**WEEK: 1**

**6.5** Write code using find() and string slicing (see section 6.10) to extract the number at the end of the line below. Convert the extracted value to a floating point number and print it out

s = "X-DSPAM-Confidence: 0.8475";

p=s.find('0')

s1=s[p:p+6]

f=float(s1)

print(f)

**WEEK: 2**

You can write any code you like in the window below. There are three files loaded and ready for you to open if you want to do file processing: "mbox-short.txt", "romeo.txt", and "words.txt"

s=' kishalay'

s1=s.upper()

print(s1)

s2=s1.strip()

print(s2)

**WEEK: 3**

**7.1** Write a program that prompts for a file name, then opens that file and reads through the file, and print the contents of the file in upper case. Use the file **words.txt** to produce the output below.

You can download the sample data at [http://www.py4e.com/code3/words.txt](https://www.py4e.com/code3/words.txt)

# Use words.txt as the file name

fname = input("Enter file name: ")

fh = open(fname)

s=fh.read()

s1=s.upper()

s1=s1.rstrip()

print(s1)

**7.2** 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](https://www.py4e.com/code3/mbox-short.txt) when you are testing below enter **mbox-short.txt** as the file name.

# Use the file name mbox-short.txt as the file name

fname = input("Enter file name: ")

fh = open(fname)

val=0

cnt=0

for i in fh:

if i.startswith("X-DSPAM-Confidence:") :

i=i.rstrip()

p=i.find('0')

s=i[p:]

fs=float(s)

val=val+fs

cnt=cnt+1

else:

continue

avg=val/cnt

print("Average spam confidence:",avg)

**WEEK: 4**

**8.4** 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 alphabetical order.

You can download the sample data at [http://www.py4e.com/code3/romeo.txt](https://www.py4e.com/code3/romeo.txt)

fname = input("Enter file name: ")

fh = open(fname)

a = list()

for line in fh:

line.rstrip()

x=line.split()

for i in x:

if i not in a:

a.append(i)

else: continue

a.sort()

print(a)

**8.5** 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:'.

You can download the sample data at [http://www.py4e.com/code3/mbox-short.txt](https://www.py4e.com/code3/mbox-short.txt)

fname = input("Enter file name: ")

fh = open(fname)

a=list()

cnt = 0

for i in fh:

if i.startswith('From '):

x=i.split()

a.append(x[1])

cnt=cnt+1

for i in a:

print(i)

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

**WEEK: 5**

**9.4** 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

name = input("Enter file:")

fh = open(name)

a=list()

d=dict()

for i in fh:

if i.startswith('From '):

x=i.split()

a.append(x[1])

for j in a:

d[j]=d.get(j,0)+1

bw=None

bc=None

for a,b in d.items():

if bc is None or b>bc:

bw=a

bc=b

print(bw,bc)

**WEEK: 6**

**10.2** 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 2008

Once you have accumulated the counts for each hour, print out the counts, sorted by hour as shown below.

name = input("Enter file:")

fh = open(name)

a=list()

d=dict()

for i in fh:

if i.startswith('From '):

x=i.split()

y=x[5].split(':')

a.append(y[0])

for j in a:

d[j]=d.get(j,0)+1

d=sorted(d.items())

for k,v in d:

print(k,v)