FILE_PROGRAM_VHA_2025

November 29, 2025

1 3.Write a "pager" program. Your solution should prompt for a filename, and display the text file 25 lines at a time, pausing each time to ask the user to "press a key to continue."

[ ]:
```
def pagetext(text_lined, num_lines=25):
   for index,line in enumerate(text_lined):
       if index % num_lines == 0 and index:
           X=input("Hit any key to continue press q to quit")
           if X.lower() != 'q':
                print(line)
           if X.lower() == 'q':
                break

       else:
               print (line)
text_lined=open(input("enter file name: "))
pagetext(text_lined,25)
```

```
enter file name: bhai.txt
.

File Filtering. Display all lines of a file, except those that start with a
pound sign ( # ), the comment character for Python, Perl, Tcl, and most other
scripting languages.



Extra credit: Also strip out comments that begin after the first character.



9-2.

File Access. Prompt for a number N and file F, and display the first N lines of
F.
```

Vishal Acharya

9-3.

File Information. Prompt for a filename and display the number of lines in that text file.

9-4.

File Access. Write a "pager" program. Your solution should prompt for a filename, and display the text file 25 lines at a time, pausing each time to ask the user to "press a key to continue."

9-5.

Test Scores. Update your solution to the test scores problems (Exercises 5-3 and 6-4) by allowing a set of test scores be loaded from a file. We leave the file format to your discretion.

9-6.

File Comparison. Write a program to compare two text files. If they are different, give the line and column numbers in the files where the first difference occurs.

9-7.

Parsing Files. Win32 users: Create a program that parses a Windows .ini file. POSIX users: Create a program that parses the /etc/services file. All other platforms: Create a program that parses a system file with some kind of structure to it.

9-8.

Module Introspection. Extract module attribute information. Prompt the user for a module name (or accept it from the command line). Then, using dir() and other built-in functions, extract all its attributes, and display their names, types, and values.

Vishal Acharya

Hit any key to continue press q to quitd

9-9.

"PythonDoc." Go to the directory where your Python standard library modules are located. Examine each .py file and determine whether a __doc__ string is available for that module. If so, format it properly and catalog it. When your program has completed, it should present a nice list of those modules that have documentation strings and what they are. There should be a trailing list showing which modules do not have documentation strings (the shame list). Extra credit: Extract documentation for all classes and functions within the standard library modules.

9-10.

Home Finances. Create a home finance manager. Your solution should be able to manage savings, checking, money market, certificate of deposit (CD), and similar accounts. Provide a menu-based interface to each account as well as operations such as deposits, withdrawals, debits, and credits. An option should be given to a user to remove transactions as well. The data should be stored to file when the user quits the application (but randomly during execution for backup purposes).

9-11.

Web site Addresses.

Write a URL bookmark manager. Create a text-driven menu-based application that allows the user to add, update, or delete entries. Entries include a site name, Web site URL address, and perhaps a one-line description (optional). Allow search functionality so that a search "word" looks through both names and URLs for possible matches. Store the data to a disk file when the user quits the application, and load up the data when the user restarts.

(b) Upgrade your solution to part (a) by providing output of the bookmarks to a legible and syntactically correct HTML file (.htm or .html) so that users can then point their browsers to this output file and be presented with a list of their bookmarks. Another feature to implement is allowing the creation of "folders" to allow grouping of related bookmarks. Extra credit: Read the

Vishal Acharya

literature on regular expressions and the Python re module. Add regular expression validation of URLs that users enter into their databases.

9-12.

Users and Passwords.

Do Exercise 7-5, which keeps track of usernames and passwords. Update your code to support a "last login time" (7-5a). See the documentation for the time module to obtain timestamps for when users "log in" to the system.

Also, create the concept of an "administrative" user that can dump a list of all the users, their passwords (you can add encryption on top of the passwords if you wish [7-5c]), and their last login times (7-5b).

The data should be stored to disk, one line at a time, with fields delimited by colons (:), e.g., "joe:boohoo:953176591.145", for each user. The number of lines in the file will be the number of users that are part of your system.

Further update your example such that instead of writing out one line at a time, you pickle the entire data object and write that out instead. Read the documentation on the pickle module to find out how to flatten or serialize your object, as well as how to perform I/O using picked objects. With the addition of this new code, your solution should take up fewer lines than your solution in part (a).

Vishal Acharya

2 **4.Text Processing. You are tired of seeing lines on your e-mail wrap because people type lines that are too long for your mail reader application. Create a program to scan a text file for all lines longer than 80 characters. For each of the offending lines, find the closest word before 80 characters and break the line there, inserting the remaining text to the next line (and pushing the previous next line down one). When you are done, there should be no lines longer than 80 characters.**

[6]:
```
f=open('mbox-short.txt','r',encoding="utf-8")

lis=[]
def ding(a):

    if len(a)<=80:
        lis.append(a)
        return
    else:
        if a[79]==' ':

            lis.append(a[:80])
            ding(a[80:])

        elif a[79]!=' ':
            ind=a.rfind(' ',0,79)

            lis.append(a[:ind+1])
            ding(a[ind+1:])

for x in f:
    if len(x)>80:
        ding(x)
    else:
        lis.append(x)

ty=open('v123.txt','w',encoding="utf-8")
for x in lis:
    if x[-1]==' ':
        x=x[:-1]+'\n'
    else :
        x+='\n'
    ty.write(x)
f.close()
ty.close()
f=open("v123.txt")
```

Vishal Acharya

```
print(f.read())
```

```
Math Essay: Mathematics is generally defined as the science that deals with
numbers. It involves operations among numbers,

and it also helps you to calculate the product price, how many discounted
prizes here, and If you

good in maths so you can calculate very fast. Mathematicians and scientists
rely on mathematics principles in their real-life to experiments

with new things every day. Many students say that  I hate mathematics  and
maths is a useless subject,

but it is wrong because without mathematics your life is tough to survive. Math
has its applications in every field.

You can also find more Essay Writing articles on events, persons, sports,
technology and many more.

Long and Short Essays on Math for Students and Kids in English

We are presenting students with essay samples on an extended essay of 500 words
and a

short of 150 words on the topic of math for reference.

Long Essay on Math 500 Words in English
```

3 16.Write a python program to find the longest words in a read file.

Friends are crazy, Friends are naughty !

Friends are honest, Friends are best !

Friends are like keygen, friends are like license key !

We are nothing without friends, Life is not possible without friends !

[5]:
```
d={}
f=open("friends.txt")
for lines in f:
    for word in lines.split():
        if word not in d:
            d[word]=len(word)
```

Vishal Acharya

```
a=max(d.values())
for i,j in d.items():
    if j==a:
        print(i,j)
```

```
friends, 8
possible 8
```

4 20.Write a Python program to count the frequency of words in a file

[2]:
```
from collections import Counter
def word_count(fname):
        with open(fname) as f:
                return Counter(f.read().split())

print("Number of words in the file :",word_count("example.txt"))
```

```
Number of words in the file : Counter({'and': 6, 'is': 3, 'Python': 3, 'a': 3,
'programming': 3, 'dynamic': 2, 'code': 2, 'to': 2, 'in': 2, 'or': 2, 'what': 1,
'language?': 1, 'widely': 1, 'used': 1, 'high-level,': 1, 'general-purpose,': 1,
'interpreted,': 1, 'language.': 1, 'Its': 1, 'design': 1, 'philosophy': 1,
'emphasizes': 1, 'readability,': 1, 'its': 1, 'syntax': 1, 'allows': 1,
'programmers': 1, 'express': 1, 'concepts': 1, 'fewer': 1, 'lines': 1, 'of': 1,
'than': 1, 'possible': 1, 'languages': 1, 'such': 1, 'as': 1, 'C++': 1, 'Java.':
1, 'supports': 1, 'multiple': 1, 'paradigms,': 1, 'including': 1, 'object-
oriented,': 1, 'imperative': 1, 'functional': 1, 'procedural': 1, 'styles.It':
1, 'features': 1, 'type': 1, 'system': 1, 'automatic': 1, 'memory': 1,
'management': 1, 'has': 1, 'large': 1, 'comprehensive': 1, 'standard': 1,
'library.': 1, 'The': 1, 'best': 1, 'way': 1, 'we': 1, 'learn': 1, 'anything':
1, 'by': 1, 'practice': 1, 'exercise': 1, 'questions.': 1, 'We': 1, 'have': 1,
'started': 1, 'this': 1, 'section': 1, 'for': 1, 'those': 1, '(beginner': 1,
'intermediate)': 1, 'who': 1, 'are': 1, 'familiar': 1, 'with': 1, 'Python.': 1})
```

[5]:
```
f=open("example.txt")
d={}
for line in f:
    for words in line.split():
        d[words]=d.get(words,0)+1
print(d)
```

```
{'what': 1, 'is': 3, 'Python': 3, 'language?': 1, 'a': 3, 'widely': 1, 'used':
1, 'high-level,': 1, 'general-purpose,': 1, 'interpreted,': 1, 'dynamic': 2,
'programming': 3, 'language.': 1, 'Its': 1, 'design': 1, 'philosophy': 1,
'emphasizes': 1, 'code': 2, 'readability,': 1, 'and': 6, 'its': 1, 'syntax': 1,
'allows': 1, 'programmers': 1, 'to': 2, 'express': 1, 'concepts': 1, 'in': 2,
'fewer': 1, 'lines': 1, 'of': 1, 'than': 1, 'possible': 1, 'languages': 1,
'such': 1, 'as': 1, 'C++': 1, 'or': 2, 'Java.': 1, 'supports': 1, 'multiple': 1,
```

