

**Visvesvaraya Technological University  
Belagavi-590018, Karnataka**



A Mini Project Report on

**“Document Processing using keywords”**

Submitted in partial fulfilment of the requirement for the  
File Structures Laboratory with mini project [17ISL68]

**Bachelor of Engineering  
In  
Information Science and Engineering**

Submitted by  
**LAVANYA G[1JT17IS018]**  
Under the guidance of Mr.Vadiraja  
Asst.Professor,Department of ISE



**Department of Information Science and Engineering  
Jyothy Institute of Technology  
Tataguni, Bengaluru-560082**

# Jyothy Institute of Technology

Tataguni, Bengaluru-560082

## Department of Information Science and Engineering



### CERTIFICATE

Certified that the mini project work entitled “**Document processing using keywords**” carried out by **Lavanya G [1JT17IS018]** bonafide student of Jyothy Institute of Technology, in partial fulfilment for the award of **Bachelor of Engineering in Information Science and Engineering** department of the **Vishvesvaraya Technological University, Belagavi** during the year **2019-2020**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said Degree.

Guide, ProfessorGuide, Asst. Professor  
Dept. Of ISE Dept. Of ISE

Professor and HoD  
Dept. Of ISE

External Viva Examiner

- 1.
- 2.

Signature with Date :

# ACKNOWLEDGEMENT

Firstly, we are very grateful to this esteemed institution “**Jyothy Institute of Technology**” for providing us an opportunity to complete our project.

We express our sincere thanks to our Principal **Dr. Gopalakrishna K** for providing us with adequate facilities to undertake this project.

We would like to thank **Dr. Harshwardhan Tiwari, Professor and Head** of Information Science and Engineering Department for providing for his valuable support.

We would like to thank our guides **Mr.Vadiraja, Asst. Prof.** for her keen interest and guidance in preparing this work.

Finally, we would thank all our friends who have helped us directly or indirectly in this project.

Lavanya G [1JT17IS018]

## **ABSTRACT**

Document processing using keywords is to perform operations on a text file. A text file is uploaded, and we are performing search, modify, indexing and annotation operations on that file. It helps the user to find a word, replace, find index value and retrieve the meaning of the given word. It searches fastly. As annotation is implemented, it helps the user to get the meaning. The web browser is also imported, so the user can directly connect to the google. Hence, the information can be accessed easily with the website they want to.

## TABLE OF CONTENTS

| Sl. No | Description                         | Page No. |
|--------|-------------------------------------|----------|
|        | Chapter 1                           |          |
| 1      | <b>Introduction</b>                 |          |
|        | 1.1 Introduction to file structures | 1        |
|        | 1.2 Introduction to file systems    | 1        |
|        | 1.3 Introduction to python          | 1-2      |
|        | 1.4 Document processing             | 2        |
| 2      | <b>Design</b>                       |          |
|        | 2.1 Domain understanding            | 3        |
|        | 2.2 Requirements                    | 3        |
|        | 2.3 System analysis                 | 3        |
|        | 2.4 Block diagram                   | 4        |
|        |                                     |          |
| 3      | <b>Implementation</b>               | 5-8      |
|        |                                     |          |
|        |                                     |          |
| 4      | <b>Result and snapshots</b>         | 9-11     |
| 5      | <b>Analysis</b>                     | 12-13    |
| 6      | <b>Conclusion and future scope</b>  | 13       |
| 7      | <b>References</b>                   | 14       |

# ***CHAPTER 1***

## ***INTRODUCTION***

## **1.1 Introduction to file structures**

File Structures is the Organization of Data in Secondary Storage Device in such a way that minimize the access time and the storage space. It is a combination of representations for data in files and of operations for accessing the data. A File Structure allows applications to read, write and modify data. It might also support finding the data that matches some search criteria or reading through the data in some particular order.

But there is one important distinction that must be made at the outset when discussing file structures. And that is the difference between the logical and physical organization of the data.

On the whole a file structure will specify the logical structure of the data, that is the relationships that will exist between data items independently of the way in which these relationships may actually be realized within any computer. It is this logical aspect that we will concentrate on. The physical organization is much more concerned with optimizing the use of the storage medium when a particular logical structure is stored on, or in it. Typically for every unit of physical store there will be a number of units of the logical structure to be stored in it.

## **1.2 File systems**

A file system controls how data is stored and retrieved. Without a file system, data placed in a storage medium would be one large body of data with no way to tell where one block of data stops and the next begins. By separating the data into blocks and giving each block a name, the data is easily divided and identified. Taking its name from the way paper-based data management system is named, each group of data is called a "file". The structure and logic rules used to manage the groups of data and their names is called a "file system".

