Viviyan _lab6

January 19, 2021

0.1 Question:3

Write a program for Stock Price Analysis File Creation: Continually prompt a user for stock name, followed by price values for 5 days. Each row indicates stock name and daily prices of one stock. Store these values in a text file called “stock-prices.txt”. Open the file in Append Mode. Prompt message “Do you want to continue? “ and stop reading values accordingly. Then, you can close your file. File Processing: Now, open your file for processing. Print stock name, minimum price, maximum price and average price values. You mcan also print which day stock price was lowest in the week and which day stock price was highest. So, modify your print statement to print stock name, minimum price & day of minimum price, maximum price & day of maximum price and average price values. (Hint: Use enumerate to get index values)

[3]:
```
while True:
    st_name=str(input("Enter the name:  "))
    file=open("stock_prices.txt","a")
    file.write(st_name)
    file.write("  ")
    for i in range(5):
        p=input()
        file.write(p)
        file.write("  ")
    file.write("\n")
    con = str(input("want to continue  :  "))
    if con =='n':
        break
file.close()
```

```
Enter the name:  apple
32
43
54
65
76
want to continue : m
Enter the name: Dell
32
43
54
65
76
want to continue : m
Enter the name: hp
34
45
```

```
56
67
78
want to continue : n
```

[5]:
```python
for st in open("stock_prices.txt","r").readlines():
    p_min=[]
    calc=st.split()
    print(calc[0])
    for i in range(1,6):
        p_min.append(int(calc[i]))
    print(min(p_min))
    print(max(p_min))
    av=sum(p_min)
    avg=av/5
    print(avg)
    print("\n")
```

```
mango
5
6
5.2


pineappl
e 4
8
6.0


sony
34
232
86.8


apple
34
98
66.6
blackberry
32
65


50.0
```

```
samsung
34
67
50.8


sony
21
54
34.6


apple
56
98
79.0


apple
34
67
47.2


32
43
87
65.0


21
23
67
50.4


sony
54
87
67.2
sony
32
76
54.0
```

```
samsung
43
89     6
5.4


apple
56
90     7
6.0


sony
21
65
43.0


samsung
21
76
47.4


apple
43
87
65.0


39
2
6
4.0


2
1
52
12.4
9
6
8
7.0
```

```
apple
3
7
5.0


dell
5
9
7.0


hp
3
9
6.6


dell
21
65
39.0


89
32
78
55.2


apple
65
98
83.2


apple
43
87
62.8
dell 54
89
74.2


hp
64
```

```
89
75.6


apple
32
76
54.0


hp
12
43
26.2


dell
45
89
68.8


apple
32
76
54.0


Dell
32
76
54.0


hp
34
78
56.0
```

[9]:
```
for st in open("stock_prices.txt","r").readlines():
    p_min=[]
    print("-------------------------")
    calc=st.split()
    print(calc[0])
    for i in range(1,6):
        p_min.append(int(calc[i]))
    mip=min(p_min)
    mxp=max(p_min)
    im=p_min.index(mip)
    ix=p_min.index(mxp)
    print("min price ",mip," on day ",im+1)
    print("max price ",mxp," on day ",ix+1)
```

```
-------------------------
mango
min price  5  on day  1
max price  6  on day  5
-------------------------
--pineapple
min price  4  on day  1
max price  8  on day  5
-------------------------
--sony
min price  34  on day  1
max price  232  on day  2
-------------------------
--apple
min price  34  on day  1
max price  98  on day  5
-------------------------
--blackberry
min price  32  on day  5
max price  65  on day  2
-------------------------
--samsung
min price  34  on day  1
max price  67  on day  4
-------------------------
--sony
min price  21  on day  1
max price  54  on day  5
-------------------------
--apple
min price  56  on day  1
max price  98  on day  5
```

```
----------------------
--apple
min price 34 on day 1
max price 67 on day 5
----------------------
---
32
min price 43 on day 1
max price 87 on day 5
----------------------
---
21
min price 23 on day 1
max price 67 on day 5
----------------------
--sony
min price 54 on day 4
max price 87 on day 1
----------------------
--sony
min price 32 on day 1
max price 76 on day 5
----------------------
--samsung
min price 43 on day 1
max price 89 on day 5
----------------------
--apple
min price 56 on day 5
max price 90 on day 1
----------------------
--sony
min price 21 on day 1
max price 65 on day 5
----------------------
--samsung
min price 21 on day 3
max price 76 on day 5
----------------------
--apple
min price 43 on day 1
max price 87 on day 5
----------------------
---
39
min price 2 on day 1
max price 6 on day 5
```

```
----------------------
---
2
min price 1 on day 4
max price 52 on day 3
----------------------
---
9
min price 6 on day 4
max price 8 on day 1
----------------------
--apple
min price 3 on day 1
max price 7 on day 5
----------------------
--dell
min price 5 on day 5
max price 9 on day 1
----------------------
--hp
min price 3 on day 5
max price 9 on day 1
----------------------
--dell
min price 21 on day 4
max price 65 on day 5
----------------------
---
89
min price 32 on day 5
max price 78 on day 1
----------------------
--apple
min price 65 on day 5
max price 98 on day 2
```

