Bài tập chương 6 - String

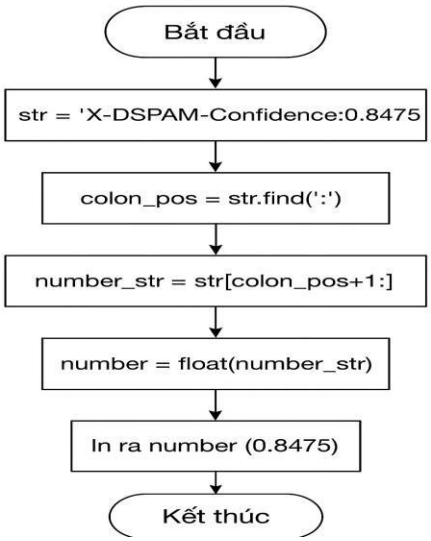
Exercise 5:

```
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
```

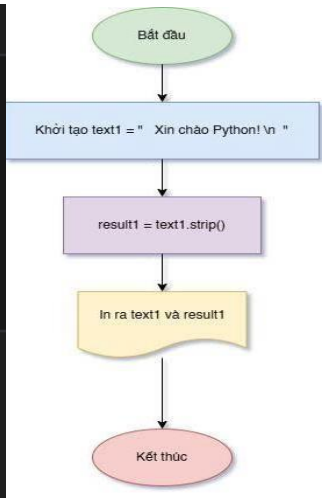


Excersise 6:

```
# 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]

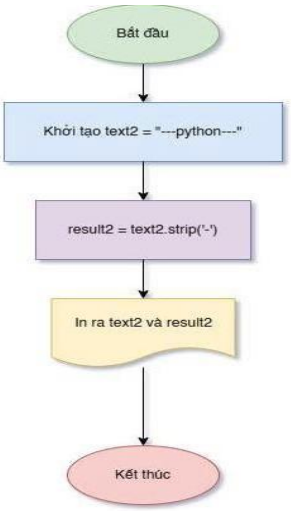
```
Chuỗi ban đầu: '   Xin chào Python!
 '
Chuỗi sau strip(): 'Xin chào Python!'
```



```
# 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]

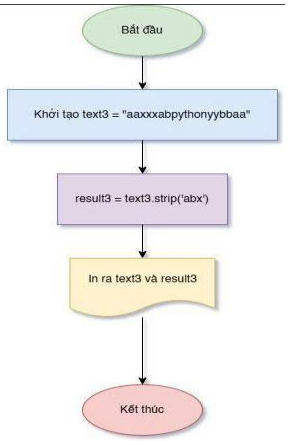
```
Chuỗi ban đầu: '---python---'
Chuỗi sau strip('-'): 'python'
```




```
# 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]

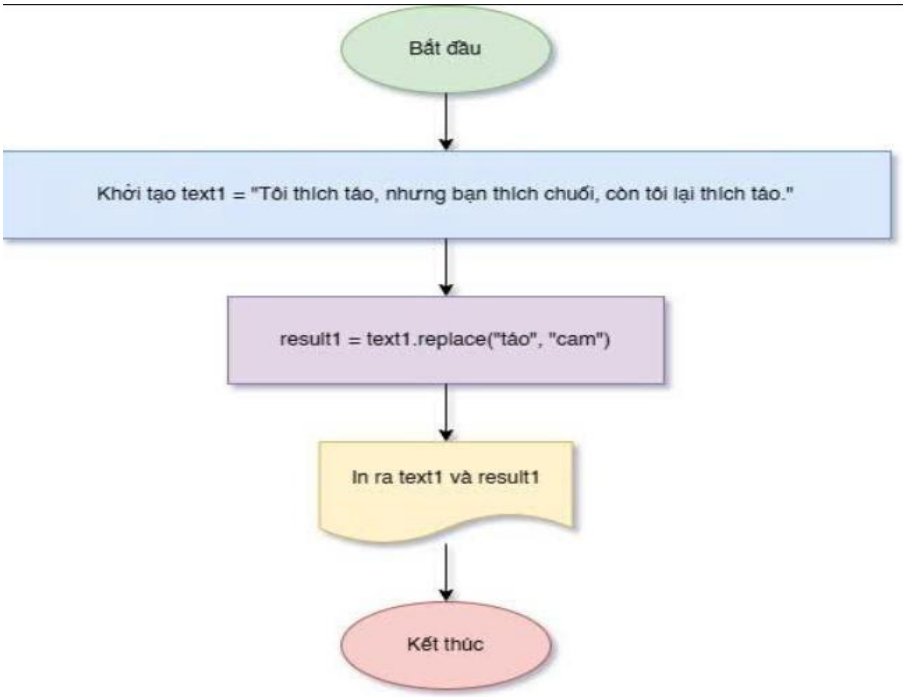
```
Chuỗi ban đầu: 'aaxxxabpythonyybbaa'
Chuỗi sau strip('abx'): 'pythonyy'
```