There are many different kinds of file systems. Each one has different structure and logic, properties of speed, flexibility, security, size and more.

## **1.3 Introduction to python**

Python is an interpreted, high-level, general-purpose programming language. It supports multiple programming paradigms, including structured ,object-oriented, and functional programming.

Like C++ and Java, Python is case sensitive so "a" and "A" are different variables. The end of a line marks the end of a statement, so unlike C++ and Java, Python does not require a semicolon at the end of each statement. Comments begin with a '#' and extend to the end of the line.

Python source files use the ".py" extension and are called "modules." Python feature is that the whitespace indentation of a piece of code affects its meaning. One unusual Python feature is that the whitespace indentation of a piece of code affects its meaning.

- **1.31 Basic file operations and modes using python**

The modes are:

- 'r' – Read mode which is used when the file is only being read
- 'w' – Write mode which is used to edit and write new information to the file
- 'a' – Appending mode, which is used to add new data to the end of the file; that is new information is automatically amended to the end
- 'r+' – Special read and write mode, which is used to handle both actions when working with a file

**Opening and closing a file:**

- File\_object=open('file\_name', 'mode')
- File\_object.close()

## **1.4 Document processing**

A document is a form of information . A document can be put into an electronic form and stored in a computer as one or more file s. Often a single document becomes a single file. An entire document or individual parts may be treated as individual data items.

**IMPROVEMENTS:**

Improvement in document processing should, in the rational world of economics, be determined by balancing:

- 1) The cost of operating the system utilizing the new technology versus the old.
- 2) Cost of transition from one to another.
- 3) The advantages derivable from the new technology.

Here we are using the document processing in such a way that we are searching for specific words in a certain electronic document. We can modify and replace the words we have found. We can also perform indexing to know the index values of the keywords.



# ***CHAPTER 2***

## ***DESIGN***

## 2.1 Domain understanding

The main object of this project is to perform operations like searching, modifying and indexing all the keywords of the file in the separate file.

- **Searching:** In this operation, it prints the entire line of the given word to be searched followed by line number.
- **Modifying:** If any word needs to be replaced, then the user can modify the particular word using line number.
- **Indexing:** It prints all the keywords index value in a separate file present in the respective original file.
- **Annotation:** It retrieves the meaning for the given word.

## 2.2 Requirements

### 2.21 User Requirements

- Operating system: Windows
- Python3 installed

### 2.22 Software and hardware requirements

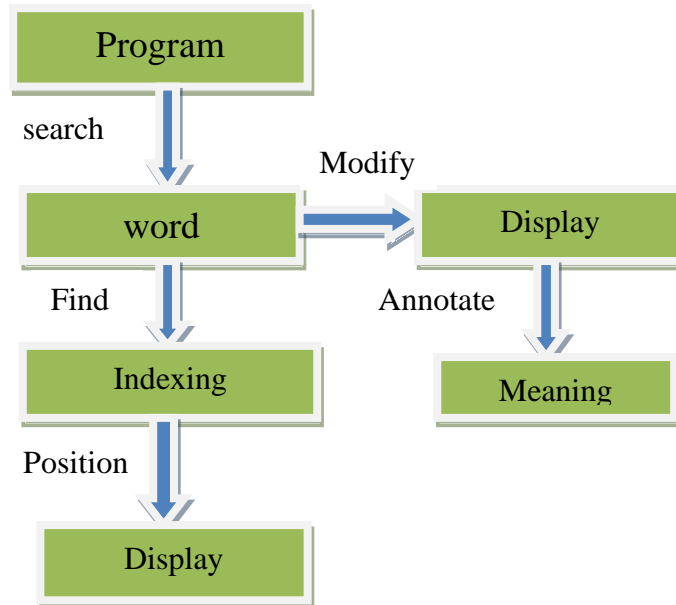
- Programming language: python

## 2.3 System Analysis

When the program is executed, it primarily asks the user to operate the operation presented that is to find.

When the user chooses the option to find, then it asks for the user to input the file in which he wants to “find” the word in. After the file is obtained, the entire file is indexed and the index values are maintained in a separate file.