Vishal Acharya

```
'paradigms,': 1, 'including': 1, 'object-oriented,': 1, 'imperative': 1,
'functional': 1, 'procedural': 1, 'styles.It': 1, 'features': 1, 'type': 1,
'system': 1, 'automatic': 1, 'memory': 1, 'management': 1, 'has': 1, 'large': 1,
'comprehensive': 1, 'standard': 1, 'library.': 1, 'The': 1, 'best': 1, 'way': 1,
'we': 1, 'learn': 1, 'anything': 1, 'by': 1, 'practice': 1, 'exercise': 1,
'questions.': 1, 'We': 1, 'have': 1, 'started': 1, 'this': 1, 'section': 1,
'for': 1, 'those': 1, '(beginner': 1, 'intermediate)': 1, 'who': 1, 'are': 1,
'familiar': 1, 'with': 1, 'Python.': 1}
```

5 21.write file from list

[6]:
```
color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
with open('abc.txt', "w") as myfile:
        for c in color:
                myfile.write(c+"\n")

content = open('abc.txt')
print(content.read())
```

```
Red
Green
White
Black
Pink
Yellow
```

6 22.Write a Python program to combine each line from first file with the corresponding line in second file

[7]:
```
with open('example.txt') as fh1, open('friends.txt') as fh2:
    for line1, line2 in zip(fh1, fh2):
        print(line1+line2)
```

```
what is Python language?
Friends are crazy, Friends are naughty !

Python is a widely used high-level, general-purpose, interpreted, dynamic
programming language. Its design philosophy emphasizes code readability, and its
syntax allows programmers to express concepts in fewer lines of code than
possible in
Friends  6 are honest, Friends are  best !

languages such as C++ or Java.
Friends are like keygen, friends are like license key !

Python supports multiple programming paradigms, including object-oriented,
```

Vishal Acharya

imperative and functional programming or procedural styles.It features a dynamic type system and automatic memory management and has a large and comprehensive standard library. The best way we learn anything is by practice and exercise questions. We have started this section for those (beginner to intermediate) who are familiar with Python.new We are nothing without friends, Life is not possible without friends !

7 23.write a Python program that takes a text file as input and returns the number of words of a given text file.

Note: Some words can be separated by a comma with no space.

[9]:
```
def count_words(filepath):
    with open(filepath) as f:
        data = f.read()
        data.replace(",", " ")
        return len(data.split(" "))
print(count_words("example.txt"))
```

```
189
```

8 24.Write a function in python to count the number of lines from a text file “example2.txt” which is not starting with an alphabet “T”.

A boy is playing there.

There is a playground.

An aeroplane is in the sky.

The sky is pink.

Alphabets and numbers are allowed in the password.

[12]:
```
def line_count():
    file = open("example2.txt","r")
    count=0
    for line in file:
        if line[0] not in 'T':
            count+= 1
    file.close()
    print("No of lines not starting with 'T'=",count)

line_count()
```

```
No of lines not starting with 'T'= 3
```

Vishal Acharya

9 25. Write a function in Python to read lines from a text file "example3.txt". Your function should find and display the occurrence of the word "the".

India is the fastest-growing economy. India is looking for more investments around the globe. The whole world is looking at India as a great market. Most of the Indians can foresee the heights that India is capable of reaching.

[13]:
```
def count_words():
    file = open("example3.txt","r")
    count = 0
    data = file.read()
    words = data.split()
    for word in words:
        if word =="the" or word =="The":
            count += 1
    print(count)
    file.close()

count_words()
```

```
5
```

10 26.Write a function display_words() in python to read lines from a text file "example2.txt", and display those words, which are less than 4 characters.

[15]:
```
def display_words():
    file = open("example2.txt","r")
    data = file.read()
    words = data.split()
    for word in words:
        if len(word) < 4:
            print(word, end=" ")
    file.close()

display_words()
```

```
A boy is is a An is in the The sky is and are in the
```

Vishal Acharya

11 27.Write a function in Python to count words in a text file those are ending with alphabet “e”.

[18]:
```
def count_words():
    file = open("example2.txt","r")
    count = 0
    data = file.read()
    words = data.split()
    for word in words:
        if word[-1] == 'e':
            print(word)
            count+=1
    print(count)
    file.close()

count_words()
```

```
There
aeroplane
the
The
are
the
6
```

12 28.student has used a text editing software to type some text. After saving the article as example4.TXT, she realised that she has wrongly typed alphabet J in place of alphabet I everywhere in the article.

Write a function definition for JTOI() in Python that would display the corrected version of entire content of the file WORDS.TXT with all the alphabets “J” to be displayed as an alphabet “I” on screen.

Note: Assuming that example4.TXT does not contain any J alphabet otherwise.

input WELL, THJS JS A WORD BY JTSELF. YOU COULD STRETCH THJS TO BE A SENTENCE

The function JTOI() should display the following content:

WELL, THIS IS A WORD BY ITSELF. YOU COULD STRETCH THIS TO BE A SENTENCE

[20]:
```
def JTOI():
    file = open("example4.txt","r")
    data = file.read()
    for letter in data:
        if letter == 'J':
```

Vishal Acharya

```
            print("I",end="")
        else:
            print(letter,end="")

    file.close()

JTOI()
```

```
WELL, THIS IS A WORD BY ITSELF. YOU COULD STRETCH THIS TO BE A SENTENCE
```

13 29.Read text file into a variable and replace all newlines with space

[21]:
```
with open('example2.txt', 'r') as file:
    data = file.read().replace('\n', ' ')
    print(data)
```

```
A boy is playing there. There is a playground. An aeroplane is in the sky. The
sky is pink. Alphabets and numbers are allowed in the password.
```

14 30.Take input from user and store in .txt file in Python

[23]:
```
temp = input("Please enter your information!!   ")
try:
    with open('gfg.txt', 'w') as gfg:
        gfg.write(temp)
except Exception as e:
    print("There is a Problem", str(e))
f=open("gfg.txt")
print(f.read())
```

```
Please enter your information!!   mnjh
mnjh
```

15 31. Change case of all characters in .txt file using Python

[26]:
```
with open('example2.txt', 'r') as data_file:
    print(data_file.read())
    with open('output.txt', 'a') as output_file:
         output_file.write(data_file.read().swapcase())
f=open("output.txt")
print(f.read())
```

```
A boy is playing there.
There is a playground.
```

Vishal Acharya

```
An aeroplane is in the sky.
The sky is pink.
Alphabets and numbers are allowed in the password.
a BOY IS PLAYING THERE.
tHERE IS A PLAYGROUND.
aN AEROPLANE IS IN THE SKY.
tHE SKY IS PINK.
aLPHABETS AND NUMBERS ARE ALLOWED IN THE PASSWORD.
```

16 32.Python Program to Replace Text in a File

Replacing Text could be either erasing the entire content of the file and replacing it with new text or it could mean modifying only specific words or sentences within the existing text.

[3]:
```
s = input("Enter text to replace the existing contents:")
f = open("example2.txt", "r+")
f.truncate(0)
f.write(s)
f.close()
print("Text successfully replaced")
f=open("example2.txt")
print(f.read())
```

```
Enter text to replace the existing contents:a BOY IS PLAYING THERE. tHERE IS A
PLAYGROUND. aN AEROPLANE IS IN THE SKY. tHE SKY IS PINK. aLPHABETS AND NUMBERS
ARE ALLOWED IN THE PASSWORD
Text successfully replaced
a BOY IS PLAYING THERE. tHERE IS A PLAYGROUND. aN AEROPLANE IS IN THE SKY. tHE
SKY IS PINK. aLPHABETS AND NUMBERS ARE ALLOWED IN THE PASSWORD
```

[5]:
```
# Python program to replace text in a file
x = input("enter text to be replaced:")
y = input("enter text that will replace:")
f = open("example2.txt", "r")
l = f.readlines()
k=[]
c = 0
for i in l:
    if x in i:
        Replacement = i.replace(x, y)
        k.append(Replacement)
    else:
        k.append(i)
    c += 1
f.close()
f=open("example2.txt","w")
f.writelines(k)
f.close()
```

Vishal Acharya

```
print("Text successfully replaced")
f=open("example2.txt","r")
print(f.read())
```

```
enter text to be replaced:is
enter text that will replace:i#s
Text successfully replaced
A boy i#s playing there.
There i#s a playground.
An aeroplane i#s in the sky.
The sky i#s pink.
```

17 33.Python Program to Delete Specific Line from File

1

2

3

4

5

6

7

8

9

10

11

12

[6]:
```
line_delete=int(input("enter line number which you want to delete"))
with open('number.txt', 'r') as fr:

        lines = fr.readlines()
        ptr = 1
with open('number.txt', 'w') as fw:
    for line in lines:

        if ptr != line_delete:
            fw.write(line)
        ptr += 1
print("Deleted")
f=open("number.txt")
print(f.read())
```

Vishal Acharya

```
enter line number which you want to delete5
Deleted
1
2
3
4
6
7
8
9
10
11
12
```

18 34.Python Program to Print Lines Containing Given String in File

Friends are crazy, Friends are naughty !

Friends are honest, Friends are best !

Friends are like keygen, friends are like license key !

We are nothing without friends, Life is not possible without friends !

