**Data Programming Assignment-3**

**Chapter 5:**

**Exercise 1:**

num\_total = 0

num\_count = 0

num\_avg = 0

while True:

try:

x = input("Enter a number: ")

if (x == "done"):

break

value = int(x)

num\_total = value + num\_total

num\_count = num\_count + 1

num\_avg = num\_total / num\_count

except:

print("Invalid input.")

print(num\_total, num\_count, num\_avg)

**Exercise 2:**

num\_total = 0

num\_count = 0

max\_num = 0

min\_num = 0

while True:

try:

num = input("Enter a number: ")

if (num == "done"):

break

value = int(num)

if max\_num is 0 and min\_num is 0:

max\_num = value

min\_num = value

elif value > max\_num:

max\_num = value

elif value < min\_num:

min\_num = value

num\_total = value + num\_total

num\_count = num\_count + 1

except:

print("Invalid input.")

print(num\_total, num\_count, max\_num, min\_num)

**Chapter 6:**

**Exercise 5:**

given\_str = 'X-DSPAM-Confidence:0.8475'

output\_str = slice(19, 25, None)

float\_str = given\_str[output\_str]

print(float(float\_str))

**Chapter 7:**

**Exercise 2:**

fname = input("Enter the desired file name: ")

if len(fname) == 0:

fname = 'Desktop/mbox-short.txt'

fhandle = open(fname)

no\_of\_lines = 0

total\_lines = 0

avg\_spam = 0

for line in fhandle:

if not line.startswith("X-DSPAM-Confidence:") : continue

no\_of\_lines = no\_of\_lines + 1

number = float(line[21:])

total\_lines = number + total\_lines

avg\_spam = tot / no\_of\_lines

print("Average spam confidence:", avg\_spam)

**Chapter 8:**

**Exercise 4:**

fname= 'Desktop/romeo.txt'

fhand=open(fname)

word\_list=list()

for word in fhand:

word= word.rstrip()

word= word.lower()

word= word.split()

for sep\_word in word:

if sep\_word not in word\_list:

word\_list.append(sep\_word)

word\_list.sort()

print (word\_list)

**Chapter 9:**

**Exercise 4:**

fname = input('Enter file name: ')

try:

fhandle = open(fname)

except:

print ('File cannot be opened:', fname)

exit()

email\_list = dict()

for line in fhandle:

if line.startswith('From '):

line = line.split()

email = line[1]

email\_list[email] = email\_list.get(email,0) + 1

most\_email = None

for keyword in email\_list:

if most\_email is None or email\_list[keyword] > most\_email:

most\_email = email\_list[keyword]

send\_email = keyword

print (send\_email, most\_email)

**Exercise 5:**

fname = input("Enter a file name: ")

email\_lines = [line.strip('\n') for line in open(fname, 'r')

if line.startswith("From ")]

email\_dict = {}

for line in email\_lines:

line = line.split()

email = line[1]

email\_domain = email.split("@")[1]

email\_dict[email\_domain] = email\_dict.get(email\_domain, 0) + 1

print (email\_dict)

**Chapter 10:**

**Exercise 2:**

dict\_hours = dict()

list\_words = list()

fname = input('Enter file name: ')

fhandle = open(fname)

for line in fhandle:

words\_list = line.split()

if len(words\_list) < 2 or words\_list[0] != 'From':

continue

track = words\_list[5].find(':')

hour = words\_list[5][:track]

if hour not in dict\_hours:

dict\_hours[hour] = 1

else:

dict\_hours[hour] += 1

for keyword, value in list(dict\_hours.items()):

list\_words.append((keyword, value))

list\_words.sort()

for keyword, value in list\_words:

print(keyword, value)

**Exercise 3:**

import string

count = 0

dict\_count = dict()

match\_word = list()

fname = input('Enter file name: ')

fhandle = open(fname)

for line in fhandle:

line = line.translate(str.maketrans('', '', string.digits))

line = line.translate(str.maketrans('', '', string.punctuation))

line = line.lower()

words\_list = line.split()

for word in words\_list:

for letter in word:

count += 1

if letter not in dict\_count:

dict\_count[letter] = 1

else:

dict\_count[letter] += 1

for keyword, value in list(dict\_count.items()):

match\_word.append((value / count, keyword))

match\_word.sort(reverse=True)

for keyword, value in relative\_lst:

print(keyword, value)