## 2.4 Block diagram of the operations performed in the project.



# ***CHAPTER 3***

## ***IMPLEMENTATION***

## 3.1 Searching

### Algorithm:

1. Open file in read mode.
2. Get the input from user which is to be searched.
3. An empty string is declared and count=1 is initialized
4. Read the lines of a file and split it using split() function  
    S= f1.readline()  
    l= s.split()
5. By using while loop ,check the given word is present in s or not  
    while(s):  
        s=f1.readline()  
        l=s.split()  
        if word in l:  
            print("line number:",count,":",s)  
            count+=1

```
13
14 start2=time.time()
15 f1=open(r"C:\Users\LAVANYA\Desktop\f5.txt",encoding="utf8")
16 word=input("enter the word to be searched:")
17 s=""
18 count=1
19 s=f1.readline()
20 l=s.split()
21 print(l)
22
23 while(s):
24     s=f1.readline()
25     l=s.split()
26     if word in l:
27         print("line number:",count,":",s)
28         count+=1
29 end2=time.time()
30 print("searching completed in:",end2-start2,'milliseconds')
31
```

### 3.1.1

## 3.2 Modification

### Algorithm:

- Step 1 : Get the word which is to be modified from the user.
- Step 2: Get the new word to be replaced
- Step 3: Open the file in write mode and write the word to be replaced.
- Step 4: Display the entire line for which the word is modified.

### Code:

```

31
32 print("do you want to modify?y/n")
33 n=input()
34 if(n=='y'):
35
36     linenumber=int(input("enter the line number to be modified:"))
37     w=input("enter the word to be modified:")
38     w1=input("enter the new word:")
39     f=open(r"C:\Users\LAVANYA\Desktop\f5.txt", "rt")
40     data=f.read()
41     data=data.replace(w,w1)
42     f.close()
43     f=open(r"C:\Users\LAVANYA\Desktop\f5.txt", "wt")
44     f.write(data)
45     f.close()
46     print(linecache.getline(r"C:\Users\LAVANYA\Desktop\f5.txt",linenumber+1))
47     print("modified successfully!!!!")
48 else:
49     ("end of modification")

```

### 3.2.1

## 3.3 Indexing

### Algorithm:

Step 1:Open the text file which contains data in read mode and also create an index file in a write mode.

Step 2:Read all the lines in file and initialize count=0

Step 3:       for line in s.split(" "):  
                   stri = line[:]  
                   index.write(stri+" ")  
                   c = str(count)  
                   index.write(c + "\n")  
                   count = len(line) + count

Step 4:The index value of the words will be stored in the index file.

### Code:

```

50 print("do you want to see indexing?y/n")
51 q=input()
52 start1=time.time()
53 if(q=='y'):
54     l=input("enter the word:")
55     with open(r"C:\Users\LAVANYA\Desktop\f5.txt",encoding="utf8") as pack:
56         with open(r"C:\Users\LAVANYA\Desktop\inde.txt", "w") as index:
57             count=0
58             f=pack.read()
59             s=f.replace("\n", " ")
60             for line in s.split(" "):
61                 stri = line[:]  

62                 index.write(stri+" ")
63                 c = str(count)
64                 index.write(c + "\n")
65                 count = len(line) + count
66 print("indexing completed")
67 end1=time.time()
68 print("indexing is done in:",end1-start1,"milliseconds")

```

### 3.3.1

### 3.4 Annotation:

#### Algorithm:

Step 1: Import pydictionary

Step 2: Get the word for which meaning to be extracted.

Step 3: Through built in function extract the meaning with the internet connection.

#### Code:

```
69 print("do you want to annotate?y/n")
70 z=input()
71 if(z=='y'):
72     ab=input("enter the word :")
73     print(dictionary.meaning(ab))
74 else:
75     print("end of annotation")
76
77
```

