

Bài tập chương 6 - String

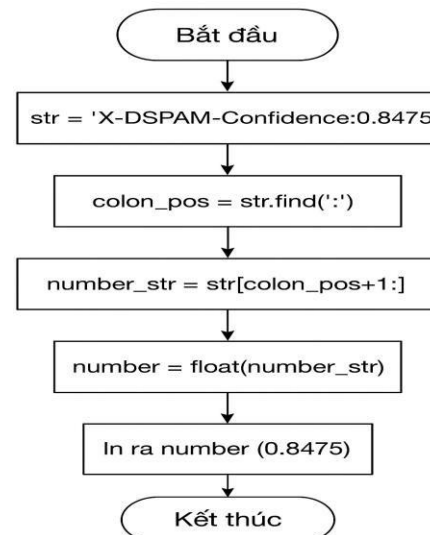
Exercise 5:

```
str = 'X-DSPAM-Confidence:0.8475'
# Tìm vị trí của dấu hai chấm
pos = str.find(':')
# Cắt chuỗi sau dấu hai chấm
number_str = str[pos+1:]
# Chuyển sang kiểu float
number = float(number_str)

print(number)
```

[1] ✓ 0.0s

... 0.8475



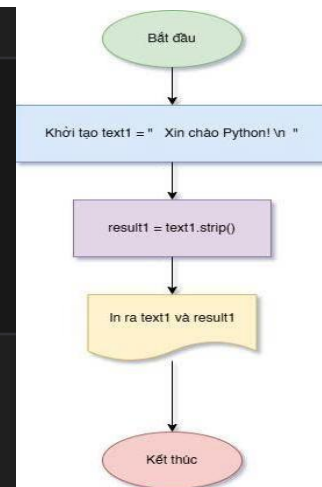
Excercise 6:

```
# Ví dụ 1: Loại bỏ khoảng trắng mặc định
text1 = "  Xin chào Python! \n "
result1 = text1.strip()
print(f"Chuỗi ban đầu: '{text1}'")
print(f"Chuỗi sau strip(): '{result1}'")
```

[1]

... Chuỗi ban đầu: ' Xin chào Python! \n '

Chuỗi sau strip(): 'Xin chào Python!'



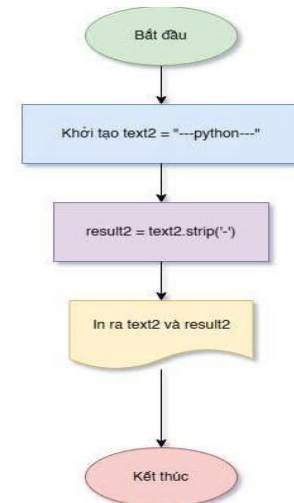
```
# Ví dụ 2: Loại bỏ ký tự cụ thể
text2 = "---python---"
# Loại bỏ các ký tự '-' ở đầu và cuối
result2 = text2.strip('-')
print(f"\nChuỗi ban đầu: '{text2}'")
print(f"Chuỗi sau strip('-'): '{result2}'")
```

[2]

...

Chuỗi ban đầu: '---python---'

Chuỗi sau strip('-'): 'python'



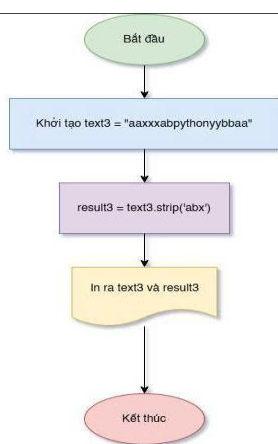
```
# Ví dụ 3: Loại bỏ tập hợp ký tự (thứ tự không quan trọng)
text3 = "aaxxxabpythonyybbaa"
# Loại bỏ 'a', 'b', và 'x' ở đầu và cuối
# Nó sẽ cắt cho đến khi gặp ký tự *không* thuộc tập hợp này
result3 = text3.strip('abx')
print(f"\nChuỗi ban đầu: '{text3}'")
print(f"Chuỗi sau strip('abx'): '{result3}'")
```

[3]

...