[7]:
```
file_name = input("Enter The File's Name: ")
try:
    file_read = open(file_name, "r")
    text = input("Enter the String: ")
    lines = file_read.readlines()

    new_list = []
    idx = 0
    for line in lines:
        if text in line:
            new_list.insert(idx, line)
            idx += 1

    file_read.close()


    if len(new_list)==0:
        print("\n\"" +text+ "\" is not found in \"" +file_name+ "\"!")
    else:

        lineLen = len(new_list)
        print("\n**** Lines containing \"" +text+ "\" ****\n")
        for i in range(lineLen):
```

Vishal Acharya

```
            print(end=new_list[i])
        print()
except :
  print("\nThe file doesn't exist!")
```

```
Enter The File's Name: dost.txt
Enter the String: friends

**** Lines containing "friends" ****

Friends are like keygen, friends are like license key !
We are nothing without friends, Life is not possible without friends !
```

19 35.How to remove lines starting with any prefix using Python?

Friends are crazy, Friends are naughty !

Friends are honest, Friends are best !

Friends are like keygen, friends are like license key !

We are nothing without friends, Life is not possible without friends !

[8]:
```
file1 = open('dost.txt','r')
file2 = open('dost1.txt','w')
for line in file1.readlines():
    if not (line.startswith('We')):
        print(line)
        file2.write(line)
file2.close()
file1.close()
print("*"*50)
file=open("dost1.txt")
print(file.read())
```

```
Friends are crazy, Friends are naughty !

Friends are honest, Friends are  best !

Friends are like keygen, friends are like license key !

**************************************************
Friends are crazy, Friends are naughty !
Friends are honest, Friends are  best !
Friends are like keygen, friends are like license key !
```

Vishal Acharya

20 36.Eliminating repeated lines from a file using Python

Friends are crazy, Friends are naughty !

Friends are honest, Friends are best !

Friends are like keygen, friends are like license key !

Friends are honest, Friends are best !

We are nothing without friends, Life is not possible without friends !

[9]:
```
outputFile = open('pre.txt', "w")
inputFile = open('dost.txt', "r")
lines_seen_so_far = set()
for line in inputFile:
    if line not in lines_seen_so_far:
        outputFile.write(line)
        lines_seen_so_far.add(line)
inputFile.close()
outputFile.close()
f=open("pre.txt")
print(f.read())
```

```
Friends are crazy, Friends are naughty !
Friends are honest, Friends are  best !
Friends are like keygen, friends are like license key !
We are nothing without friends, Life is not possible without friends !
```

[10]:
```
file = open('dost.txt')
read = file.read()
file.seek(0)
line = 1
for word in read:
    if word == '\n':
        line += 1
print("Number of lines in file is: ", line)
array = []
for i in range(line):
    array.append(file.readline())
print(array)
```

```
Number of lines in file is:  5
['Friends are crazy, Friends are naughty !\n', 'Friends are honest, Friends are
best !\n', 'Friends are like keygen, friends are like license key !\n', 'Friends
are honest, Friends are  best !\n', 'We are nothing without friends, Life is not
possible without friends !']
```

21 37.Python – Append content of one text file to another

Friends are crazy, Friends are naughty !

Vishal Acharya

Friends are honest, Friends are best !

Friends are like keygen, friends are like license key !

Friends are honest, Friends are best !

We are nothing without friends, Life is not possible without friends !

[11]:
```
firstfile = input("Enter the name of first file ")
secondfile = input("Enter the name of second file ")
f1 = open(firstfile, 'r')
f2 = open(secondfile, 'r')
print('content of first file before appending -', f1.read())
print('content of second file before appending -', f2.read())
f2.close()
f1 = open(firstfile, 'a+')
f2 = open(secondfile, 'r')
f1.write(f2.read())
f1.seek(0)
f2.seek(0)
print('content of first file after appending -', f1.read())
print('content of second file after appending -', f2.read())
f1.close()
f2.close()
```

```
Enter the name of first file dost.txt
Enter the name of second file number.txt
content of first file before appending - Friends are crazy, Friends are naughty
!
Friends are honest, Friends are  best !
Friends are like keygen, friends are like license key !
Friends are honest, Friends are  best !
We are nothing without friends, Life is not possible without friends !
content of second file before appending - 1
2
3
4
5
6
7
8
9
10
11
12
content of first file after appending - Friends are crazy, Friends are naughty !
Friends are honest, Friends are  best !
Friends are like keygen, friends are like license key !
Friends are honest, Friends are  best !
We are nothing without friends, Life is not possible without friends !1
```

Vishal Acharya

```
2
3
4
5
6
7
8
9
10
11
12
content of second file after appending - 1
2
3
4
5
6
7
8
9
10
11
12
```

22 38.Python Program to Reverse the Content of a File using Stack

[14]:
```
class Stack:

    def __init__(self):
        self._arr = []
    def push(self, val):
        self._arr.append(val)

    def is_empty(self):
        return len(self._arr) == 0
    def pop(self):

        if self.is_empty():
            print("Stack is empty")
            return

        return self._arr.pop()
def reverse_file(filename):

    S = Stack()
```

Vishal Acharya

```
    original = open(filename)

    for line in original:
        S.push(line.rstrip("\n"))

    original.close()


    output = open(filename, 'w')

    while not S.is_empty():
        output.write(S.pop()+"\n")

    output.close()
filename = "number.txt"

# Calling the reverse_file function
reverse_file(filename)

# Now reading the content of the file
with open(filename) as file:
        for f in file.readlines():
            print(f, end ="")
```

```
12
11
10
9
8
7
6
5
4
3
2
1
```

23 39.Python program to Reverse a single line of a text file.line number given by user

[30]:
```
f = open('dost.txt', 'r')
lines = f.readlines()
f.close()
choice = int(input("enter choice: "))
line = lines[choice].split()
Reversed = " ".join(line[::-1])
```

Vishal Acharya

```
lines.pop(choice)
Reversed=Reversed+"\n"
lines.insert(choice, Reversed)
u = open('dost.txt', 'w')
u.writelines(lines)
u.close()
k=open("dost.txt","r")
print(k.read())
```

```
enter choice: 0
! naughty are Friends crazy, are Friends
Friends are honest, Friends are  best !
Friends are like keygen, friends are like license key?
We are nothing without friends, Life is not possible without friends !
```

24 40Python program to reverse the content of a file and store it in another file

25 full reverse

[31]:
```
f1 = open("output1.txt", "w")
with open("dost.txt", "r") as myfile:
    data = myfile.read()
data_1 = data[::-1]
f1.write(data_1)
f1.close()
f= open("output1.txt", "r")
print(f.read())
```

```
! sdneirf tuohtiw elbissop ton si efiL ,sdneirf tuohtiw gnihton era eW
?yek esnecil ekil era sdneirf ,negyek ekil era sdneirF
! tseb  era sdneirF ,tsenoh era sdneirF
! ythguan era sdneirF ,yzarc era sdneirF
```

[32]:
```
# Open the file in write mode
f2 = open("output2.txt", "w")
with open("dost.txt", "r") as myfile:
    data = myfile.readlines()
data_2 = data[::-1]
f2.writelines(data_2)
f2.close()
f= open("output2.txt", "r")
print(f.read())
```

```
We are nothing without friends, Life is not possible without friends !Friends
are like keygen, friends are like license key?
Friends are honest, Friends are  best !
```

Vishal Acharya

```
Friends are crazy, Friends are naughty !
```

26 41.Python program to Count the Number of occurrences of a key-value pair in a text file

[33]:
```
f = open("dost.txt", "r")
d = dict()

for res in f:

    res = res.strip()
    res = res.lower()
    lines = res.split()

    for line in lines:

        if line in d:
            d[line] = d[line]+1
        else:
            d[line] = 1

f.close()
for key in list(d.keys()):
    print("The count of {} is {}".format(key,d[key]))
```

```
The count of friends is 7
The count of are is 7
The count of crazy, is 1
The count of naughty is 1
The count of ! is 3
The count of honest, is 1
The count of best is 1
The count of like is 2
The count of keygen, is 1
The count of license is 1
The count of key? is 1
The count of we is 1
The count of nothing is 1
The count of without is 2
The count of friends, is 1
The count of life is 1
The count of is is 1
The count of not is 1
The count of possible is 1
```

Vishal Acharya

27 42.Python Program to merge two files into a third file

[34]:
```
data = data2 = ""
with open('dost.txt') as fp:
    data = fp.read()
with open('number.txt') as fp:
    data2 = fp.read()
data += "\n"
data += data2

with open ('file3.txt', 'w') as fp:
    fp.write(data)
f= open ('file3.txt', 'r')
print(f.read())
```

```
Friends are crazy, Friends are naughty !
Friends are honest, Friends are  best !
Friends are like keygen, friends are like license key?
We are nothing without friends, Life is not possible without friends !
12
11
10
9
8
7
6
5
4
3
2
1
```

[36]:
```
class String:
    def __init__(self):
        self.vowels=0
        self.spaces=0
        self.consonants=0
        self.uppercase=0
        self.lowercase=0
        self.string=str(input("Enter string: "))
    def count_uppercase(self):
        for letter in self.string:
            if letter.isupper():
                self.uppercase+=1
    def count_lowercase(self):
        for letter in self.string:
            if letter.islower():
```

Vishal Acharya

```
                self.lowercase+=1
    def count_vowels(self):
        for letter in self.string:
            if (letter in ("a","e","i","o","u","A","E","I","O","U")):
                self.vowels+=1
    def count_spaces(self):
        for letter in self.string:
            if letter==" ":
                self.spaces+=1
    def count_consonants(self):
        for letter in self.string:
            if (letter not in ("a","e","i","o","u","A","E","I","O","U") and␣
 ↪letter!=" "):
                self.consonants+=1
    def compute_stat(self):
        self.count_uppercase()
        self.count_lowercase()
        self.count_vowels()
        self.count_spaces()
        self.count_consonants()
    def display(self):
        print("vowels:",self.vowels)
        print("consonants:",self.consonants)
        print("spaces:",self.spaces)
        print("uppercase:",self.uppercase)
        print("lowercase:",self.lowercase)
s=String()
s.compute_stat()
s.display()
```

```
Enter string: lkj bgh  vgnb#%$%  bgSEDSF cc
vowels: 1
consonants: 22
spaces: 6
uppercase: 5
lowercase: 14
```

28 43.Write a program that prompts for a file name, then opens that file and reads through the file, looking for lines of the form:

X-DSPAM-Confidence: 0.8475

Count these lines and extract the floating point values from each of the lines and compute the average of those values and produce an output as shown below. Do not use the sum() function or a variable named sum in your solution. You can download the sample data at http://www.py4e.com/code3/mbox-short.txt when you are testing below enter mbox-short.txt as

Vishal Acharya

the file name.