Python console

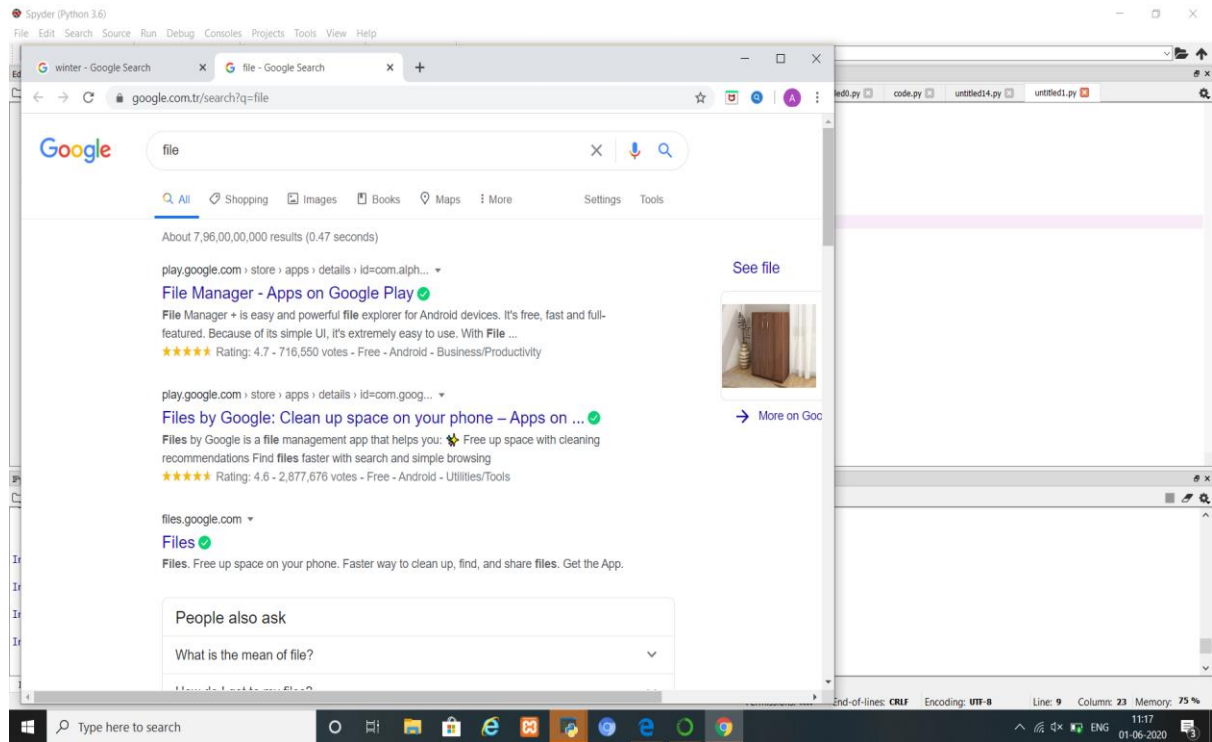
#### 3.4.1

Alternate way for annotation:

```
6
7
8 import webbrowser
9 search_terms = ['file']
10 for term in search_terms:
11     url = "https://www.google.com.tr/search?q={}".format(term)
12     webbrowser.open_new_tab(url)
```

#### 3.5.1

By importing web browser, we can directly get connected to the google. Hence, apart from getting the meaning for a given word, we can read information by selecting any website displayed in the page. The output is also shown below.



### 3.6.1

Here, we also tried to implement an alternate way. So, not only the meaning, the entire information can be read which helps the user to get more information.

