

Exercise 6.14

Exercise 5: Take the following Python code that stores a string: ' str = 'X-DSPAM-Confidence:0.8475' Use find and string slicing to extract the portion of the string after the colon character and then use the float function to convert the extracted string into a floating point number.

```
huhu.py > 614.py > ...
1  str = 'X-DSPAM-Confidence:0.8475'
2  colon_pos = str.find(':')
3  number_str = str[colon_pos + 1:]
4  number = float(number_str)
5  print(number)
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\New folder> & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/DELL/New folder/huhu.py/614.py"
0.8475

Exercise 6: Read the documentation of the string methods at <https://docs.python.org/3.5/library/stdtypes.html#string-methods>. You might want to experiment with some of them to make sure you understand how they work. `strip` and `replace` are particularly useful. The documentation uses a syntax that might be confusing. For example, in `find(sub[, start[, end]])`, the brackets indicate optional arguments. So `sub` is required, but `start` is optional, and if you include `start`, then `end` is optional.

The screenshot shows a code editor interface with a dark theme. At the top, there are three tabs: "Untitled-1.py" (active), "Untitled-2", and "Untitled-3". Below the tabs, the code editor displays the following Python script:

```
C: > Users > DELL > Untitled-1.py > ...
1 text = " hello world "
2 print(text.strip())
3 print(text.replace("world", "Python"))
4 print(text.find("hello"))
```

At the bottom of the screen, there is a terminal window showing the execution of the script and its output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DELL> & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe c:/Users/DELL/Untitled-1.py
hello world
hello Python
```

7.11 Exercises

Exercise 1: Write a program to read through a file and print the contents of the file (line by line) all in upper case. Executing the program will look as follows:

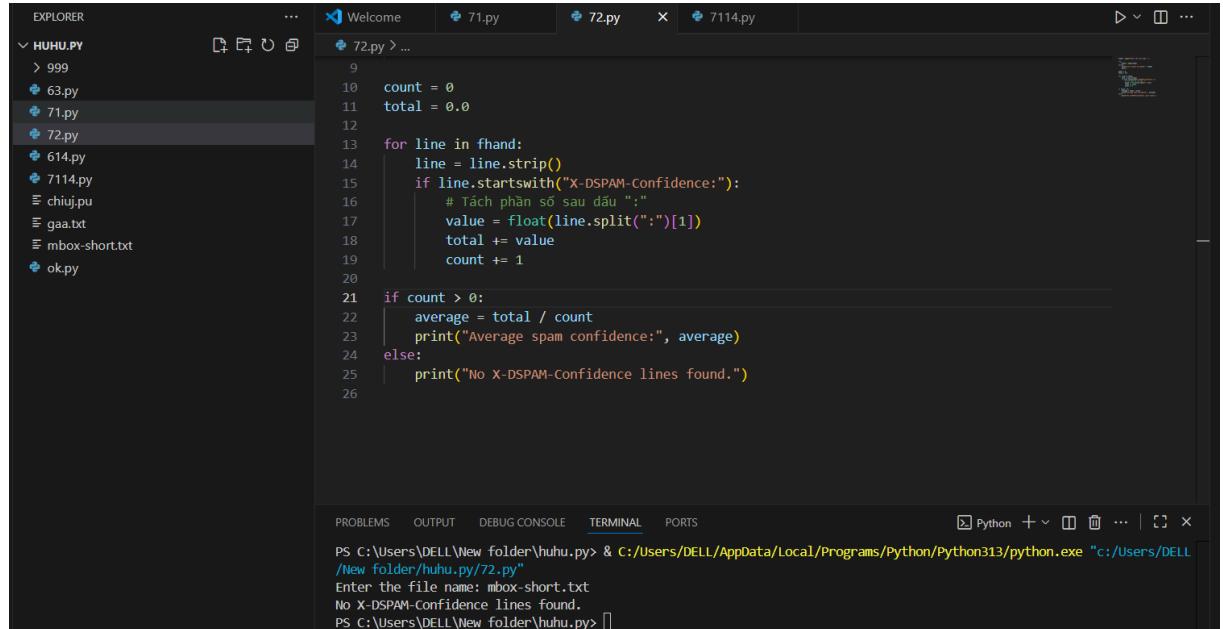
The screenshot shows the Visual Studio Code interface. The left sidebar (Explorer) lists files: HUHU.PY, 999, 63.py, 71.py (selected), 614.py, 7114.py, chiu.jpu, gaa.txt, mbox-short.txt, and ok.py. The main area (Editor) displays the code for 71.py:

```
1 name = input("Enter a file name: ")
2
3 try:
4     fhand = open(name)
5 except:
6     print("File cannot be opened:", name)
7     quit()
8
9 for line in fhand:
10    line = line.rstrip()
11    print(line.upper())
12
13
```