```python
# 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

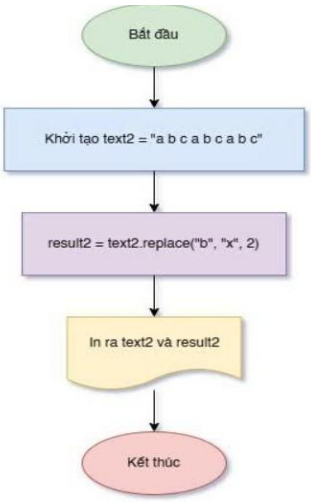
```
Chuỗi ban đầu: Tôi thích táo, nhưng bạn thích chuối, còn tôi lại thích táo.
Chuỗi sau replace(): Tôi thích cam, nhưng bạn thích chuối, còn tôi lại thích cam.
```




```python
# 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}")
```

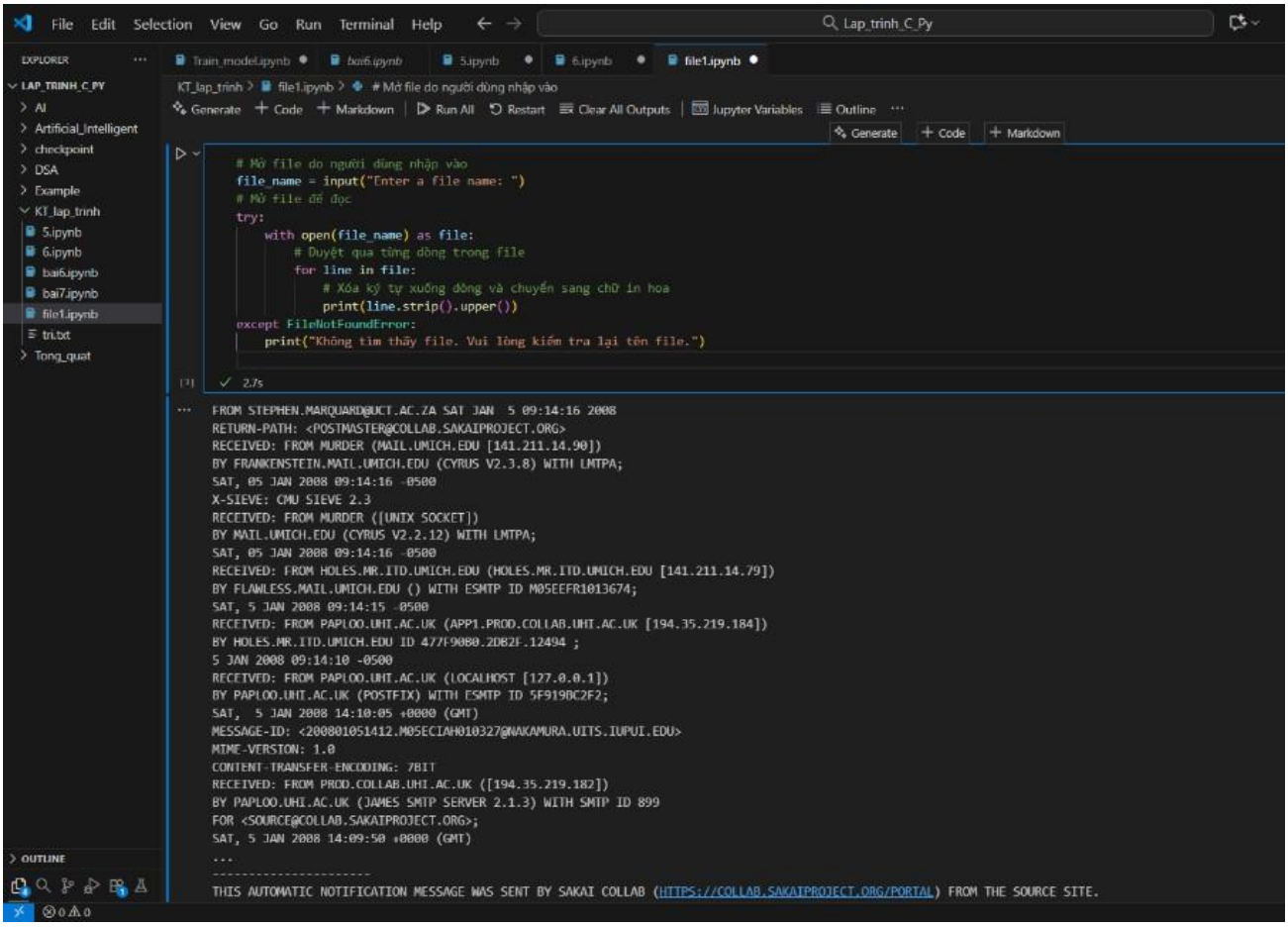
[2] ✓ 0.0s

```
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
```