Chuỗi ban đầu: 'aaxxxabpythonyybbaa'

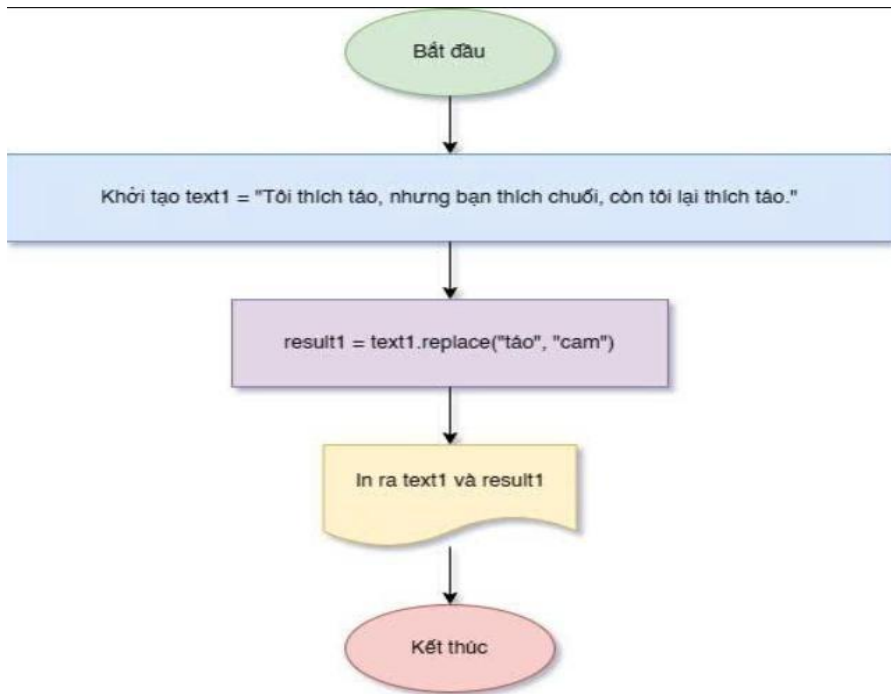
Chuỗi sau strip('abx'): 'pythonyy'



```
# Ví dụ 4: Thay thế tất cả các lần xuất hiện
text1 = "Tôi thích táo, nhưng bạn thích chuối, còn tôi lại thích táo."
result1 = text1.replace("táo", "cam")
print(f"Chuỗi ban đầu: {text1}")
print(f"Chuỗi sau replace(): {result1}")
```

[1] ✓ 0.0s

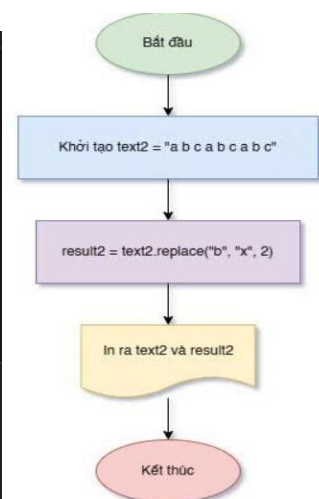
... Chuỗi ban đầu: Tôi thích táo, nhưng bạn thích chuối, còn tôi lại thích táo.
Chuỗi sau replace(): Tôi thích cam, nhưng bạn thích chuối, còn tôi lại thích cam.



```
# Ví dụ 5: Thay thế với giới hạn số lần (count)
text2 = "a b c a b c a b c"
# Thay thế 'b' bằng 'x' tối đa 2 lần
result2 = text2.replace("b", "x", 2)
print(f"\nChuỗi ban đầu: {text2}")
print(f"Chuỗi sau replace(..., 2): {result2}")
```