[37]:
```
# Use the file name mbox-short.txt as the file name
fname = input("Enter file name: ")
fh = open(fname)
count=0
total=0
for line in fh:
    if not line.startswith("X-DSPAM-Confidence:"):
        continue
    t=line.find("0")
    number=float(line[t: ])
    total=total+number
    count=count+1
Average=total/count
print("Average spam confidence:",Average)
```

```
Enter file name: mbox-short.txt
Average spam confidence: 0.7507185185185187
```

29 44.Open the file romeo.txt and read it line by line. For each line, split the line into a list of words using the split() method. The program should build a list of words. For each word on each line check to see if the word is already in the list and if not append it to the list. When the program completes, sort and print the resulting words in python sort() order as shown in the desired output.

[38]:
```
fname = input("Enter file name: ")
fh = open(fname)
lst = list()
for line in fh:
    words=line.split()
    for i in words:
        if i in lst:
            continue
        lst.append(i)
ist=sorted(lst)
print(ist)
```

```
Enter file name: romeo.txt
['Arise', 'But', 'It', 'Juliet', 'Who', 'already', 'and', 'breaks', 'east',
'envious', 'fair', 'grief', 'is', 'kill', 'light', 'moon', 'pale', 'sick',
'soft', 'sun', 'the', 'through', 'what', 'window', 'with', 'yonder']
```

Vishal Acharya

30 45.Open the file mbox-short.txt and read it line by line. When you find a line that starts with 'From' like the following line:

From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008 You will parse the From line using split() and print out the second word in the line (i.e. the entire address of the person who sent the message). Then print out a count at the end. Hint: make sure not to include the lines that start with 'From:'. Also look at the last line of the sample output to see how to print the count.

[39]:
```
fname = "mbox-short.txt"
count=0
fh = open(fname)
for line in fh :
    line = line.rstrip()
    if not line.startswith('From '): continue
    count+=1
    words = line.split()
    print(words[1])


print("There were", count, "lines in the file with From as the first word")
```

```
stephen.marquard@uct.ac.za
louis@media.berkeley.edu
zqian@umich.edu
rjlowe@iupui.edu
zqian@umich.edu
rjlowe@iupui.edu
cwen@iupui.edu
cwen@iupui.edu
gsilver@umich.edu
gsilver@umich.edu
zqian@umich.edu
gsilver@umich.edu
wagnermr@iupui.edu
zqian@umich.edu
antranig@caret.cam.ac.uk
gopal.ramasammycook@gmail.com
david.horwitz@uct.ac.za
david.horwitz@uct.ac.za
david.horwitz@uct.ac.za
david.horwitz@uct.ac.za
stephen.marquard@uct.ac.za
louis@media.berkeley.edu
louis@media.berkeley.edu
ray@media.berkeley.edu
cwen@iupui.edu
cwen@iupui.edu
cwen@iupui.edu
```

Vishal Acharya

```
There were 27 lines in the file with From as the first word
```

31 **46. Write a program to read through the mbox-short.txt and figure out who has sent the greatest number of mail messages. The program looks for 'From' lines and takes the second word of those lines as the person who sent the mail. The program creates a Python dictionary that maps the sender's mail address to a count of the number of times they appear in the file. After the dictionary is produced, the program reads through the dictionary using a maximum loop to find the most prolific committer.**

[42]:
```
lst=list()
d=dict()
name = input("Enter file:")
if len(name) < 1:
    name = "mbox-short.txt"
handle = open(name)
for line in handle:
    line = line.rstrip()
    if not line.startswith('From '): continue
    words = line.split()
    lst.append(words[1])
for i in lst:
    d[i]=d.get(i,0)+1

b=max(d.values())
for k,v in d.items():
    if b==v:
        print(k,v)
```

```
Enter file:mbox-short.txt
cwen@iupui.edu 5
```

Vishal Acharya

32 47.Write a program to read through the mbox-short.txt and figure out the distribution by hour of the day for each of the messages. You can pull the hour out from the 'From' line by finding the time and then splitting the string a second time using a colon.From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008Once you have accumulated the counts for each hour, print out the counts, sorted by hour as shown below.

[41]:
```
lst=list()
gst=list()
d=dict()
name = input("Enter file:")
if len(name) < 1:
    name = "mbox-short.txt"
handle = open(name)
for line in handle:
    line = line.rstrip()
    if not line.startswith('From '): continue
    words = line.split()
    lst.append(words[5])

for i in lst:
    g=i.split(":")
    gst.append(g[0])
gst=sorted(gst)
for j in gst:
    d[j]=d.get(j,0)+1
for k,v in d.items():
    print(k,v)
```

```
Enter file:mbox-short.txt
04 3
06 1
07 1
09 2
10 3
11 6
14 1
15 2
16 4
17 2
18 1
19 1
```

Vishal Acharya

[9]:
```
f = open("last.txt", "r")
d = dict()
d={}
for s in f:
    for i in s.split():
        if i[0] not in d.keys():
            d[i[0]]=[]
            d[i[0]].append(i)
        else:
            d[i[0]].append(i)
f.close()
print(d)
```

```
{'a': ['are', 'aeroplanr', 'asd'], 'b': ['boy', 'banana', 'bghy', 'bghy'], 'v':
['vfg', 'vfg'], 'c': ['car', 'cat', 'cdf', 'cdf', 'caw'], 'r': ['rty', 'ret'],
'x': ['xsd'], 'z': ['zsd', 'zsx'], 'f': ['fgv'], 'd': ['dcf']}
{'a': ['are', 'aeroplanr', 'asd'], 'b': ['boy', 'banana', 'bghy', 'bghy'], 'v':
['vfg', 'vfg'], 'c': ['car', 'cat', 'cdf', 'cdf', 'caw'], 'r': ['rty', 'ret'],
'x': ['xsd'], 'z': ['zsd', 'zsx'], 'f': ['fgv'], 'd': ['dcf']}
```

[1]:
```
%%writefile word.txt
Cse student
I am student
We are engineers
Apple
1234apple
Juikmnhh
Vishal Acharya
```

```
Writing word.txt
```

33 48 print all three letter word

[2]:
```
wordlist = [line.strip() for line in open('word.txt')]
for word in wordlist:
    #print(word)
    for i in word.split():
        if len(i)==3:
            print(i)
```

```
Cse
are
```

Vishal Acharya

34 49 print word with start st or Ju

[3]:
```
wordlist = [line.strip() for line in open('word.txt')]
for word in wordlist:
    #print(word)
    for i in word.split():
        if i[0:2]=="st" or i[0:2]=="Ju":
            print(i)
```

```
student
student
Juikmnhh
```

35 50 Determine what percentage of words start with a vowel.

[5]:
```
f=open("word.txt")
word_count=0
count=0
for line in f:
    line=line.strip()
    for word in line.split():
        word_count=word_count+1
        if ((word[0] in "aeiou") or (word[0] in "AEIOU")):
            count=count+1
print(count/word_count,"%","start with a vowel")
print(word_count,count)
f.close()
```

```
0.46153846153846156 % start with a vowel
13 6
```

36 51 Find the longest word that can be made using the letters “s” and”t” in word

[4]:
```
f=open("words.txt")
largest = 0
largest_word=""
for line in f:
    line=line.strip()
    for i in line.split():
        if ("s" in i) and ("t" in i):
            if len(i)>largest:
                largest=len(i)
                largest_word=i
print(largest_word)
f.close()
```

Vishal Acharya

```
staudebntc
```

[29]:
```
%%writefile words.txt
Cse staudebntc
I am student
We are engineers
Apple
1234apple
Juikmnhh
Vishal Acharya
```

```
Overwriting words.txt
```

37 52 Use word.txt to find all the words that can be made from a user's string

[10]:
```
user=input("enter string: ")
d={}
for i in user:
    d[i]=d.get(i,0)+1
f=open("words.txt")
for line in f:
    line=line.strip()
    for i in line.split():
        e={}
        for x in i:
            e[x]=e.get(x,0)+1
            flag=True
        for j in d:
            if j not in e or d[j]>e[j]:
                flag=False
        if flag:
                print(i)
f.close()
```

```
enter string: abc
staudebntc
```

38 53 You are given a file called class_scores.txt, where each line of thefile contains a oneword

username and a test score separated by spaces, like below:.