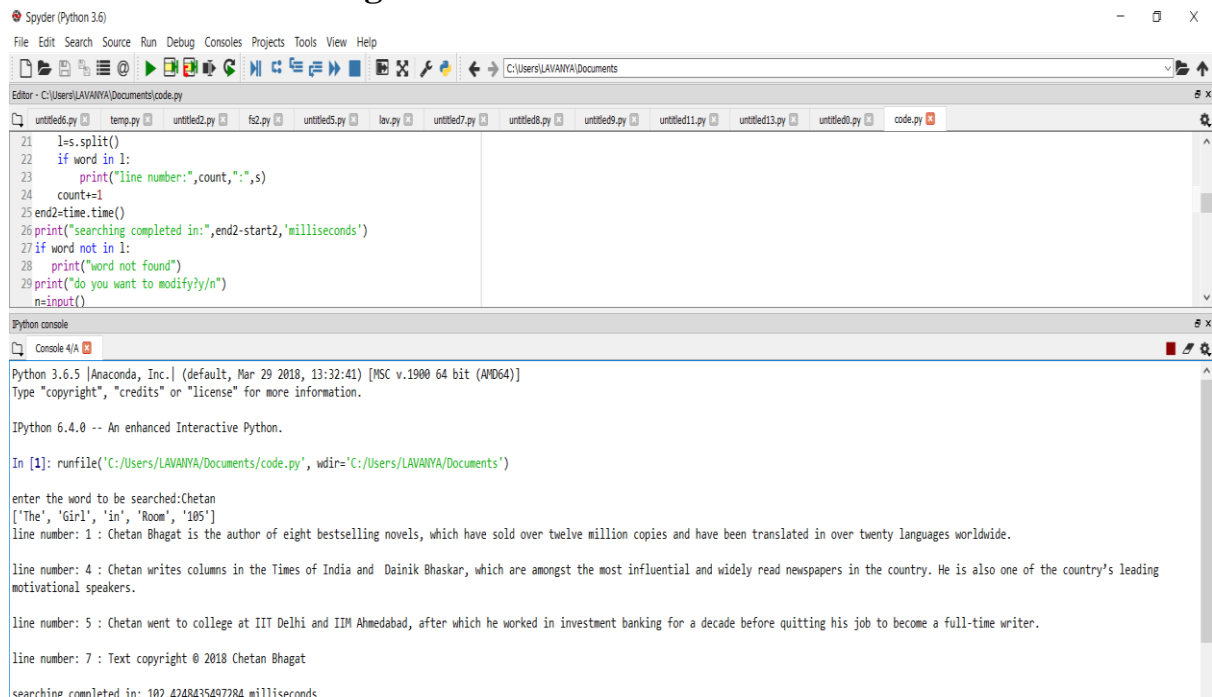


# ***CHAPTER 4***

## ***RESULTS AND SNAPSHOTS***

# SNAPSHOTS

## 4.1 Result for searching:



```
21 l=s.split()
22 if word in l:
23     print("line number:",count,"-",s)
24     count+=1
25 end2=time.time()
26 print("searching completed in:",end2-start2,'milliseconds')
27 if word not in l:
28     print("word not found")
29 print("do you want to modify?y/n")
n=input()
```

Python console

Console 4/A

Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)]  
Type "copyright", "credits" or "license()" for more information.

IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/LAVANYA/Documents/code.py', wdir='C:/Users/LAVANYA/Documents')

enter the word to be searched:Chetan  
['The', 'Girl', 'in', 'Room', '105']  
line number: 1 : Chetan Bhagat is the author of eight bestselling novels, which have sold over twelve million copies and have been translated in over twenty languages worldwide.

line number: 4 : Chetan writes columns in the Times of India and Dainik Bhaskar, which are amongst the most influential and widely read newspapers in the country. He is also one of the country's leading motivational speakers.

line number: 5 : Chetan went to college at IIT Delhi and IIM Ahmedabad, after which he worked in investment banking for a decade before quitting his job to become a full-time writer.

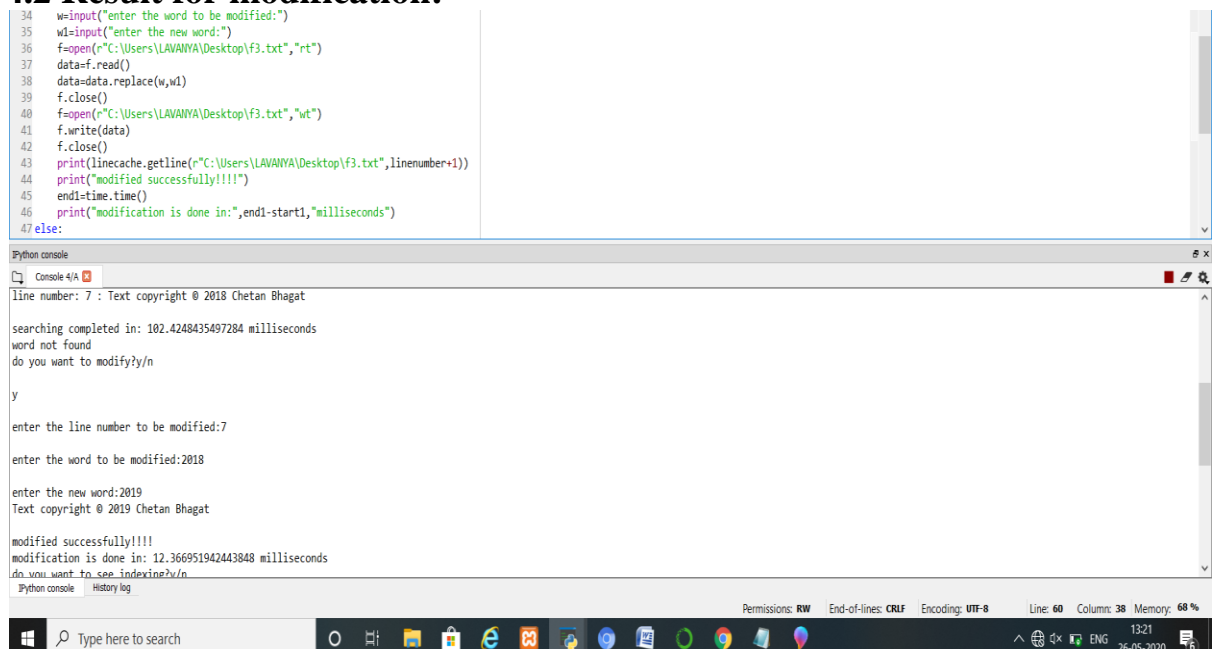
line number: 7 : Text copyright © 2018 Chetan Bhagat

searching completed in: 102.4248435497284 milliseconds

### 4.1.1

Here, we can observe that if any word is searched it prints entire line containing that particular word with the line number.