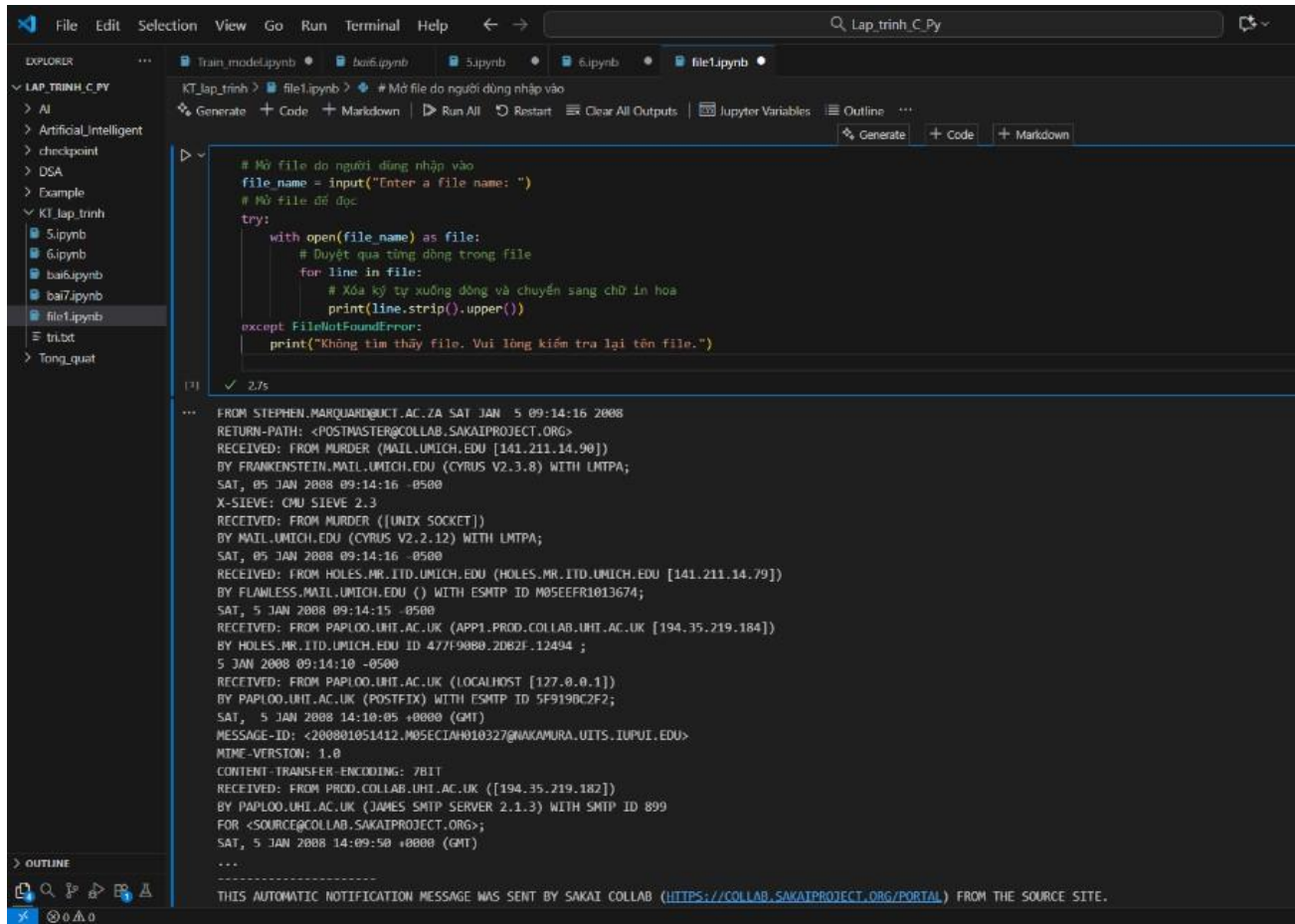
[2] ✓ 0.0s

... Chuỗi ban đầu: a b c a b c a b c
Chuỗi sau replace(..., 2): a x c a x c a b c



Chương 7 file

Exercise 1:



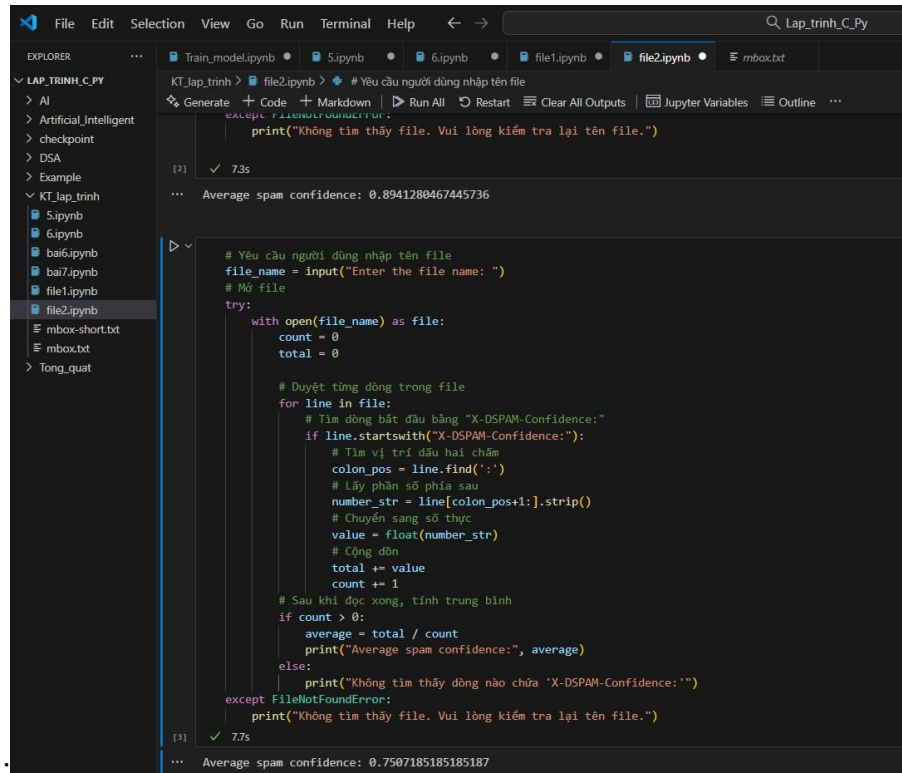
The screenshot displays a JupyterLab environment with a dark theme. The Explorer panel on the left shows a project structure with folders like 'AI', 'Artificial_Intelligent', 'checkpoint', 'DSA', 'Example', and 'KT_lap_trinh'. The 'KT_lap_trinh' folder is expanded, showing files '5.ipynb', '6.ipynb', 'ba6.ipynb', 'ba7.ipynb', and 'file1.ipynb'. The 'file1.ipynb' file is selected, and its code is visible in the main editor. The code is a Python script that prompts the user for a file name, attempts to open and read it, and prints the contents in uppercase. It includes error handling for 'FileNotFoundError'. The output of the script is shown in the console, displaying a detailed email header and body text.

```
# Mở file do người dùng nhập vào
file_name = input("Enter a file name: ")
# Mở file để đọc
try:
    with open(file_name) as file:
        # Duyệt qua từng dòng trong file
        for line in file:
            # Xóa ký tự xuống dòng và chuyển sang chữ in hoa
            print(line.strip().upper())
except FileNotFoundError:
    print("Không tìm thấy file. Vui lòng kiểm tra lại tên file.")
```

```
✓ 2.7s

...
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
X-SIEVE: CHU SIEVE 2.3
RECEIVED: FROM MURDER ([UNIX SOCKET])
BY MAIL.UMICH.EDU (CYRUS V2.2.12) WITH LMTPA;
SAT, 05 JAN 2008 09:14:16 -0500
RECEIVED: FROM HOLES.MR.ITD.UMICH.EDU (HOLES.MR.ITD.UMICH.EDU [141.211.14.79])
BY FLAMELESS.MAIL.UMICH.EDU ( ) WITH ESMTP ID M05EEFR1013674;
SAT, 5 JAN 2008 09:14:15 -0500
RECEIVED: FROM PAPLOO.UHI.AC.UK (APP1.PROD.COLLAB.UHI.AC.UK [194.35.219.184])
BY HOLES.MR.ITD.UMICH.EDU ID 477F9080.2D82F.12494 ;
5 JAN 2008 09:14:10 -0500
RECEIVED: FROM PAPLOO.UHI.AC.UK (LOCALHOST [127.0.0.1])
BY PAPLOO.UHI.AC.UK (POSTFIX) WITH ESMTP ID 5F919DC2F2;
SAT, 5 JAN 2008 14:10:05 +0800 (GMT)
MESSAGE-ID: <200801051412.M05ECIAH010327@NAKAMURA.UITS.IUPUI.EDU>
MIME-VERSION: 1.0
CONTENT-TRANSFER-ENCODING: 7BIT
RECEIVED: FROM PROD.COLLAB.UHI.AC.UK ([194.35.219.182])
BY PAPLOO.UHI.AC.UK (JAMES SMTP SERVER 2.1.3) WITH SMTP ID 899
FOR <SOURCE@COLLAB.SAKAIPROJECT.ORG>;
SAT, 5 JAN 2008 14:09:50 +0800 (GMT)
...
-----
THIS AUTOMATIC NOTIFICATION MESSAGE WAS SENT BY SAKAI COLLAB (https://collab.sakaiproject.org/portal) FROM THE SOURCE SITE.
```