GWashington 83

JAdams 86

Vishal Acharya

- Write code that scans through the file, adds 5 points to each test score, and outputs the usernames and new test scores to a new file,scores2.txt.

[16]:
```
lines = [line.strip() for line in open("class_scores.txt")]
inside_lines = [line.split(' ') for line in lines]
#print("inside lines",inside_lines)
for num in inside_lines:
    num[1]=str(int(num[1])+5)

lines=[" ".join(line) for line in inside_lines]

f=open('scores2.txt','w')
for line in lines:
    f.write(line+"\n")

f.close()
f=open('scores2.txt')
print(f.read())
f.close()
```

```
GWashington 88
JAdams 91
TJefferson 84
JMadison 102
JMonroe 72
JQAdams 97
AJackson 82
MVanBuren 66
JKPolk 96
```

39 54 You are given a file called grades.txt, where each line of the filecontains a one-word student

username and three test scores separated by spaces, like below:.

GWashington 83 77 54 JAdams 86 69 90

Write code that scans through the file and determines how many students passed all three tests

[28]:
```
passed_all3 =[]
failed_atleast1 =[]
with open("grades.txt") as f:
    s = f.readlines()
structure_content = []
for lines in s:
    lines=lines.strip()
    structure_content.append(lines.split())
```

Vishal Acharya

```
for score in structure_content:
    c = 0
    print(score)
    for i in range(1, len(score)):

        if int(score[i]) > 50 :
            c +=1
        if c == 3:
            passed_all3.append(score[0])
    if c !=3 :
        failed_atleast1.append(score[0])
print("Student who passed all three Subjects:",passed_all3,"Their number␣
 ↪is",len(passed_all3))
print("Student who Failed Atleast 1 Subject:",failed_atleast1,"Their number␣
 ↪is",len(failed_atleast1))
```

```
['GWashington', '83', '77', '54']
['JAdams', '86', '69', '90']
['TJefferson', '79', '57', '99']
['JMadison', '97', '99', '87']
['JMonroe', '67', '38', '90']
['JQAdams', '92', '18', '49']
['AJackson', '77', '99', '56']
['MVanBuren', '61', '78', '98']
['JKPolk', '91', '98', '39']
Student who passed all three Subjects: ['GWashington', 'JAdams', 'TJefferson',
'JMadison', 'AJackson', 'MVanBuren'] Their number is 6
Student who Failed Atleast 1 Subject: ['JMonroe', 'JQAdams', 'JKPolk'] Their
number is 3
```

40 55 You are given a file called logfile.txt that lists log-on and log-offtimes for users of a system.

A typical line of the file looks like this:

Van Rossum, 14:22, 14:37

Each line has three entries separated by commas: a username, a log-on time, and a log-off time. Times are given in 24-hour format. You mayassume that all log-ons and log-offs occur within a single workday.

Write a program that scans through the file and prints out all users who were online for at least an hour

[30]:
```
file = open("logfile.txt")

for line in file:
    L=line.split()
```

Vishal Acharya

```
    log_in=int(L[1].split(':')[0])+int(L[1].split(':')[1])/60
    log_of=int(L[2].split(':')[0])+int(L[2].split(':')[1])/60
    time_onine = log_of - log_in
        #print(log_of)
        #print("log in",log_in)
    if time_onine >= 1:
            print(L[0]," Have Been Active For Over An Hour.",time_onine)
```

```
Richie  Have Been Active For Over An Hour. 1.700000000000001
Armstrong  Have Been Active For Over An Hour. 1.3000000000000007
Wirth  Have Been Active For Over An Hour. 2.7666666666666675
Matz  Have Been Active For Over An Hour. 2.299999999999999
Gosling  Have Been Active For Over An Hour. 1.5500000000000007
```

41 56 You are given a file called students.txt. A typical line in the filelooks like:

walter melon melon@email.msmary.edu (mailto:melon@email.msmary.edu) 555-3141 There is a name, an email address, and a phone number, each separated by tabs. Write a program that reads through the file line-by-line,and for each line, capitalizes the first letter of the first and last name and adds the area code 301 to the phone number. Your program shouldwrite this to a new file called students2.txt. Here is what the first line of the new

file should look like:

Walter Melon melon@email.msmary.edu (mailto:melon@email.msmary.edu) 301-555-3141

[21]:
```
with open("students.txt") as student:
    s =student.readlines()
lines= [line.strip() for line in s]
structure_content = [line.split("\t") for line in lines]
#print("structured Content",structure_content)
for data in structure_content:
    data[0] = data[0].title()
    data[1] = data[1].title()
    data[2] = "305-" + data[2]
#print("new list",structure_content)
f=open("student2.txt","w")
for i in structure_content:
        f.writelines(i)
        f.write("\n")
f.close()
f=open("student2.txt","r")
print(f.read())
```

```
Walter MelonMelon@Email.Msmary.Edu305-447-3141
Warren PiecePiece@Email.Msmary.Edu305-447-5926
Winnie BagoBago@Email.Msmary.Edu305-447-5358
```

Vishal Acharya

Telly VisionVision@Email.Msmary.Edu305-447-9793
Shirley KnotKnot@Email.Msmary.Edu305-447-2384

42 57 You are given a file namelist.txt that contains a bunch of names.Some of the names are a first name and a last name separated byspaces, like George Washington, while others have a middle name,like John Quincy Adams. There are no names consisting of just oneword or more than three words. Write a program that asks the user toenter initials, like GW or JQA, and prints all the names that matchthose initials. Note that initials like JA should match both John Adamsand John Quincy Adams.

[38]:
```
user=input("enter user Initial: ")

with open("namelist.txt") as n:
    for line in n:
        L=line.split()
        if len(user)==2:
            if user[0]==L[0][0] and user[1]==L[-1][0]:
                print(" ".join(L))
        if len(user)==3 and len(L)==3:
            if user[0]==L[0][0] and user[1]==L[1][0] and user[2]==L[2][0]:
                print(" ".join(L))
```

```
enter user Initial:  VA

Vishal H Acharya
Vishal Acharya
```

43 58Given a file baseball.txt of baseball stats, print out all the players that hit at least 20 homeruns with an average of at least .300.

[44]:
```
lines = [line.strip() for line in open('baseball.txt')]
for line in lines:
    try:
        L = line.split('\t')
        name = L[0]
        team = L[1]
        home_runs = L[8]
        average = L[14]
```

Vishal Acharya

```
        if int(home_runs) >= 20 and float(average) >= .300:
            print(name, team, home_runs, average)
    except :
        print(" data not define")
```

```
Robinson Cano  NYY  29 0.319
Ryan Braun  MIL  25 0.304
Matt Holliday  STL  28 0.312
Vladimir Guerrero  BAL  29 0.300
Adrian Beltre  TEX  28 0.321
Carlos Gonzalez  COL  34 0.336
Albert Pujols  STL  42 0.312
Shin-Soo Choo  CLE  22 0.300
Paul Konerko  CWS  39 0.312
Miguel Cabrera  DET  38 0.328
Joey Votto  CIN  37 0.324
Hanley Ramirez  FLA  21 0.300
Ryan Zimmerman  WAS  25 0.307
Josh Hamilton  TEX  32 0.359
Victor Martinez  DET  20 0.302
Troy Tulowitzki  COL  27 0.315
Nelson Cruz  TEX  22 0.318
 data not define
```

[46]:
```
file1 = open("baseball.txt","r")
#print(file1.read())
str_data =[lines.split() for lines in file1]
for player in str_data:
    if len(player) > 13:
        if int(player[9]) > 20 and int(player[13])>= 20:
            print("Player Names: ", player[0:2])
```

```
Player Names:  ['Carlos', 'Gonzalez']
Player Names:  ['Chris', 'Young']
Player Names:  ['Alex', 'Rios']
Player Names:  ['Shin-Soo', 'Choo']
Player Names:  ['Hanley', 'Ramirez']
Player Names:  ['Drew', 'Stubbs']
```

44 59 For this problem, use the file of NCAA basketball scores .

(a) Find the average of the points scored over all the games in the file.

(b) Pick your favorite team and scan through the file to determine how many games they won and how many games they lost.

(c) Find the team(s) that lost by 30 or more points the most times

Vishal Acharya

(d) Find all the teams that averaged at least 70 points a game.

(e) Find all the teams that had winning records but were collectively outscored by their opponents.