## 4.2 Result for modification:



```
34 w=input("enter the word to be modified:")
35 w1=input("enter the new word:")
36 f=open(r"C:\Users\LAVANYA\Desktop\3.txt","rt")
37 data=f.read()
38 data=data.replace(w,w1)
39 f.close()
40 f=open(r"C:\Users\LAVANYA\Desktop\3.txt","wt")
41 f.write(data)
42 f.close()
43 print(linecache.getline(r"C:\Users\LAVANYA\Desktop\3.txt",linenumber+1))
44 print("modified successfully!!!!")
45 end1=time.time()
46 print("modification is done in:",end1-start1,"milliseconds")
47 else:
```

Python console

Console 4/A

line number: 7 : Text copyright © 2018 Chetan Bhagat

searching completed in: 102.4248435497284 milliseconds  
word not found  
do you want to modify?y/n  
y

enter the line number to be modified:7  
enter the word to be modified:2018  
enter the new word:2019  
Text copyright © 2019 Chetan Bhagat

modified successfully!!!!  
modification is done in: 12.366951942443848 milliseconds  
do you want to see indexine?y/n

Python console History log

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 60 Column: 38 Memory: 68 %

### 4.2.1

In the above figure, the modification is done. If any word needs to be replaced then it can be changed in this operation by mentioning line number.

### 4.3 Result for indexing:

```

index1 - Notepad
File Edit Format View Help
speakers, 655
Chetan 664
went 670
to 674
college 676
at 683
IIT 685
Delhi 688
and 693
IIM 696
Ahmedabad, 699
after 709
which 714
he 719
worked 721
in 727
investment 729
banking 739
for 746
a 749
decade 750
before 756
quitting 762
his 770
job 773
to 776
become 778
a 784
full-time 785
writer. 794
This 801
is 805
a 807
work 808
of 812
fiction. 814
Name, 822
characters, 828
organisations, 839
places, 853
events, 860
and 867

```

#### 4.3.1

In indexing, when a word is searched all the words present in the document are indexed and will be stored in a separate file.

### 4.4 Time analysis for searching and indexing:

```

Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/LAVANYA/Documents/code.py', wdir='C:/Users/LAVANYA/Documents')

enter the word to be searched:Chetan
['The', 'Girl', 'in', 'Room', '105']
line number: 1 : Chetan Bhagat is the author of eight bestselling novels, which have sold over twelve million copies and have been translated in over twenty languages worldwide.

line number: 4 : Chetan writes columns in the Times of India and Dainik Bhaskar, which are amongst the most influential and widely read newspapers in the country. He is also one of the country's leading motivational speakers.

line number: 5 : Chetan went to college at IIT Delhi and IIM Ahmedabad, after which he worked in investment banking for a decade before quitting his job to become a full-time writer.

line number: 7 : Text copyright © 2018 Chetan Bhagat
searching completed in: 102.4248435497284 milliseconds