Exercise 2:



The screenshot shows a Jupyter Notebook window titled "Lap_trinh_C_Py". The left sidebar displays the Explorer view with a folder named "LAP_TRINH_C_PY" containing several files. The main editor area shows a Python script with the following code:

```
KT_lap_trinh > file2.ipynb > # Yêu cầu người dùng nhập tên file
EXCEPT FileNotFoundError:
    print("Không tìm thấy file. Vui lòng kiểm tra lại tên file.")

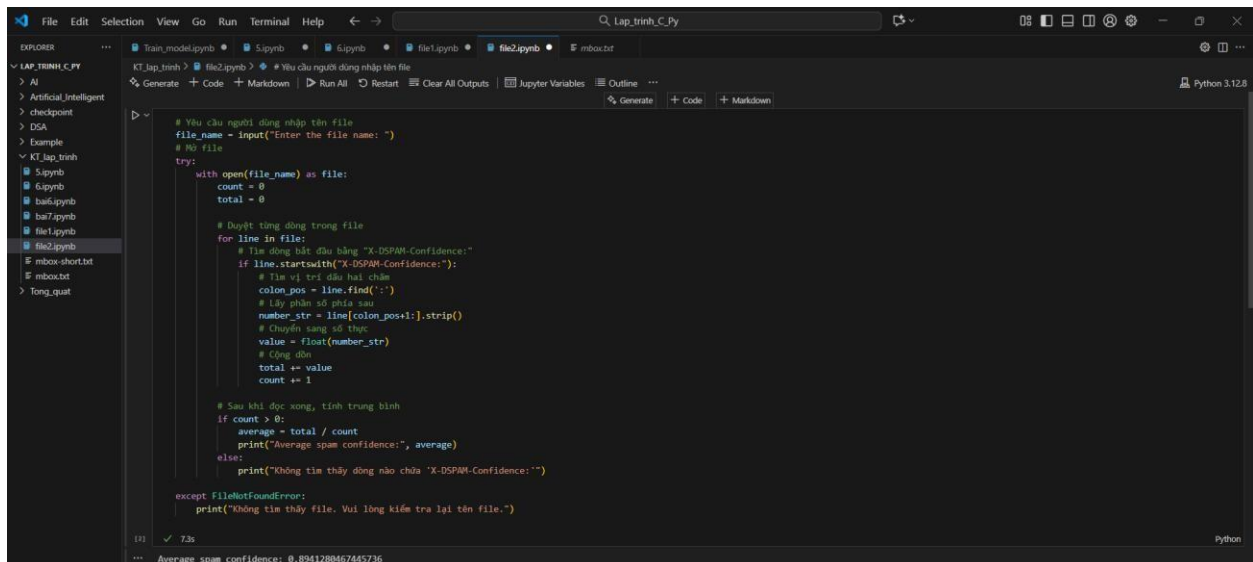
[2] ✓ 73s
... Average spam confidence: 0.8941280467445736

# Yêu cầu người dùng nhập tên file
file_name = input("Enter the file name: ")
# Mở file
try:
    with open(file_name) as file:
        count = 0
        total = 0

        # Duyệt từng dòng trong file
        for line in file:
            # Tìm dòng bắt đầu bằng "X-DSPAM-Confidence:"
            if line.startswith("X-DSPAM-Confidence:"):
                # Tìm vị trí dấu hai chấm
                colon_pos = line.find(':')
                # Lấy phần số phía sau
                number_str = line[colon_pos+1:].strip()
                # Chuyển sang số thực
                value = float(number_str)
                # Cộng dồn
                total += value
                count += 1

        # Sau khi đọc xong, tính trung bình
        if count > 0:
            average = total / count
            print("Average spam confidence:", average)
        else:
            print("Không tìm thấy dòng nào chứa 'X-DSPAM-Confidence:'")
except FileNotFoundError:
    print("Không tìm thấy file. Vui lòng kiểm tra lại tên file.")

[3] ✓ 77s
... Average spam confidence: 0.7507185185185187
```



The screenshot shows a Jupyter Notebook window titled "Lap_trinh_C_Py". The left sidebar displays the Explorer view with a folder named "LAP_TRINH_C_PY" containing several files. The main editor area shows a Python script with the following code:

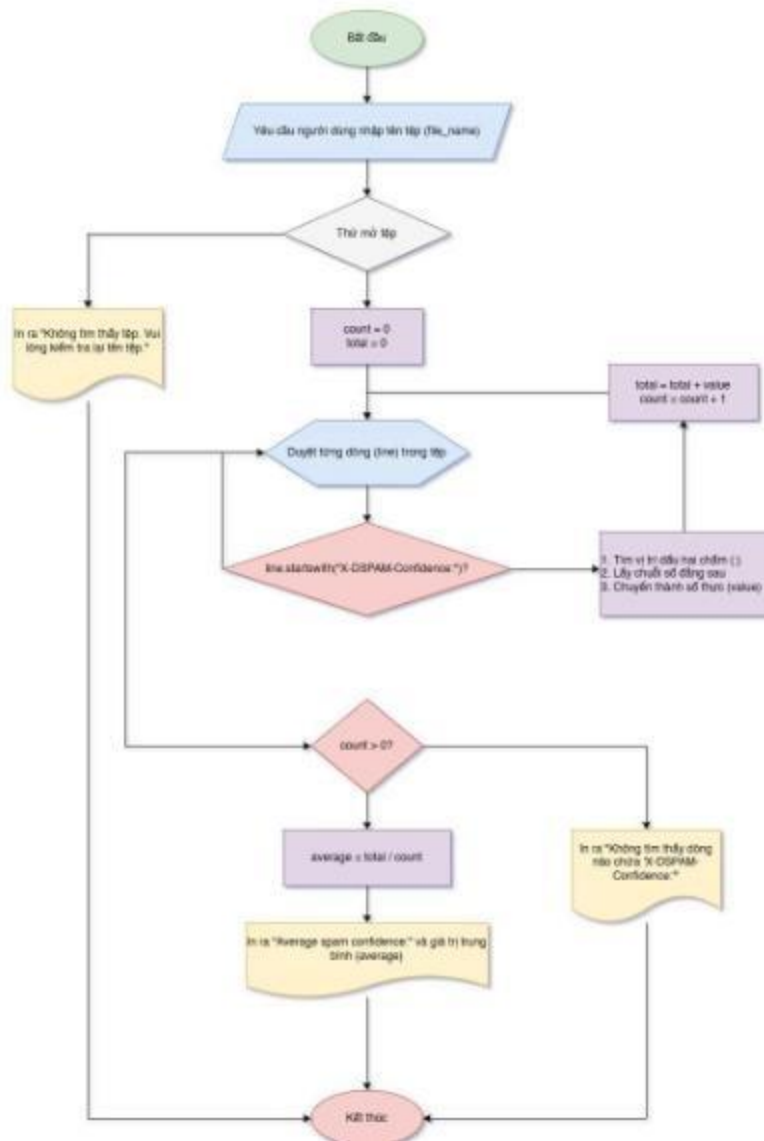
```
KT_lap_trinh > file2.ipynb > # Yêu cầu người dùng nhập tên file
EXCEPT FileNotFoundError:
    print("Không tìm thấy file. Vui lòng kiểm tra lại tên file.")

# Yêu cầu người dùng nhập tên file
file_name = input("Enter the file name: ")
# Mở file
try:
    with open(file_name) as file:
        count = 0
        total = 0

        # Duyệt từng dòng trong file
        for line in file:
            # Tìm dòng bắt đầu bằng "X-DSPAM-Confidence:"
            if line.startswith("X-DSPAM-Confidence:"):
                # Tìm vị trí dấu hai chấm
                colon_pos = line.find(':')
                # Lấy phần số phía sau
                number_str = line[colon_pos+1:].strip()
                # Chuyển sang số thực
                value = float(number_str)
                # Cộng dồn
                total += value
                count += 1

        # Sau khi đọc xong, tính trung bình
        if count > 0:
            average = total / count
            print("Average spam confidence:", average)
        else:
            print("Không tìm thấy dòng nào chứa 'X-DSPAM-Confidence:'")
except FileNotFoundError:
    print("Không tìm thấy file. Vui lòng kiểm tra lại tên file.")

[2] ✓ 73s
... Average spam confidence: 0.8941280467445736
```



Exercise 3:

The screenshot displays a Jupyter Notebook environment with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with folders like 'LAP_TRINH_C_PY' and 'KT_lap_trinh', and various files including '.ipynb' files and '.txt' files. The code editor shows two cells of Python code. The first cell, labeled [1], runs successfully and outputs 'There were 1797 subject lines in mbox.txt'. The second cell, labeled [2], runs and outputs 'File cannot be opened: missing'.

```
KT_lap_trinh > file3.ipynb > # Nhập tên file từ người dùng
Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter V

# Nhập tên file từ người dùng
file_name = input("Enter the file name: ")
# Kiểm tra Easter Egg
if file_name.lower() == "na na boo boo":
    print("NA NA BOO BOO TO YOU - You have been punk'd!")
    quit()
# Nếu không phải Easter Egg thì mở file
try:
    with open(file_name) as file:
        count = 0
        # Đếm số dòng bắt đầu bằng "Subject:"
        for line in file:
            if line.startswith("Subject:"):
                count += 1
        print("There were", count, "subject lines in", file_name)
except FileNotFoundError:
    print("File cannot be opened:", file_name)
```

[1] ✓ 3.5s

... There were 1797 subject lines in mbox.txt

```
KT_lap_trinh > file3.ipynb > # Nhập tên file từ người dùng
Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter V

# Nhập tên file từ người dùng
file_name = input("Enter the file name: ")
# Kiểm tra Easter Egg
if file_name.lower() == "na na boo boo":
    print("NA NA BOO BOO TO YOU - You have been punk'd!")
    quit()
# Nếu không phải Easter Egg thì mở file
try:
    with open(file_name) as file:
        count = 0
        # Đếm số dòng bắt đầu bằng "Subject:"
        for line in file:
            if line.startswith("Subject:"):
                count += 1
        print("There were", count, "subject lines in", file_name)
except FileNotFoundError:
    print("File cannot be opened:", file_name)
```

[2] ✓ 7.2s

... File cannot be opened: missing

EXPLORER

LAP_TRINH_C_PY

> AI

> Artificial_Intelligent

> checkpoint

> DSA

> Example

KT_lap_trinh

5.ipynb

6.ipynb

bai6.ipynb

bai7.ipynb

file1.ipynb

file2.ipynb

file3.ipynb

mbox-short.txt

mbox.txt

Tong_quat

Train_model.ipynb

5.ipynb

6.ipynb

file1.ipynb

file2.ipynb

KT_lap_trinh > file3.ipynb

Nhập tên file từ người dùng

Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Jupyter

Nhập tên file từ người dùng

file_name = input("Enter the file name: ")

Kiểm tra Easter Egg

if file_name.lower() == "na na boo boo":

print("NA NA BOO BOO TO YOU - You have been punk'd!")

quit()

Nếu không phải Easter Egg thì mở file

try:

with open(file_name) as file:

count = 0

Đếm số dòng bắt đầu bằng "Subject:"

for line in file:

if line.startswith("Subject:"):

count += 1

print("There were", count, "subject lines in", file_name)

except FileNotFoundError:

print("File cannot be opened:", file_name)

[3]

... NA NA BOO BOO TO YOU - You have been punk'd!

File cannot be opened: na na boo boo

```

graph TD
    Start([Bắt đầu]) --> Input[/Nhập tên tệp (file_name)/]
    Input --> CheckEgg{file_name.lower() == "na na boo boo"?}
    CheckEgg --> PrintEgg[/In ra "NA NA BOO BOO..." và thoát/]
    CheckEgg --> TryExcept{Thử mở tệp (try...except)}
    TryExcept --> PrintError[/In ra "File cannot be opened"/]
    TryExcept --> CountInit[count = 0]
    CountInit --> LoopStart(( ))
    LoopStart --> ReadLine[/Duyệt từng dòng (line) trong tệp/]
    ReadLine --> CheckSubject{line.startswith("Subject:")?}
    CheckSubject --> CountInc[count += 1]
    CountInc --> LoopStart
    CheckSubject --> PrintCount[/In ra kết quả (count)/]
    PrintCount --> End([Kết thúc])
    
```

The flowchart illustrates the logic of the provided Python code. It begins with a start node, followed by an input node for the file name. A decision diamond checks if the file name (in lowercase) is "na na boo boo". If yes, it prints the message and exits. If no, it enters a try-except block. If the file cannot be opened, it prints an error message. If successful, it initializes a counter to 0 and enters a loop to read each line of the file. A decision diamond checks if each line starts with "Subject:". If yes, it increments the counter. After the loop, it prints the final count and reaches the end node.