A team is collectively outscored by their opponents if the total number of points the team scored over all their games is less than the totalnumber of points their opponents scored in their games against the team

[30]:
```
# Find the average of the points scored over all the games in the file.
f=open("scores.txt")
scores=[]
for line in f:
    scores.extend([int(line.split()[2]),int(line.split()[4])])
print(f"avg. score is {sum(scores)*2/len(scores)}")
```

```
avg. score is 137.6658566221142
```

[2]:
```
#Pick your favorite team and scan through the file to determine how many games␣
 ↪they won and how many games they lost.
my_team="AlcornSt."
f=open("scores.txt")
win=0
loss=0
draw=0
for line in f:
    l=line.split()
    l[2]=int(l[2])
    l[4]=int(l[4])
    if my_team==l[1]:
        if l[2]>l[4]:
            win+=1
        elif l[2]==l[4]:
            draw+=1
        else:
            loss+=1
    if my_team==l[3]:
        if l[2]<l[4]:
            win+=1
        elif l[2]==l[4]:
            draw+=1
        else:
            loss+=1
print(f"total match - {my_team}= {win+loss+draw}")
print(f" % of win ({win}) - {my_team} ={win/(win+loss+draw)}")
print(f" % of loss ({loss}) - {my_team} ={loss/(win+loss+draw)}")
```

```
total match - AlcornSt.= 31
 % of win (2) - AlcornSt. =0.06451612903225806
 % of loss (29) - AlcornSt. =0.9354838709677419
```

Vishal Acharya

[34]:
```
# Find the team(s) that lost by 30 or more points the most times
f=open("scores.txt")
l=[]
for line in f:
    details=line.split()
    if (int(details[2])-int(details[4]))>=30:
        l.append([details[3],(int(details[2])-int(details[4]))])
    if (int(details[4])-int(details[2]))>=30:
        l.append([details[1],(int(details[4])-int(details[2]))])
d={}
for i in l:
    d[i[0]]=d.get(i[0],[])+[i[1]]

k=sorted(d.items(),key= lambda x:len(x[1]),reverse=True)
for i,j in k:
    print(i,"that lost by 30 or more points the" ,len(j) )
```

```
AlcornSt. that lost by 30 or more points the 8
Presbyterian that lost by 30 or more points the 6
NJInstofTechnology that lost by 30 or more points the 6
GardnerWebb that lost by 30 or more points the 5
Centenary that lost by 30 or more points the 5
NorthCarolinaCentral that lost by 30 or more points the 4
TennesseeTech that lost by 30 or more points the 4
WinstonSalemSt. that lost by 30 or more points the 4
AlabamaA&M that lost by 30 or more points the 4
Longwood that lost by 30 or more points the 4
SIUEdwardsville that lost by 30 or more points the 4
Southern that lost by 30 or more points the 4
HoustonBaptist that lost by 30 or more points the 4
Oakland that lost by 30 or more points the 4
Howard that lost by 30 or more points the 4
LeesMcRae that lost by 30 or more points the 3
Bryant that lost by 30 or more points the 3
SacramentoSt. that lost by 30 or more points the 3
FloridaA&M that lost by 30 or more points the 3
EastCarolina that lost by 30 or more points the 3
VirginiaMilitaryInst that lost by 30 or more points the 3
MayvilleSt. that lost by 30 or more points the 3
NCAsheville that lost by 30 or more points the 3
IdahoSt. that lost by 30 or more points the 3
Stetson that lost by 30 or more points the 3
MississippiValleySt. that lost by 30 or more points the 3
SoutheastMissouriSt. that lost by 30 or more points the 3
SouthernUtah that lost by 30 or more points the 3
Toledo that lost by 30 or more points the 3
ChicagoSt. that lost by 30 or more points the 3
Hartford that lost by 30 or more points the 3
```

Vishal Acharya

MarylandEasternShore that lost by 30 or more points the 3
Towson that lost by 30 or more points the 3
WesternIllinois that lost by 30 or more points the 2
Dartmouth that lost by 30 or more points the 2
VirginiaIntermont that lost by 30 or more points the 2
NCGreensboro that lost by 30 or more points the 2
NewMexicoSt. that lost by 30 or more points the 2
FloridaInternational that lost by 30 or more points the 2
PortlandSt. that lost by 30 or more points the 2
Brescia that lost by 30 or more points the 2
Pennsylvania that lost by 30 or more points the 2
AlabamaSt. that lost by 30 or more points the 2
Liberty that lost by 30 or more points the 2
Arkansas that lost by 30 or more points the 2
NichollsSt. that lost by 30 or more points the 2
Charlotte that lost by 30 or more points the 2
ChampionBaptist that lost by 30 or more points the 2
CentralArkansas that lost by 30 or more points the 2
SantaClara that lost by 30 or more points the 2
MDBaltimoreCounty that lost by 30 or more points the 2
Radford that lost by 30 or more points the 2
Mercer that lost by 30 or more points the 2
Grambling that lost by 30 or more points the 2
TexasPanAmerican that lost by 30 or more points the 2
Delaware that lost by 30 or more points the 2
TennesseeSt. that lost by 30 or more points the 2
Chattanooga that lost by 30 or more points the 2
CoppinSt. that lost by 30 or more points the 2
St.FrancisPA that lost by 30 or more points the 2
NorthernArizona that lost by 30 or more points the 2
SanDiego that lost by 30 or more points the 2
Elon that lost by 30 or more points the 2
TexasSouthern that lost by 30 or more points the 2
CalSt.Bakersfield that lost by 30 or more points the 2
Oregon that lost by 30 or more points the 2
Buffalo that lost by 30 or more points the 2
TexasSt. that lost by 30 or more points the 2
UtahValley that lost by 30 or more points the 2
JacksonSt. that lost by 30 or more points the 2
NorthernIllinois that lost by 30 or more points the 2
Fordham that lost by 30 or more points the 2
EasternWashington that lost by 30 or more points the 2
NorthDakota that lost by 30 or more points the 2
IllinoisChicago that lost by 30 or more points the 2
MissouriKansasCity that lost by 30 or more points the 2
NCWilmington that lost by 30 or more points the 2
KennesawSt. that lost by 30 or more points the 2
ColoradoSt. that lost by 30 or more points the 2

Vishal Acharya

```
AirForce that lost by 30 or more points the 2
SaintJoseph's that lost by 30 or more points the 2
Albany that lost by 30 or more points the 1
RobertMorris that lost by 30 or more points the 1
Detroit that lost by 30 or more points the 1
Reinhardt that lost by 30 or more points the 1
NebraskaKearney that lost by 30 or more points the 1
Hofstra that lost by 30 or more points the 1
LoyolaChicago that lost by 30 or more points the 1
CharlestonSouthern that lost by 30 or more points the 1
FloridaGulfCoast that lost by 30 or more points the 1
SouthwesternAZ that lost by 30 or more points the 1
ScienceandArtsOK that lost by 30 or more points the 1
HowardPayne that lost by 30 or more points the 1
SouthDakota that lost by 30 or more points the 1
HopeInternational that lost by 30 or more points the 1
WestVirginiaWesleyan that lost by 30 or more points the 1
Covenant that lost by 30 or more points the 1
NCPembroke that lost by 30 or more points the 1
SouthernVirginia that lost by 30 or more points the 1
NorthernMichigan that lost by 30 or more points the 1
MissouriSt.Louis that lost by 30 or more points the 1
Suffolk that lost by 30 or more points the 1
UCDavis that lost by 30 or more points the 1
Schreiner that lost by 30 or more points the 1
LeTourneau that lost by 30 or more points the 1
Piedmont that lost by 30 or more points the 1
TrumanSt. that lost by 30 or more points the 1
PittsburgSt. that lost by 30 or more points the 1
BethuneCookman that lost by 30 or more points the 1
UCIrvine that lost by 30 or more points the 1
Hiram that lost by 30 or more points the 1
MountSaintMary that lost by 30 or more points the 1
NovaSoutheastern that lost by 30 or more points the 1
StephenF.Austin that lost by 30 or more points the 1
TexasCollege that lost by 30 or more points the 1
NWOklahomaSt. that lost by 30 or more points the 1
SaintJoseph'sNY that lost by 30 or more points the 1
MacMurray that lost by 30 or more points the 1
BowlingGreen that lost by 30 or more points the 1
Fisk that lost by 30 or more points the 1
Marist that lost by 30 or more points the 1
WesternCarolina that lost by 30 or more points the 1
PeruSt. that lost by 30 or more points the 1
St.Gregory that lost by 30 or more points the 1
Rider that lost by 30 or more points the 1
Southwest that lost by 30 or more points the 1
SanFrancisco that lost by 30 or more points the 1
```

Vishal Acharya

IUPUFortWayne that lost by 30 or more points the 1
FarmingdaleSt. that lost by 30 or more points the 1
AtlantaChristian that lost by 30 or more points the 1
FDUFlorham that lost by 30 or more points the 1
Voorhees that lost by 30 or more points the 1
LouisianaCollege that lost by 30 or more points the 1
RochesterMI that lost by 30 or more points the 1
SacredHeart that lost by 30 or more points the 1
Valparaiso that lost by 30 or more points the 1
HardinSimmons that lost by 30 or more points the 1
TrinityFL that lost by 30 or more points the 1
Winthrop that lost by 30 or more points the 1
Vermont that lost by 30 or more points the 1
Geneva that lost by 30 or more points the 1
Jacksonville that lost by 30 or more points the 1
IndianaEast that lost by 30 or more points the 1
FredoniaSt. that lost by 30 or more points the 1
CalMaritime that lost by 30 or more points the 1
JarvisChristian that lost by 30 or more points the 1
ArkansasFortSmith that lost by 30 or more points the 1
CumberlandTN that lost by 30 or more points the 1
VirginiaWise that lost by 30 or more points the 1
Allen that lost by 30 or more points the 1
EasternIllinois that lost by 30 or more points the 1
Lafayette that lost by 30 or more points the 1
St.Ambrose that lost by 30 or more points the 1
Tenn.Wesleyan that lost by 30 or more points the 1
Dana that lost by 30 or more points the 1
Millsaps that lost by 30 or more points the 1
SpringHill that lost by 30 or more points the 1
Massachusetts that lost by 30 or more points the 1
Greenville that lost by 30 or more points the 1
MidlandLthrn that lost by 30 or more points the 1
Colgate that lost by 30 or more points the 1
FreedHardeman that lost by 30 or more points the 1
MidAmericaChristian that lost by 30 or more points the 1
ToccoaFalls that lost by 30 or more points the 1
PaulQuinn that lost by 30 or more points the 1
Cameron that lost by 30 or more points the 1
DakotaSt. that lost by 30 or more points the 1
SUNYCobleskill that lost by 30 or more points the 1
SalemInternational that lost by 30 or more points the 1
EasternMichigan that lost by 30 or more points the 1
Samford that lost by 30 or more points the 1
St.Bonaventure that lost by 30 or more points the 1
Brown that lost by 30 or more points the 1
Maine that lost by 30 or more points the 1
MoreheadSt. that lost by 30 or more points the 1