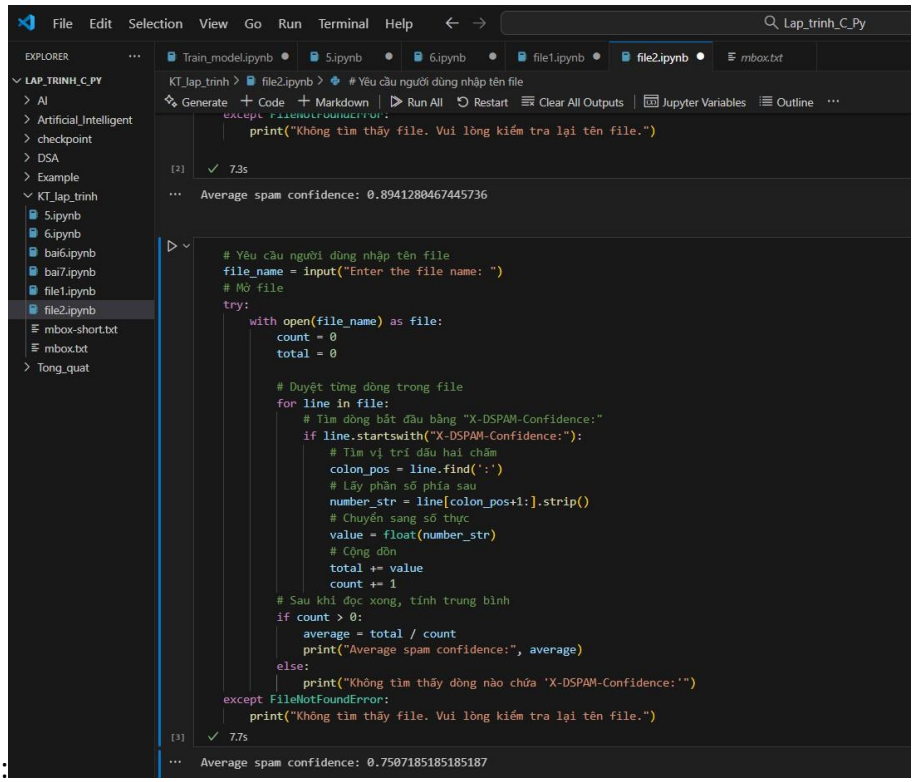


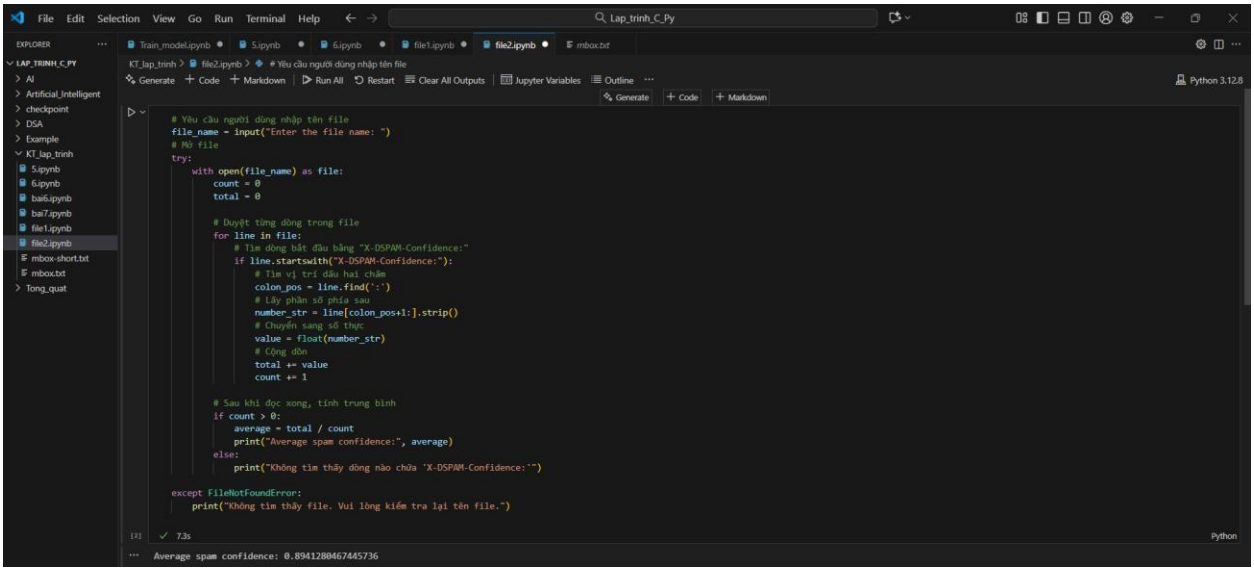
