

## Bài 1. Đếm dòng code

Một cách để đo độ phức tạp của chương trình là đếm số dòng mã (LOC), không bao gồm các dòng trống và chú thích. Ví dụ, một chương trình như

```
# Say hello
```

```
name = input("What's your name? ")
print(f'hello, {name}')
```

chỉ có hai dòng mã, không phải bốn dòng, vì dòng đầu tiên của nó là một nhận xét và dòng thứ hai của nó trống (tức là chỉ có khoảng trắng). Con số đó không nhiều nên rất có thể chương trình không phức tạp đến thế. Tất nhiên, chỉ vì một chương trình (hoặc thậm chí là hàm) có nhiều dòng mã hơn chương trình khác không nhất thiết có nghĩa là nó phức tạp hơn. Ví dụ, một chức năng như

```
def is_even(n):
    if n % 2 == 0:
        return True
    else:
        return False
```

thực sự không phức tạp gấp đôi so với một hàm như

```
def is_even(n):
    return n % 2 == 0
```

mặc dù cái trước có số dòng mã nhiều hơn (hơn) gấp đôi. Trên thực tế, cái trước có thể được cho là đơn giản hơn nếu nó dễ đọc hơn!

Mặc dù vậy, trong một tệp có tên `lines.py`, hãy triển khai một chương trình yêu cầu chính xác một đối số dòng lệnh, tên (hoặc đường dẫn) của tệp Python và xuất ra số dòng mã trong tệp đó, không bao gồm các nhận xét và dòng trống. Nếu người dùng không chỉ định chính xác một đối số dòng lệnh hoặc nếu tên tệp được chỉ định không kết thúc bằng `.py`, hoặc nếu tệp được chỉ định không tồn tại thì chương trình sẽ thoát qua `sys.exit`.

Giả sử rằng bất kỳ dòng nào bắt đầu bằng #, tùy ý có khoảng trắng trước, đều là một nhận xét. ([Chuỗi tài liệu](#) không được coi là một nhận xét.) Giả sử rằng bất kỳ dòng nào chỉ chứa khoảng trắng đều là dòng trống.

### Gợi ý

- Hãy nhớ lại rằng a str đi kèm với khá nhiều phương thức, theo [docs.python.org/3/library/stdtypes.html#string-methods](https://docs.python.org/3/library/stdtypes.html#string-methods), bao gồm `lstrip`, `startswith`, và `endswith`.
- Lưu ý rằng open có thể raise/ném ra một ngoại lệ `FileNotFoundError`, theo [docs.python.org/3/library/Exceptions.html#FileNotFoundError](https://docs.python.org/3/library/Exceptions.html#FileNotFoundError).
- Bạn có thể thấy hữu ích khi thử nghiệm chương trình của mình trên một số [mã nguồn của Tuần 6](#) cũng như trên các chương trình của riêng bạn.

### Cách test:

- Chạy chương trình của bạn với `python lines.py`. Chương trình của bạn sẽ thoát `sys.exit` và đưa ra thông báo lỗi: `Too few command-line arguments`
- Tạo hai chương trình python `hello.py` và `goodbye.py`. Chạy `python lines.py hello.py goodbye.py`. Chương trình của bạn sẽ thoát `sys.exit` và đưa ra thông báo lỗi: `Too many command-line arguments`
- Tạo một tệp văn bản có tên `invalid_extension.txt`. Chạy chương trình của bạn với `python lines.py invalid_extension.txt`. Chương trình của bạn sẽ thoát `sys.exit` và đưa ra thông báo lỗi: `Not a Python file`
- Chạy chương trình của bạn với `python lines.py non_existent_file.py`. Giả sử `non_existent_file.py` không tồn tại, chương trình của bạn sẽ thoát `sys.exit` và đưa ra thông báo lỗi: `File does not exist`
- Tạo các chương trình python bổ sung có độ phức tạp khác nhau: tạo một số chương trình có nhận xét, một số chuỗi tài liệu và một số khoảng trắng. Đối với mỗi tệp này, tên của tệp sẽ chạy `python lines.py FILENAME`. Chương trình sẽ xuất ra số dòng, không bao gồm nhận xét và khoảng trắng, có trong tệp `FILENAME` đã cho.

## Bài 2. pizza

Có lẽ địa điểm bán pizza phổ biến nhất ở [Quảng trường Harvard](#) là [Pinocchio's Pizza & Subs](#), hay còn gọi là Noch's, được biết đến với [món pizza Sicilia](#), là "một loại bánh pizza đế sâu hoặc vỏ dày".