Vishal Acharya

LongBeachSt. that lost by 30 or more points the 1
MorganSt. that lost by 30 or more points the 1
PrairieViewA&M that lost by 30 or more points the 1
AugustanaIL that lost by 30 or more points the 1
Linfield that lost by 30 or more points the 1
CulverStockton that lost by 30 or more points the 1
DePaul that lost by 30 or more points the 1
MainePresqueIsle that lost by 30 or more points the 1
Ecclesia that lost by 30 or more points the 1
KentuckyChristian that lost by 30 or more points the 1
IndianaSouthBend that lost by 30 or more points the 1
Wilberforce that lost by 30 or more points the 1
BridgewaterVA that lost by 30 or more points the 1
DelawareSt. that lost by 30 or more points the 1
SavannahSt. that lost by 30 or more points the 1
Malone that lost by 30 or more points the 1
CSUMontereyBay that lost by 30 or more points the 1
SouthCarolinaSt. that lost by 30 or more points the 1
Marian that lost by 30 or more points the 1
FairleighDickinson that lost by 30 or more points the 1
OaklandCity that lost by 30 or more points the 1
WilliamsBaptist that lost by 30 or more points the 1
NorthernNewMexico that lost by 30 or more points the 1
Occidental that lost by 30 or more points the 1
WesternMichigan that lost by 30 or more points the 1
Roanoke that lost by 30 or more points the 1
Montreat that lost by 30 or more points the 1
CarverBible that lost by 30 or more points the 1
OralRoberts that lost by 30 or more points the 1
NorthFlorida that lost by 30 or more points the 1
CalPoly that lost by 30 or more points the 1
Berry that lost by 30 or more points the 1
CollegeofNewJersey that lost by 30 or more points the 1
Vassar that lost by 30 or more points the 1
Gonzaga that lost by 30 or more points the 1
Fla.Christian that lost by 30 or more points the 1
CollegeofCharleston that lost by 30 or more points the 1
Portland that lost by 30 or more points the 1
SouthCarolinaUpstate that lost by 30 or more points the 1
Bryan that lost by 30 or more points the 1
HighPoint that lost by 30 or more points the 1
TexasArlington that lost by 30 or more points the 1
Drexel that lost by 30 or more points the 1
TennesseeTemple that lost by 30 or more points the 1
WilliamJessup that lost by 30 or more points the 1
Bucknell that lost by 30 or more points the 1
Marshall that lost by 30 or more points the 1
Navy that lost by 30 or more points the 1

Vishal Acharya

CalSt.Northridge that lost by 30 or more points the 1
EastCentral that lost by 30 or more points the 1
Franklin that lost by 30 or more points the 1
Idaho that lost by 30 or more points the 1
SouthDakotaSt. that lost by 30 or more points the 1
GreatFalls that lost by 30 or more points the 1
Tulsa that lost by 30 or more points the 1
CornellIA that lost by 30 or more points the 1
PanhandleSt. that lost by 30 or more points the 1
SamHoustonSt. that lost by 30 or more points the 1
PurdueN.Central that lost by 30 or more points the 1
Alma that lost by 30 or more points the 1
Arizona that lost by 30 or more points the 1
NorfolkSt. that lost by 30 or more points the 1
Belmont that lost by 30 or more points the 1
WesternOregon that lost by 30 or more points the 1
CentralBaptist that lost by 30 or more points the 1
Polytechnic that lost by 30 or more points the 1
Manhattan that lost by 30 or more points the 1
Dillard that lost by 30 or more points the 1
UnionCollege that lost by 30 or more points the 1
North.NewMexico that lost by 30 or more points the 1
Whittier that lost by 30 or more points the 1
PomonaPitzer that lost by 30 or more points the 1
SouthernMississippi that lost by 30 or more points the 1
AdamsSt. that lost by 30 or more points the 1
HolyNames that lost by 30 or more points the 1
Temple that lost by 30 or more points the 1
EasternNewMexico that lost by 30 or more points the 1
TexasA&MCorpusChris that lost by 30 or more points the 1
ColumbiaUnion that lost by 30 or more points the 1
Berea that lost by 30 or more points the 1
MichiganDearborn that lost by 30 or more points the 1
Rutgers that lost by 30 or more points the 1
OregonSt. that lost by 30 or more points the 1
Oklahoma that lost by 30 or more points the 1
Clarkson that lost by 30 or more points the 1
Hawaii that lost by 30 or more points the 1
CentralConnecticut that lost by 30 or more points the 1
Stanford that lost by 30 or more points the 1
YoungstownSt. that lost by 30 or more points the 1
Providence that lost by 30 or more points the 1
SWAssembliesofGod that lost by 30 or more points the 1
LoyolaMarymount that lost by 30 or more points the 1
CentralFlorida that lost by 30 or more points the 1
Goucher that lost by 30 or more points the 1
Seattle that lost by 30 or more points the 1
Duquesne that lost by 30 or more points the 1

Vishal Acharya

```
KentuckySt. that lost by 30 or more points the 1
TennesseeMartin that lost by 30 or more points the 1
Harvard that lost by 30 or more points the 1
RioGrande that lost by 30 or more points the 1
BallSt. that lost by 30 or more points the 1
LouisianaSt. that lost by 30 or more points the 1
Wyoming that lost by 30 or more points the 1
Pepperdine that lost by 30 or more points the 1
Nebraska that lost by 30 or more points the 1
Binghamton that lost by 30 or more points the 1
SouthAlabama that lost by 30 or more points the 1
Indiana that lost by 30 or more points the 1
FresnoSt. that lost by 30 or more points the 1
JacksonvilleSt. that lost by 30 or more points the 1
TexasChristian that lost by 30 or more points the 1
NorthCarolina that lost by 30 or more points the 1
Iowa that lost by 30 or more points the 1
LouisianaTech that lost by 30 or more points the 1
WakeForest that lost by 30 or more points the 1
```

[36]:
```python
# Find all the teams that averaged at least 70 points a game.
f=open("scores.txt")
d={}
for i in f:
    l=i.split()
    d[l[1]]=d.get(l[1],[])+[int(l[2])]
    d[l[3]]=d.get(l[3],[])+[int(l[4])]
dict1={}
for i,j in d.items():
    dict1[i]=(sum(j)/len(j))
for i,j in dict1.items():
    if j>=70:
        print(i,j)
```

```
OhioSt. 74.05405405405405
NorthCarolina 74.54054054054055
Syracuse 80.91428571428571
MurraySt. 76.47222222222223
California 77.22857142857143
MorganSt. 76.02702702702703
AppalachianSt. 74.75675675675676
Arkansas 74.71875
Niagara 72.18181818181819
Auburn 75.6875
Valparaiso 73.25
NorfolkSt. 70.63333333333334
Baylor 76.97222222222223
PortlandSt. 78.0
```

Vishal Acharya

```
Belmont 73.09677419354838
LoyolaMarymount 75.82352941176471
BoiseSt. 74.0625
BrighamYoung 83.11111111111111
NCAsheville 74.06451612903226
Charlotte 73.41935483870968
Clemson 73.4375
CollegeofCharleston 74.97058823529412
CoastalCarolina 73.82857142857142
Colorado 74.61290322580645
Connecticut 70.0
Duke 76.975
Duquesne 70.34375
Fairfield 71.0
FloridaAtlantic 73.4
GeorgiaSouthern 70.625
Harvard 73.96551724137932
HoustonBaptist 71.12121212121212
Illinois 70.75
BostonUniversity 72.22857142857143
IowaSt. 72.59375
Hofstra 70.11764705882354
Kansas 81.58333333333333
KansasSt. 79.70270270270271
NorthGreenville 72.0
KentSt. 70.05882352941177
MoreheadSt. 71.57142857142857
Kentucky 79.26315789473684
Lafayette 72.5
Vermont 70.2
Centenary 70.89655172413794
Marquette 73.08823529411765
Maryland 79.66666666666667
Memphis 75.44117647058823
Mercer 76.18181818181819
MichiganSt. 71.83783783783784
King 80.0
ConcordiaSt.Paul 73.0
TennesseeTech 74.21875
Minnesota 72.37142857142857
Mississippi 78.37142857142857
MississippiSt. 72.83333333333333
NorthTexas 73.84848484848484
NorthernArizona 70.17857142857143
NorthernColorado 73.51515151515152
NorthwesternSt. 75.37931034482759
Ohio 74.67567567567568
Providence 82.38709677419355
```