1 Question:4 Write a program for File Explorer

Display the contents of file 1.Count the number of lines in a text file. (Use splitlines()) 2.Count the number of unique words in a file. 3.Find frequency of words in a given file. (Hint: Use Counter object) 4.Show a random line in a file. (Use Random object)

[117]:
```
print("1.Display the contents of File :")
print("-------")
f = open("samplemv.txt",'r')
display = f.read()
print(display)
f.close()
print("")

print("2.Count the number of lines in a text file :")
print("-------")
file = open("samplemv.txt","r")
Counter = 0
```

```
Content = file.read()
CoList = Content.split("\n")

for i in CoList:
    if i:
        Counter += 1
print("Number of lines in the text file:",Counter)
print("\n")


print("3.Count the number of unique words in a file:")
print("-------")
num_words = 0
c = open("samplemv.txt", 'r')
for line in c:
    words = line.split()
    num_words += len(words)
print("Number of words:",num_words)
c.close()
print("\n")


print("4.Find Find frequency of words in a given file:")
print("--------")
fname = input('Enter the file name: ')
print("-------------------------------------------------------------------------------------------
try:
    fhand = open(fname)
    counts = dict()
    for line in fhand:
        words = line.split()
        for word in words:
            if word in counts:
                counts[word] += 1
            else:
                counts[word] = 1
    print(counts)
except:
    print('File cannot be opened:', fname)
print("\n")

print("5.Show a random line in a file:")
print("--------")
import random
def random_line(fname):
    lines = open(fname).read().splitlines()
    return random.choice(lines)
```

/
;
;
;
;
;
;

```
print(random_line('samplemv.txt'))
```

```
1.Display the contents of File:
-------
Hi This Is Maheshvaran
S
Hi This Is Mv
Hi This Is Mahe
Hi This Is 407mv
Hi This Is Mahesh

2.Count the number of lines in a text file:
-------
Number of lines in the text file: 5


3.Count the number of unique words in a file:
-------
Number of words: 21


4.Find Find frequency of words in a given file:
--------
Enter the file name: samplemv.txt
------------------------------------------------------------------------
---------
--------------------
{'Hi': 5, 'This': 5, 'Is': 5, 'Maheshvaran': 1, 'S': 1, 'Mv': 1,
'Mahe': 1, '407mv': 1, 'Mahesh': 1}


5.Show a random line in a file:
--------
Hi This Is Maheshvaran
S
```

1.1 Question5:

Develop an application in Python to read through the email data (“mbox-short.txt”) and when you find line that starts with “From”, you will split the line into words using the split function. We are interested in who

sent the message, which is the second word on the From line: From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008. You will parse the From line and print out the second word for each From line, then you will also count the number of From (not From:) lines and print out a count at the end.

[118]:
```
fhand = open('mbox-short.txt')
for line in fhand:
    line = line.rstrip()
    if line.startswith('From '):
        print(line)
```

```
From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008
From louis@media.berkeley.edu Fri Jan 4 18:10:48 2008
From zqian@umich.edu Fri Jan 4 16:10:39 2008
From rjlowe@iupui.edu Fri Jan 4 15:46:24 2008
From zqian@umich.edu Fri Jan 4 15:03:18 2008
From rjlowe@iupui.edu Fri Jan 4 14:50:18 2008
From cwen@iupui.edu Fri Jan 4 11:37:30 2008
From cwen@iupui.edu Fri Jan 4 11:35:08 2008
From gsilver@umich.edu Fri Jan 4 11:12:37 2008
From gsilver@umich.edu Fri Jan 4 11:11:52 2008
From zqian@umich.edu Fri Jan 4 11:11:03 2008
From gsilver@umich.edu Fri Jan 4 11:10:22 2008
From wagnermr@iupui.edu Fri Jan 4 10:38:42 2008
From zqian@umich.edu Fri Jan 4 10:17:43 2008
From antranig@caret.cam.ac.uk Fri Jan 4 10:04:14 2008
From gopal.ramasammycook@gmail.com Fri Jan 4 09:05:31 2008
From david.horwitz@uct.ac.za Fri Jan 4 07:02:32 2008
From david.horwitz@uct.ac.za Fri Jan 4 06:08:27 2008
From david.horwitz@uct.ac.za Fri Jan 4 04:49:08 2008
From david.horwitz@uct.ac.za Fri Jan 4 04:33:44 2008
From stephen.marquard@uct.ac.za Fri Jan 4 04:07:34 2008
From louis@media.berkeley.edu Thu Jan 3 19:51:21 2008
From louis@media.berkeley.edu Thu Jan 3 17:18:23 2008
From ray@media.berkeley.edu Thu Jan 3 17:07:00 2008
From cwen@iupui.edu Thu Jan 3 16:34:40
2008 From cwen@iupui.edu Thu Jan 3
16:29:07 2008
```

```
From cwen@iupui.edu Thu Jan  3 16:23:48 2008
```