Sinh viên có xu hướng mua pizza theo từng miếng, nhưng Pinocchio's cũng có cả pizza trong [thực đơn](#) của mình, theo tệp CSV này của pizza Sicilia, [sicilian.csv](#), bên dưới:

```
Sicilian Pizza,Small,Large
Cheese,$25.50,$39.95
1 item,$27.50,$41.95
2 items,$29.50,$43.95
3 items,$31.50,$45.95
Special,$33.50,$47.95
```

Xem [Regular.csv](#) để biết tệp CSV của các loại pizza thông thường.

Tất nhiên, tệp CSV không phải là định dạng thân thiện với khách hàng nhất. Đẹp hơn có thể là một bảng, được định dạng là [ASCII art](#), như bảng này:

```
+-----+-----+-----+
| Sicilian Pizza | Small | Large |
+=====+=====+=====+
| Cheese        | $25.50 | $39.95 |
+-----+-----+-----+
| 1 item        | $27.50 | $41.95 |
+-----+-----+-----+
| 2 items       | $29.50 | $43.95 |
+-----+-----+-----+
| 3 items       | $31.50 | $45.95 |
+-----+-----+-----+
| Special       | $33.50 | $47.95 |
+-----+-----+-----+
```

Trong tệp có tên `pizza.py`, hãy triển khai chương trình yêu cầu chính xác một đối số dòng lệnh, tên (hoặc đường dẫn) của tệp CSV ở định dạng của Pinocchio và xuất ra một bảng được định dạng dưới dạng nghệ thuật ASCII bằng cách sử dụng , một gói trên PyPI `tabulate` tại [pypi.org/project/tabulate](http://pypi.org/project/tabulate) . Định dạng bảng bằng định dạng grid của thư viện. Nếu người dùng không chỉ định chính xác một đối số dòng lệnh hoặc nếu tên tệp được chỉ định không kết thúc bằng `.csv`, hoặc nếu tệp được chỉ định không tồn tại thì chương trình sẽ thoát qua `sys.exit`.

## Gợi ý

- Hãy nhớ lại rằng mô-đun `csv` này đi kèm với khá nhiều phương thức, theo [docs.python.org/3/library/csv.html](https://docs.python.org/3/library/csv.html) , trong đó có `reader`, theo [docs.python.org/3/library/csv.html#csv.reader](https://docs.python.org/3/library/csv.html#csv.reader) và `DictReader`, theo [docs.python.org/3/library/csv.html#csv.DictReader](https://docs.python.org/3/library/csv.html#csv.DictReader) .
- Lưu ý rằng `open` có thể ném một ngoại lệ `FileNotFoundError`, theo [docs.python.org/3/library/Exceptions.html#FileNotFoundError](https://docs.python.org/3/library/Exceptions.html#FileNotFoundError) .
- Lưu ý rằng gói `tabulate` chỉ có một hàm, theo <http://pypi.org/project/tabulate> . Bạn có thể cài đặt gói với:

```
pip install tabulate
```

## Cách test:

- Chạy chương trình của bạn với `python pizza.py`. Chương trình của bạn sẽ thoát bằng cách sử dụng `sys.exit` và đưa ra thông báo lỗi: `Too few command-line arguments`
- Hãy nhớ tải xuống [Regular.csv](#) và [sicilian.csv](#) , đặt chúng vào cùng thư mục với `pizza.py`. Chạy chương trình của bạn với `python pizza.py regular.csv sicilian.csv`. Chương trình của bạn sẽ xuất ra: `Too many command-line arguments`
- Chạy chương trình của bạn với `python pizza.py invalid_file.csv`. Giả sử `invalid_file.csv` không tồn tại, chương trình của bạn sẽ thoát bằng cách sử dụng `sys.exit` và đưa ra thông báo lỗi: `File does not exist`
- Tạo một tệp có tên `sicilian.txt`. Chạy chương trình của bạn với `python pizza.py sicilian.txt`. Chương trình của bạn sẽ thoát bằng cách sử dụng `sys.exit` và đưa ra thông báo lỗi: `Not a CSV file`
- Chạy chương trình của bạn với `python pizza.py regular.csv`. Giả sử bạn đã tải xuống [Regular.csv](#) , chương trình của bạn sẽ in một bảng có dạng như bên dưới:

```
+-----+-----+-----+
| Regular Pizza| Small | Large |
+-----+-----+-----+
| Cheese      | $13.50 | $18.95 |
+-----+-----+-----+
| 1 topping   | $14.75 | $20.95 |
+-----+-----+-----+
| 2 toppings  | $15.95 | $22.95 |
+-----+-----+-----+
| 3 toppings  | $16.95 | $24.95 |
+-----+-----+-----+
| Special     | $18.50 | $26.95 |
+-----+-----+-----+
```