```

#### 4.4.1

```

Python console
Console 4/4
line number: 7 : Text copyright © 2018 Chetan Bhagat

searching completed in: 102.4248435497284 milliseconds
word not found
do you want to modify?y/n

y

enter the line number to be modified:7
enter the word to be modified:2018
enter the new word:2019
Text copyright © 2019 Chetan Bhagat

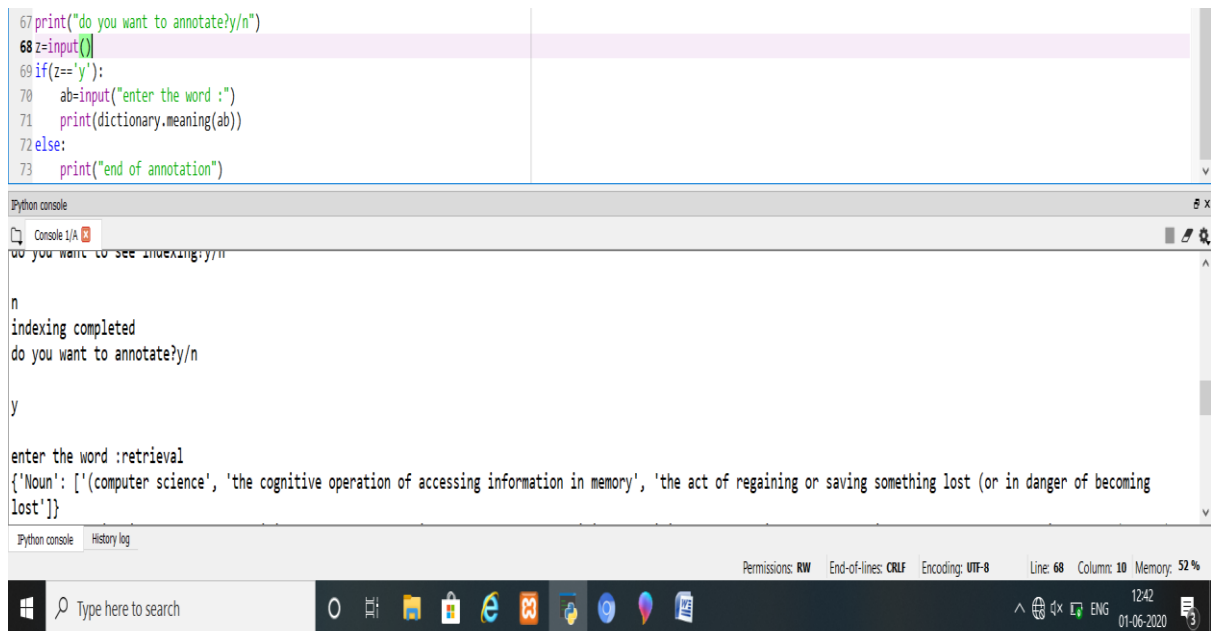
modification is done in: 12.366951942443848 milliseconds

```

#### 4.4.2

From the above figures 4.4.1 and 4.4.2, we can also analyse the time taken to perform searching and modify operations.

## 4.6 Result for annotation:



```
67 print("do you want to annotate?y/n")
68 z=input()
69 if(z=='y'):
70     ab=input("enter the word :")
71     print(dictionary.meaning(ab))
72 else:
73     print("end of annotation")
```

do you want to see indexing?y/n

n

indexing completed

do you want to annotate?y/n

y

enter the word :retrieval

{'Noun': ['(computer science', 'the cognitive operation of accessing information in memory', 'the act of regaining or saving something lost (or in danger of becoming lost)']}