[119]:
```
fhand = open("mbox-short.txt")
count = 0
for line in fhand:
    line = line.rstrip()
    if line == "": continue
    words = line.split()
    if words[0] !="From": continue
    print(words[1])
    count = count+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@gm
ail.com
david.horwitz@uct.ac.z
a
david.horwitz@uct.ac.z
a
david.horwitz@uct.ac.z
a
david.horwitz@uct.ac.z
a
stephen.marquard@uct.a
c.za
louis@media.berkeley.e
du
louis@media.berkeley.e
du
ray@media.berkeley.edu
cwen@iupui.edu
```

```
   cwen@iupui.edu
   cwen@iupui.edu
There were 27 lines in the file with From as the first
                                                      word
```

[ ]:

Question3. Write a program for *Stock Price Analysis*

- File Creation: Continually prompt a user for stock name, followed by price values for 5 days. Each row indicates stock name and daily prices of one stock. Store these values in a text file called **"stock-prices.txt"**. Open the file in Append Mode. Prompt message "Do you want to continue? " and stop reading values accordingly. Then, you can close your file.
- File Processing: Now, open your file for processing. Print stock name, minimum price, maximum price and average price values.
- You can also print which day stock price was lowest in the week and which day stock price was highest. So, modify your print statement to print stock name, minimum price & day of minimum price, maximum price & day of maximum price and average price values. (Hint: Use enumerate to get index values)

```
while True:
    st_name = str(input("Enter the name: "))
    file = open("stock_prices.txt", "a")
    file.write(st_name)
    file.write(" ")
    for i in range(5):
        p = input()
        file.write(p)
        file.write(" ")
    file.write("\n")
    con = str(input("want to continue: "))
    if con == 'n':
        break
file.close()
```

```
for st in open("stock_prices.txt", "r").readlines():
    p_min = []
    print("=" * 40)
    calc = st.split(",")
    print(calc[0])
    for i in range(1,6):
        p_min.append(int(calc[i]))
    mip = min(p_min)
    mxp = max(p_min)
    im = p_min.index(mip)
    ix = p_min.index(mxp)
    print("min price", mip, "on day", im+1)
    print("max price", mxp, "on day", ix+1)
```

```
for st in open("stock_prices.txt","r").readlines():
    P_min = []
    calc = st.split()
    print(calc[0])
    for i in range(1,6):
        P_min.append(int(calc[i]))
    print(min(P_min))
    print(max(P_min))
    av = sum(P_min)
    avg = av/5
    print(avg)
    print("\n")
```

Question4. Write a program for File Explorer

- Display the contents of file
- Count the number of lines in a text file. (Use splitlines())
- Count the number of unique words in a file.
- Find frequency of words in a given file. (Hint: Use Counter object)
- Show a random line in a file. (Use Random object)

```
print(" 1. Display the contents of file :")
print("....")
f = open("samplemv.txt", 'r')
display = f.read()
print(display)
f.close()
print(" ")
print(" 2. Count the number of lines in a text file :")
print("....")
file = open("samplemv.txt", 'r')
Counter = 0
Content = file.read()
CoList = Content.split("\n")
for i in CoList:
    if i:
        Counter += 1
print("Number of lines in the text file :", Counter)
print("\n")
print(" 3. Count the number of unique words in a file :")
print("....")
num_words = 0
c = open("samplemv.txt", 'r')
for line in c:
    words = line.split()
    num_words += len(words)
print("Number of words :", num_words)
c.close()
print("\n")
print(" 4. Find frequency of words in a given file :")
print("....")
fname = input("Enter the file name :")
print("....")
try:
    fhand = open(fname)
    counts = dict()
    for line in fhand:
        words = line.split()
        for word in words:
            if word in counts:
                counts[word] += 1
            else:
                counts[word] = 1
    print(counts)
except:
    print("File cannot be opened :", fname)
print("\n")
print(" 5. Show a random line in a file :")
print("....")
import random
def random_line(fname):
    lines = open(fname).read().splitlines()
    return random.choice(lines)
print(random_line('samplemv.txt'))
```

Question5. [File Searcher]. Develop an application in Python to read through the email data **("mbox-short.txt")** and when you find line that starts with "From", you will split the line into words using the split function. We are interested in who sent the message, which is the second word on the From line: From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008. You will parse the From line and print out the second word for each From line, then you will also count the number of From (not From:) lines and print out a count at the end

```
fhand = open('mbox-short.txt')
for line in fhand:
line = line.rstrip()
if line.startswith('From'):
    print(line)
fhand = open("mbox-short.txt")
  count = 0
for line in fhand:
    line = line.rstrip()
    if line == "": continue
    words = line.split()
    if words[0] != "From": continue
    print(words[1])
    count = count + 1
print("There were", count, "lines in the file with From as
                        the first word").
```