Chương 7 file

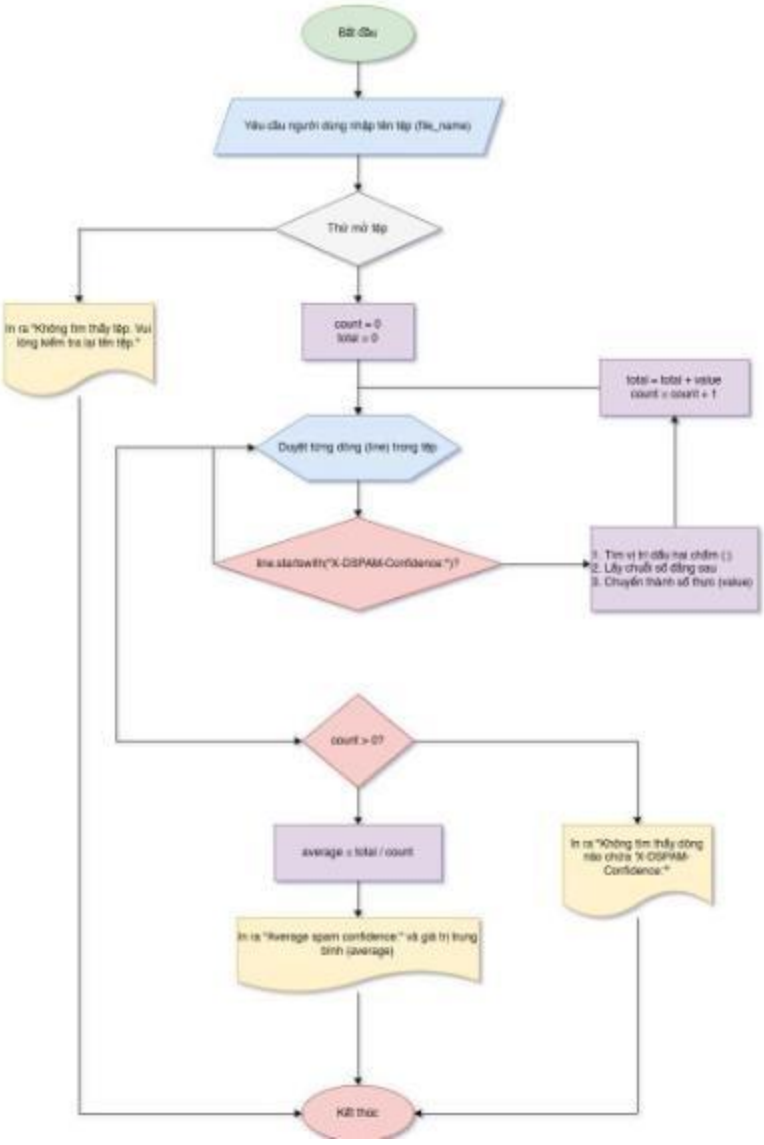
Exercise 1:



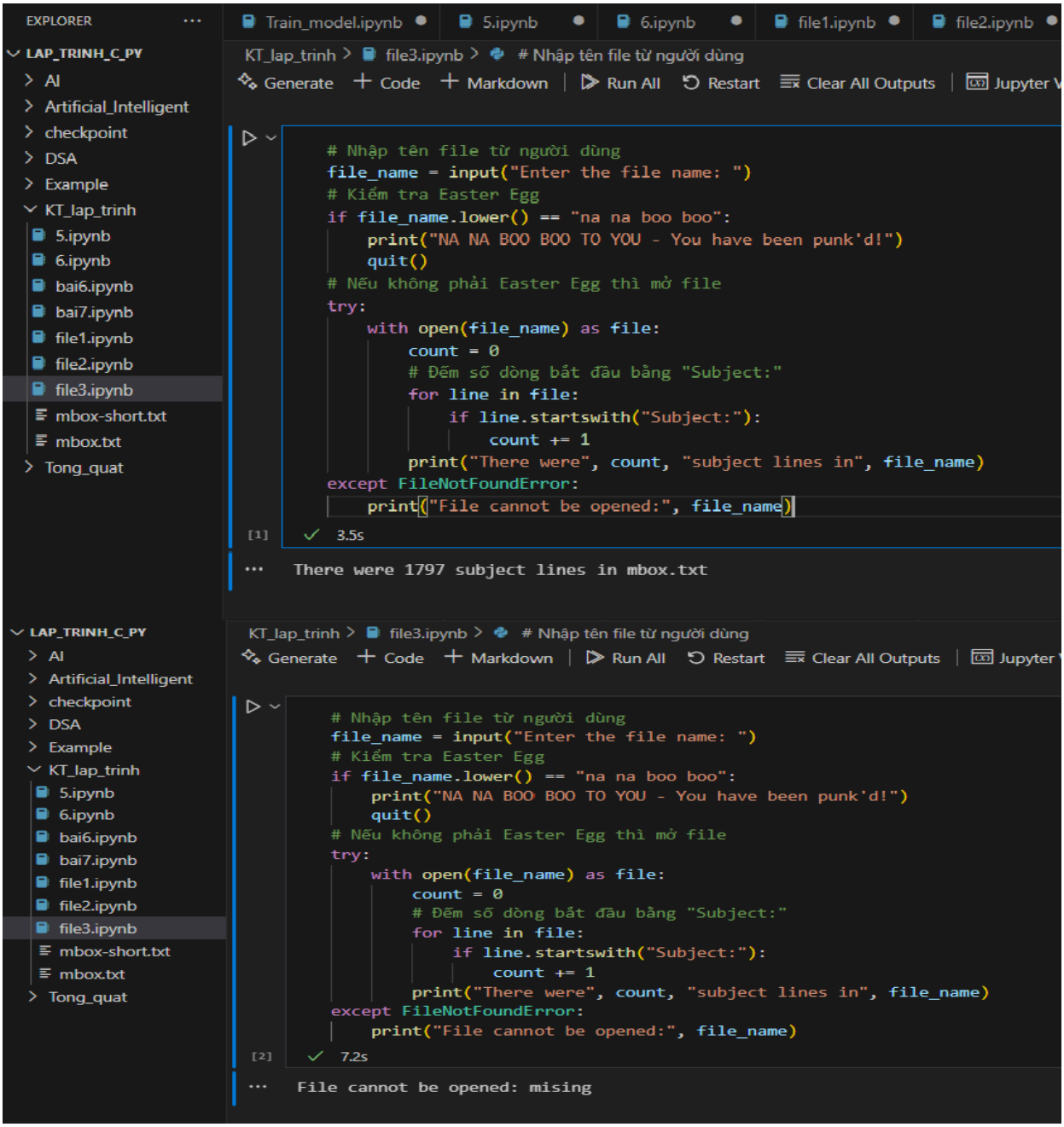
Exercise 2:

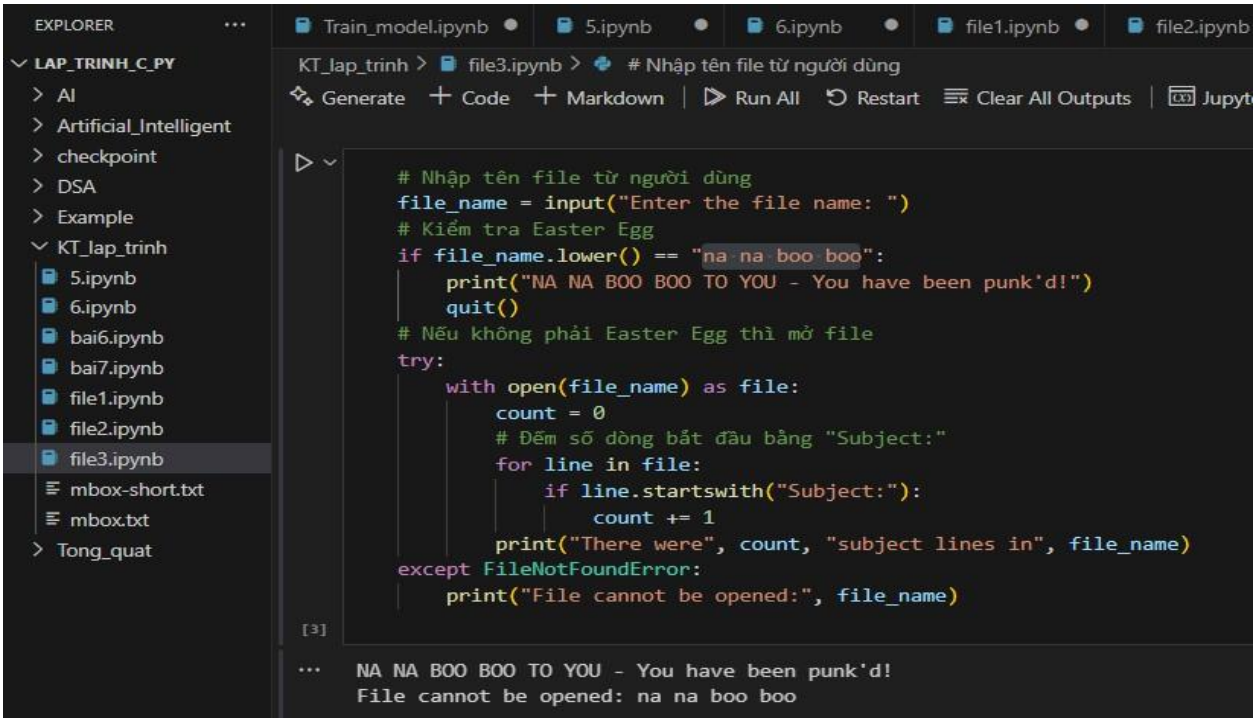

File Edit Selection View Go Run Terminal Help
Lap_trinh_C_Py
EXPLORER
LAP_TRINH_C_PY
AI
Artificial_Intelligent
checkpoint
DSA
Example
KT_lap_trinh
5.ipynb
6.ipynb
bai6.ipynb
bai7.ipynb
file1.ipynb
file2.ipynb
mbox-short.txt
mbox.txt
Tong_quat
Train_model.ipynb
5.ipynb
6.ipynb
file1.ipynb
file2.ipynb
mbox.txt
KT_lap_trinh > file2.ipynb > # Yêu cầu người dùng nhập tên file
Generate Code Markdown Run All Restart Clear All Outputs Jupyter Variables Outline
print("Không tìm thấy file. Vui lòng kiểm tra lại tên file.")
[2] 7.3s
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] 7.7s
Average spam confidence: 0.7507185185185187