Python console History log

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 68 Column: 10 Memory: 52 %

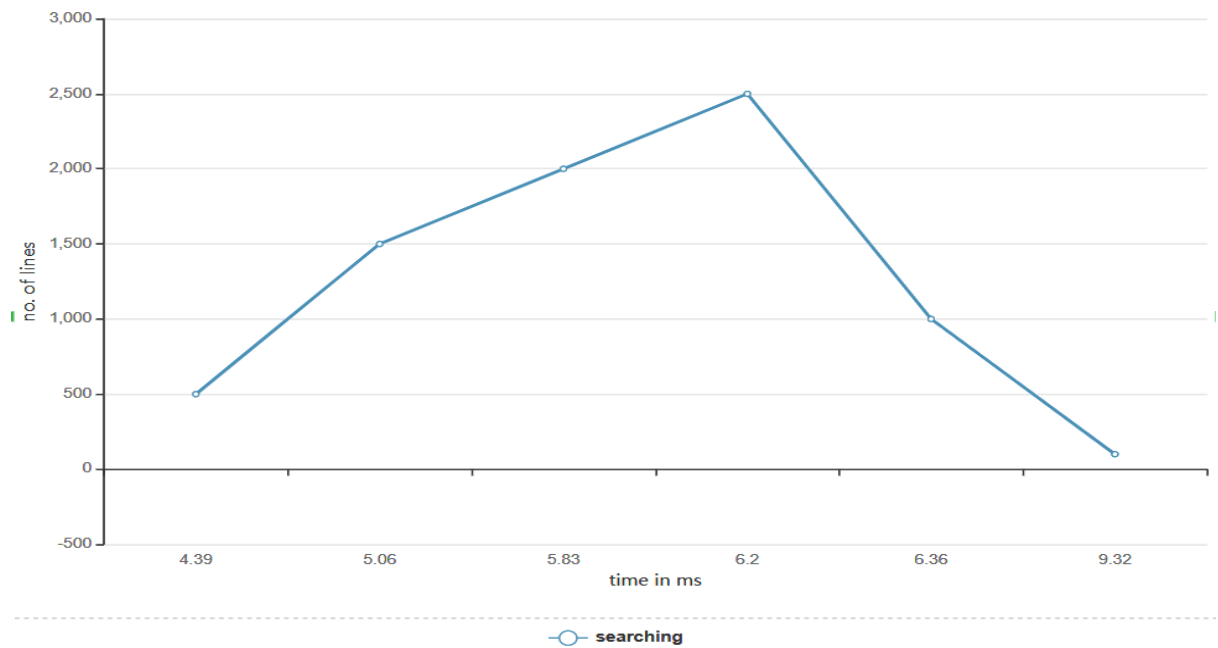
Type here to search

12:42 01-06-2020

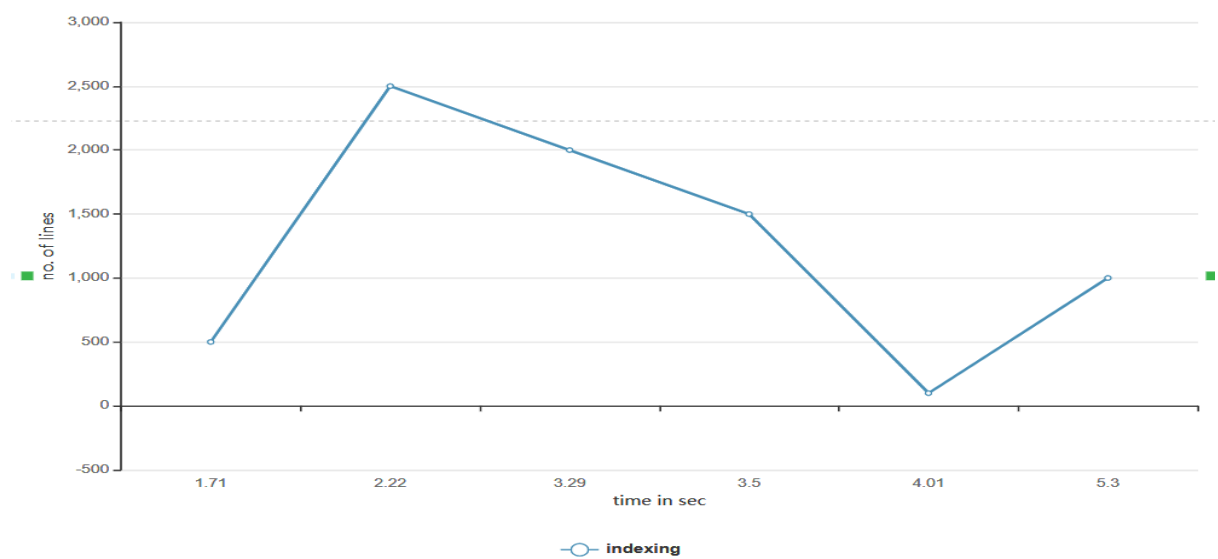
### 4.6.1

In the above figure, we are finding the meaning of the given word.

## Analysis:



4.7



4.8

The above 2 line graphs i.e., 4.7 and 4.8 indicates time analysis for searching and indexing respectively. X-axis represents time in sec and y-axis represents number of lines of data.

| No. Of lines | Time taken for searching(in ms) | Time taken for indexing(in ms) |
|--------------|---------------------------------|--------------------------------|
| 100          | 9.32                            | 4.01                           |
| 500          | 4.39                            | 1.71                           |
| 1000         | 6.36                            | 5.3                            |
| 1500         | 5.06                            | 3.5                            |
| 2000         | 5.83                            | 3.29                           |
| 2500         | 6.2                             | 2.2                            |

## **FUTURE WORK**

- We can add faster searching mechanism which reduces the time consumption.
- A separate file can be created for searching, so that oftenly searched words will be retrieved soon.
- Annotation can be performed to print the meaning of the given word.

## **CONCLUSION**

This project was progressive in nature, hence I learnt more about python and its functions. It helps me to understand about operations on a file. It increased my interest in coding. It was a very good experience in learning objectives. I hereby conclude that 'document processing' mini project has been completed with the best of my ability.

## **REFERENCES:**

1.File structures – An Object Oriented Approach with C++

Michael J.Folk , Bill Zoellitk , Greg Riccardi

2.File structures using C++

K.R. Venugopal, K. G. Srinivas , P.M. Krishnaraj

3. <https://stackoverflow.com/questions/4940032/how-to-search-for-a-string-in-text-file>