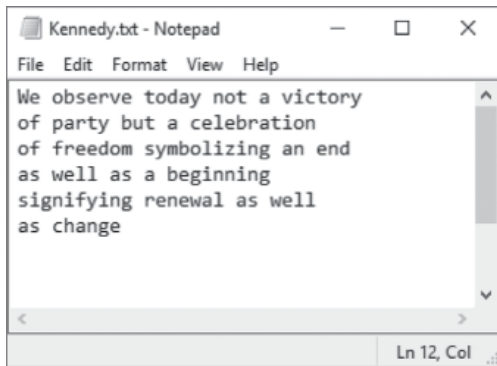
Lab 8 Dictionaries and Sets

1. Word Index

Write a program that reads the contents of a text file. The program should create a dictionary in which the key-value pairs are described as follows:

- Key. The keys are the individual words found in the file.
- Values. Each value is a list that contains the line numbers in the file where the word (the key) is found.

For example, suppose the word “robot” is found in lines 7, 18, 94, and 138. The dictionary would contain an element in which the key was the string “robot”, and the value was a list containing the numbers 7, 18, 94, and 138. Once the dictionary is built, the program should create another text file, known as a word index, listing the contents of the dictionary. The word index file should contain an alphabetical listing of the words that are stored as keys in the dictionary, along with the line numbers where the words appear in the original file. The below figure shows an example of an original text file (Kennedy.txt) and its index file (index.txt).



```
Kennedy.txt - Notepad
File Edit Format View Help
We observe today not a victory
of party but a celebration
of freedom symbolizing an end
as well as a beginning
signifying renewal as well
as change
Ln 12, Col 1
```



```
index.txt - Notepad
File Edit Format View Help
We: 1
a: 1 2 4
an: 3
as: 4 5 6
beginning: 4
but: 2
celebration: 2
change: 6
end: 3
freedom: 3
not: 1
observe: 1
of: 2 3
party: 2
renewal: 5
signifying: 5
symbolizing: 3
today: 1
victory: 1
well: 4 5
Ln 21, Col 1
```

Example

budapest.txt

```
My house in Budapest
My hidden treasure chest
Golden grand piano
My beautiful Castillo
You
You
I'd leave it all
```

Screen

Enter a file name: budapest.txt

index.txt

Budapest: 1
Castillo: 4
Golden: 3
I'd: 7
My: 1 2 4
You: 5 6
all: 7
beautiful: 4
chest: 2
grand: 3
hidden: 2
house: 1
in: 1
it: 7
leave: 7
piano: 3
treasure: 2

Source: Gaddis, T., & Agarwal, R.. Starting out with Python. Pearson

2. File Analysis

Write a program that reads the contents of two text files and compares them in the following ways:

- It should display a list of all the unique words contained in both files.
- It should display a list of the words that appear in both files.
- It should display a list of the words that appear in the first file but not the second.
- It should display a list of the words that appear in the second file but not the first.
- It should display a list of the words that appear in either the first or second file, but not both.

Output example

```
Enter file 1: a_thousand_miles.txt
Enter file 2: fall_for_you.txt
-----A list of all the unique words contained in both files-----
tonight you a the I fall see me

-----A list of the words that appear in both files-----
swear day girl You're tonight just If my miles true
fall make change over mind you will live a would
I impossible think know me by? it's won't another be
Because the again That into I'd Don't could sky Or
Do night like is thousand see time pass 'Cause

-----A list of the words that appear in the first file but not the second-----
I'd could sky walk just If would miles Do think
thousand time into pass by? 'Cause

-----A list of the words that appear in the second file but not the first-----
swear day girl You're my true for make change over
will live to impossible it's won't another be find Because
That Don't Or night like is

-----A list of the words that appear in either the first or second file, but not both-----
swear day girl You're just my If true miles for
change over mind will live would to impossible think know
it's won't another be find Because again That into I'd
Or could sky walk Do night like is thousand time
'Cause
```

Source: Gaddis, T., & Agarwal, R.. Starting out with Python. Pearson