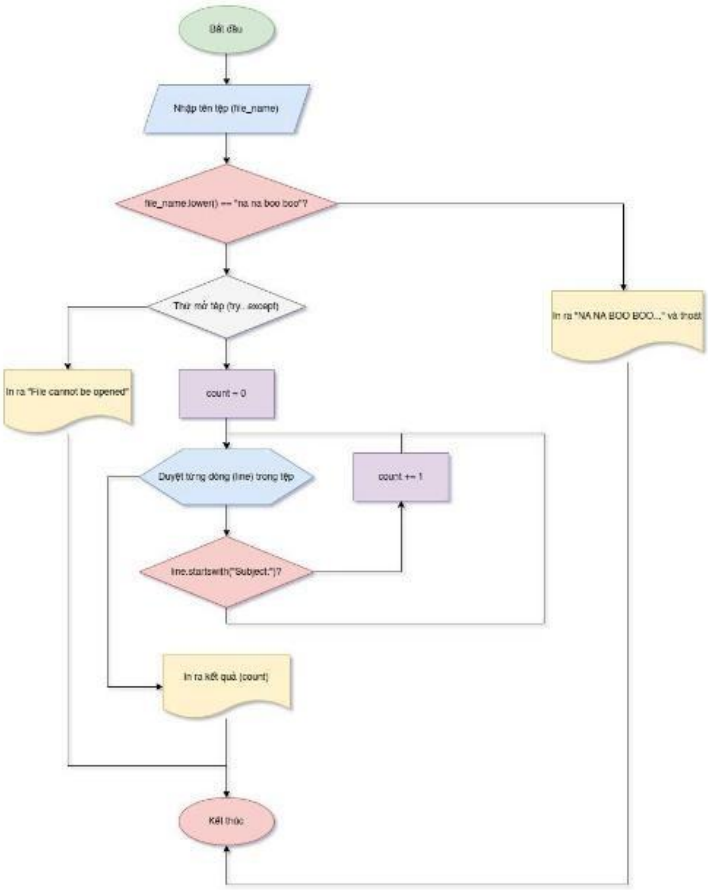

File Edit Selection View Go Run Terminal Help
Lap_trinh_C_Py
Python 3.12.8
Generate Code Markdown
# Yêu cầu người dùng nhập tên file
file_name = input("Enter the file name: ")
# Mở file
try:
with open(file_name) as file:
count = 0
total = 0
# Duyệt từng dòng trong file
for line in file:
# Tìm dòng bắt 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] 7.3s
Python
Average spam confidence: 0.8941280467445736


Bắt đầu
Yêu cầu người dùng nhập tên tệp (file_name)
Thử mở tệp
In ra "Không tìm thấy tệp. Vui lòng kiểm tra lại tên tệp."
count = 0
total = 0
total = total + value
count = count + 1
Duyệt từng dòng (line) trong tệp
line.startswith("X-DSPAM-Confidence:")?
1. Tìm vị trí dấu hai chấm (:)
2. Lấy chuỗi sở đằng sau
3. Chuyển thành số thực (value)
count > 0?
average = total / count
In ra "Không tìm thấy dòng nào chứa 'X-DSPAM-Confidence:'"
In ra "Average spam confidence:" và giá trị trung bình (average)
Kết thúc

Exercise 3:


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)
[1]
3.5s
There were 1797 subject lines in mbox.txt
[2]
7.2s
File cannot be opened: mising


EXPLORER
Train_model.ipynb
5.ipynb
6.ipynb
file1.ipynb
file2.ipynb
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
KT_lap_trinh > file3.ipynb > # Nhập tên file từ người dùng
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
# 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


Bắt đầu
Nhập tên tệp (file_name)
file_name.lower() == "na na boo boo"?
In ra "NA NA BOO BOO..." và thoát
Thử mở tệp (try...except)
In ra "File cannot be opened"
count = 0
Duyệt từng dòng (line) trong tệp
count += 1
line.startswith("Subject:")?
In ra kết quả (count)
Kết thúc