The bottom right corner shows the terminal window with the following output:

```
PS C:\Users\DELL\New folder\huhu.py & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe "C:/Users/DELL/New folder\huhi.py\71.py"
Enter a file name: mbox-short.txt
FROM STEPHEN.MARQUARD@UCT.AC.ZA SAT JAN 5 09:14:16 2008
RETURN-PATH: <POSTMASTER@COLLAB.SAKAIPROJECT.ORG>
RECEIVED: FROM MURDER (MAIL.UMICH.EDU [141.211.14.90])
BY FRANKENSTEIN.MAIL.UMICH.EDU (CYRUS V2.3.8) WITH LMTPA;
SAT, 05 JAN 2008 09:14:16 -0500
```

Exercise 2: Write a program to prompt for a file name, and then read through the file and look for lines of the form: X-DSPAM-Confidence:0.8475 When you encounter a line that starts with "X-DSPAM-Confidence:" pull apart the line to extract the floating-point number on the line. Count these lines and then compute the total of the spam confidence values from these lines. When you reach the end of the file, print out the average spam confidence. Enter the file name: mbox.txt Average spam confidence: 0.894128046745 Enter the file name: mbox-short.txt Average spam confidence: 0.750718518519



The screenshot shows a code editor interface with the following details:

- Explorer:** Shows a folder named "HUHU.PY" containing files: 999, 63.py, 71.py, 72.py, 614.py, 7114.py, chuij.pu, gaa.txt, mbox-short.txt, and ok.py. The file "72.py" is selected.
- Code Editor:** Displays the contents of "72.py". The code reads a file, extracts lines starting with "X-DSPAM-Confidence:", splits them to get the float value, and calculates the average.
- Terminal:** Shows the command "PS C:\Users\DELL\New folder\huhu.py> & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/DELL/New folder/huju.py/72.py"" followed by the user input "Enter the file name: mbox-short.txt", the output "No X-DSPAM-Confidence lines found.", and the command "PS C:\Users\DELL\New folder\huhu.py> []".

```

EXPLORER ... ✎ Welcome 71.py 72.py 7114.py
↳ HUHU.PY
  > 999
  63.py
  71.py
  72.py
  614.py
  7114.py
  chuij.pu
  gaa.txt
  mbox-short.txt
  ok.py

72.py > ...
9
10 count = 0
11 total = 0.0
12
13 for line in fhand:
14     line = line.strip()
15     if line.startswith("X-DSPAM-Confidence:"):
16         # Tách phần số sau dấu ":" 
17         value = float(line.split(":")[1])
18         total += value
19         count += 1
20
21 if count > 0:
22     average = total / count
23     print("Average spam confidence:", average)
24 else:
25     print("No X-DSPAM-Confidence lines found.")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DELL\New folder\huhu.py> & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/DELL/New folder/huju.py/72.py"
Enter the file name: mbox-short.txt
No X-DSPAM-Confidence lines found.
PS C:\Users\DELL\New folder\huhu.py> []

```

Exercise 3: Sometimes when programmers get bored or want to have a bit of fun, they add a harmless Easter Egg to their program. Modify the program that prompts the user for the file name so that it prints a funny message when the user types in the exact file name “na na boo boo”. The program should behave normally for all other files which exist and don’t exist. Here is a sample execution of the program:

The screenshot shows a Python code editor interface. On the left, the Explorer sidebar lists several files: 999, 63.py, 71.py, 72.py, 614.py (which is selected), 7114.py, chuij.py, gaa.txt, mbox-short.txt, and ok.py. The main workspace displays the contents of the 614.py file:

```
fname = input("Enter the file name: ")
# Nếu người dùng nhập câu đặc biệt
if fname.lower() == "na na boo boo":
    print("NA NA BOO BOO TO YOU - You have been punk'd!")
    quit()

# Thử mở file
try:
    fhand = open(fname)
except:
    print("File cannot be opened:", fname)
    quit()

count = 0

# Đếm số dòng bắt đầu bằng "subject:".
for line in fhand:
    if line.startswith("Subject:"):
        count += 1

print("There were", count, "subject lines in", fname)
```

Below the code editor is a terminal window showing the execution of the program:

```
PS C:\Users\DELL\New folder\huhu.py & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/DELL/New folder/huhu.py/614.py"
Enter the file name: mbox.txt
File cannot be opened: mbox.txt
PS C:\Users\DELL\New folder\huhu.py & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/DELL/New folder/huhu.py/614.py"
Enter the file name: mbox-short.txt
There were 0 subject lines in mbox-short.txt
PS C:\Users\DELL\New folder\huhu.py & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/DELL/New folder/huhu.py/614.py"
Enter the file name: na na boo boo
NA NA BOO BOO TO YOU - You have been punk'd!
PS C:\Users\DELL\New folder\huhu.py>
```