Vishal Acharya

CalSt.Northridge 71.3125
Purdue 70.48571428571428
Quinnipiac 73.18181818181819
Navy 71.26666666666667
Radford 71.51612903225806
Lehigh 75.0909090909091
SacredHeart 71.96551724137932
NewMexicoSt. 78.29411764705883
St.Mary's 78.47058823529412
CalPoly 70.19354838709677
SanFrancisco 70.53333333333333
SetonHall 80.09375
SouthCarolina 71.83870967741936
SoutheasternLouisiana 72.74193548387096
St.Bonaventure 70.09677419354838
AustinPeay 72.5
Tennessee 73.51351351351352
Siena 75.11764705882354
TexasA&M 71.3529411764706
TexasSt. 77.16129032258064
SouthDakota 79.46875
TexasTech 76.68571428571428
Georgetown 73.52941176470588
Tulsa 71.42857142857143
Massachusetts 70.6875
Idaho 71.16129032258064
NorthDakotaSt. 70.58620689655173
TexasElPaso 75.27272727272727
Villanova 81.81818181818181
Longwood 74.09677419354838
VirginiaCommonwealth 76.02777777777777
VirginiaMilitaryInst 88.58620689655173
OralRoberts 70.0909090909091
WakeForest 72.80645161290323
Washington 79.19444444444444
WashingtonSt. 71.90322580645162
UtahSt. 73.11428571428571
WeberSt. 76.16129032258064
SouthDakotaSt. 74.23333333333333
Wyoming 70.64516129032258
Xavier 79.8
Cornell 74.8529411764706
Davidson 71.03225806451613
CalSt.Fullerton 74.29032258064517
Dayton 70.24324324324324
IUPUI 74.47222222222223
WestVirginiaWesleyan 81.5
EasternKentucky 71.24242424242425

Vishal Acharya

```
Portland 72.0625
GeorgiaTech 72.80555555555556
Gonzaga 76.88235294117646
NCPembroke 72.0
HighPoint 72.3
LaSalle 70.36666666666666
Lipscomb 79.1
LongBeachSt. 71.03030303030303
MiamiFL 72.42424242424242
Nevada 79.5
NewMexico 75.85714285714286
NotreDame 75.28571428571429
Oakland 76.45714285714286
Oklahoma 72.09677419354838
Seattle 80.29032258064517
OklahomaSt. 73.84848484848484
SamHoustonSt. 79.27272727272727
SouthernIllinois 70.13333333333334
DallasBaptist 84.0
TexasArlington 73.86666666666666
LouisianaTech 73.05714285714286
Troy 74.75757575757575
NevadaLasVegas 73.08823529411765
WesternCarolina 74.26470588235294
SanDiegoSt. 70.26470588235294
Texas 81.20588235294117
Marshall 79.91176470588235
Akron 70.62857142857143
WestVirginia 72.42105263157895
Florida 72.20588235294117
VirginiaTech 72.73529411764706
GeorgeWashington 70.3225806451613
Arizona 71.90322580645162
Buffalo 72.33333333333333
TexasSanAntonio 70.7
WilmingtonOH 74.0
WesternKentucky 71.23529411764706
WilliamCarey 71.0
Vanderbilt 76.87878787878788
MissouriSt. 71.44444444444444
Oakwood 81.0
SanJoseSt. 73.83870967741936
Lynchburg 81.5
Louisville 75.9090909090909
Houston 78.6
Missouri 77.26470588235294
Whitman 88.0
FloridaMemorial 82.0
```

Vishal Acharya

```
TrinityInt'lIL 84.0
RhodeIsland 76.02777777777777
EastCentral 74.0
MinotSt. 82.0
NCWesleyan 70.0
PeruSt. 73.0
BethanyCA 75.0
ColumbiaUnion 70.33333333333333
SouthernNewOrleans 71.25
DickinsonSt. 75.0
CalSt.SanBernardino 70.0
ArkansasFortSmith 73.75
HampdenSydney 74.0
Stillman 77.0
MidlandLthrn 71.0
Milligan 75.0
FortHaysSt. 76.0
AcademyofArt 72.0
SouthwestSt. 70.0
LSUShreveport 73.0
WarnerSouthern 99.0
EasternOregon 77.0
Vanguard 73.0
Catawba 81.0
OlivetNazarene 75.0
TexasWesleyan 85.0
Talledega 86.0
TexasLutheran 70.0
Willamette 90.0
EvergreenSt. 77.0
CentralMethodist 80.0
AdamsSt. 71.0
CarsonNewman 74.0
WestGeorgia 72.0
AlbertusMagnus 71.0
Randolph 88.0
Bluefield 74.0
RobertMorrisILSpr 71.0
RioGrande 73.0
```

[40]:
```
f=open("scores.txt")
games=[]
for line in f:
    l=line.split()
    games.append([l[1],int(l[2]),l[3],int(l[4])])
team_stats = {}
```

Vishal Acharya

```
    for team1, score1, team2, score2 in games:
        if team1 not in team_stats:
            team_stats[team1] = {"wins": 0, "losses": 0, "points_scored": 0,␣
↪"points_against": 0}
        if team2 not in team_stats:
            team_stats[team2] = {"wins": 0, "losses": 0, "points_scored": 0,␣
↪"points_against": 0}

        team_stats[team1]["points_scored"] += score1
        team_stats[team1]["points_against"] += score2
        team_stats[team2]["points_scored"] += score2
        team_stats[team2]["points_against"] += score1

        if score1 > score2:
            team_stats[team1]["wins"] += 1
            team_stats[team2]["losses"] += 1
        else:
            team_stats[team1]["losses"] += 1
            team_stats[team2]["wins"] += 1

    outscored_winning_teams = {}
    for team, stats in team_stats.items():
        if stats["wins"] > stats["losses"] and stats["points_scored"] <␣
↪stats["points_against"]:
            outscored_winning_teams[team] = (stats["wins"], stats["losses"])

    print("(e) Winning teams but collectively outscored:")
    print(outscored_winning_teams)
```

```
(e) Winning teams but collectively outscored:
{'LoyolaMarymount': (18, 16), 'ArkansasPineBluff': (18, 16), 'GeorgeMason': (17,
15), 'SouthAlabama': (17, 15), 'IndianaSt.': (17, 15), 'JacksonSt.': (19, 13),
'Rider': (17, 16), 'SouthCarolinaSt.': (18, 14), 'LongBeachSt.': (17, 16),
'Arizona': (16, 15), 'AlabamaSt.': (16, 15)}
```

Vishal Acharya

45 60 Benford's law states that in real data where the values are spreadacross several orders ofmagnitude, about 30% of the values will startwith the number 1, where as only about 4.6%of the values will startwith the number 9. This is contrary to what we might expect, namelythat values starting with 1 and 9 would be equally likely. Using the fileexpenses.txt which consists of a number of costs from an expenseaccount, determine what percentage start with each of the digits 1through 9. This technique is used by accountants to detect fraud

¶

[33]:
```
file1 = open("expenses.txt","r")
str_data =[line.strip("\n") for line in file1]
total = len(str_data)
percentage_occurence =[]
count = 0
for benford_law in range(1,10):
    for number in range(0,len(str_data)):
        value = str_data[number][:1]
        if int(str_data[number][:1]) == benford_law:
            count = count + 1
    percentage = count/total * 100
    percentage_occurence.append([benford_law,"Occurence Percentage is␣
 ↪",percentage])
    count = 0
#print(percentage_occurence)
for i in percentage_occurence:

    print(i[0],i[1],i[2])
```

```
1 Occurence Percentage is  23.0
2 Occurence Percentage is  15.0
3 Occurence Percentage is  9.0
4 Occurence Percentage is  16.0
5 Occurence Percentage is  15.0
6 Occurence Percentage is  8.0
7 Occurence Percentage is  4.0
8 Occurence Percentage is  5.0
9 Occurence Percentage is  5.0
```

Vishal Acharya

46 find japanese word in secretmessage.txt text file

[7]:
```
import string
f=open("secretmessage.txt",encoding="utf-8")
con=f.read()
s=""
#print(con)
for i in con:
    if (i.lower() in "abcdefghijklmnopqrstuvwxyz") or (i in string.punctuation)␣
 ↪or (i in (" ","\n")):
         continue
    else:
         s+=i
print(s)

f.close()
```

47 meaning of this word is first char of each line

[8]:
```
f=open("secretmessage.txt",encoding="utf-8")
meaning=""
for i in f:
    meaning+=i[0]
print(meaning)
```

```
EVENAMONKEYFALLSFROMATREE
```

[10]:
```
%%writefile data.txt
Python is a powerful programming language.
It is widely used in machine learning, data science, web development and␣
 ↪automation.

Learning Python improves logical thinking.
Python makes problem solving easier and faster.

Artificial Intelligence and Machine Learning are trending technologies.
Python supports AI, ML, Data Analysis and Automation tools.

12345 is not a word but Python can handle numeric strings too.
Welcome to the world of programming with Python.
```

```
Writing data.txt
```

Vishal Acharya

48 Print the longest word in a given file

[11]:
```
f = open("data.txt", "r")
words = f.read().split()
longest = max(words, key=len)
print("Longest word =", longest)
f.close()
```

```
Longest word = technologies.
```

49 Display only the words starting with vowel

[12]:
```
f = open("data.txt", "r")
words = f.read().split()
for w in words:
    if w[0].lower() in "aeiou":
        print(w)
f.close()
```

```
is
a
It
is
used
in
and
automation.
improves
easier
and
Artificial
Intelligence
and
are
AI,
Analysis
and
Automation
is
a
of
```

Vishal Acharya

50 Find smallest word in the file

[13]:
```
f = open("data.txt", "r")
words = f.read().split()
smallest = min(words, key=len)
print("Smallest word =", smallest)
f.close()
```

```
Smallest word = a
```

Vishal Acharya