### Bài 3. Làm sạch dữ liệu

Dữ liệu cũng thường cần phải được "làm sạch", bằng cách định dạng lại nó, để các giá trị ở định dạng nhất quán, nếu không muốn nói là thuận tiện hơn. Ví dụ: hãy xem xét tệp CSV này của sinh viên [before.csv](#) bên dưới:

```
name,house
"Abbott, Hannah",Hufflepuff
"Bell, Katie",Gryffindor
"Bones, Susan",Hufflepuff
"Boot, Terry",Ravenclaw
"Brown, Lavender",Gryffindor
"Bulstrode, Millicent",Slytherin
"Chang, Cho",Ravenclaw
"Clearwater, Penelope",Ravenclaw
"Crabbe, Vincent",Slytherin
"Creevey, Colin",Gryffindor
"Creevey, Dennis",Gryffindor
"Diggory, Cedric",Hufflepuff
"Edgecombe, Marietta",Ravenclaw
"Finch-Fletchley, Justin",Hufflepuff
"Finnigan, Seamus",Gryffindor
"Goldstein, Anthony",Ravenclaw
"Goyle, Gregory",Slytherin
"Granger, Hermione",Gryffindor
"Johnson, Angelina",Gryffindor
"Jordan, Lee",Gryffindor
"Longbottom, Neville",Gryffindor
"Lovegood, Luna",Ravenclaw
"Lupin, Remus",Gryffindor
"Malfoy, Draco",Slytherin
"Malfoy, Scorpius",Slytherin
"Macmillan, Ernie",Hufflepuff
"McGonagall, Minerva",Gryffindor
"Midgen, Eloise",Gryffindor
"McLaggen, Cormac",Gryffindor
"Montague, Graham",Slytherin
"Nott, Theodore",Slytherin
"Parkinson, Pansy",Slytherin
"Patil, Padma",Gryffindor
"Patil, Parvati",Gryffindor
"Potter, Harry",Gryffindor
"Riddle, Tom",Slytherin
"Robins, Demelza",Gryffindor
"Scamander, Newt",Hufflepuff
"Slughorn, Horace",Slytherin
"Smith, Zacharias",Hufflepuff
"Snape, Severus",Slytherin
"Spinnet, Alicia",Gryffindor
"Sprout, Pomona",Hufflepuff
"Thomas, Dean",Gryffindor
"Vane, Romilda",Gryffindor
"Warren, Myrtle",Ravenclaw
```

"Weasley, Fred",Gryffindor  
"Weasley, George",Gryffindor  
"Weasley, Ginny",Gryffindor  
"Weasley, Percy",Gryffindor  
"Weasley, Ron",Gryffindor  
"Wood, Oliver",Gryffindor  
"Zabini, Blaise",Slytherin

Nguồn: [en.wikipedia.org/wiki/List of Harry Potter characters](https://en.wikipedia.org/wiki/List_of_Harry_Potter_characters)

Mặc dù mỗi “hàng” trong tệp có ba giá trị (họ, tên và nhà/house), hai giá trị đầu tiên được kết hợp thành một “cột” (name), được đặt trong cặp dấu nháy kép, với **họ/last name** và **tên/first name** được phân tách bằng dấu phẩy và dấu cách ‘, ’. Sẽ không lý tưởng nếu **Hogwarts** muốn gửi một **lá thư mẫu** cho mỗi học sinh, như thông qua **trộn thư**, vì sẽ rất lạ nếu bắt đầu một lá thư bằng:

Harry thân mến, Harry,  
Thay vì với, ví dụ:

Harry thân mến,  
Trong một tệp có tên `scourgify.py`, hãy triển khai một chương trình:

- Yêu cầu người dùng cung cấp hai đối số dòng lệnh:
  - tên của tệp CSV hiện có để đọc làm đầu vào, có các cột được giả định là, theo thứ tự, **name** và **house**, và
  - tên của một CSV mới để ghi làm đầu ra, các cột của nó phải theo thứ tự, **first**, **last** và **house**.
- Chuyển đổi đầu vào đó thành đầu ra đó, tách từng đầu vào name thành first tên và last họ. Giả sử mỗi học sinh sẽ có cả họ và tên.

Nếu người dùng không cung cấp chính xác hai đối số dòng lệnh hoặc nếu không thể đọc được đối số dòng lệnh đầu tiên, chương trình sẽ thoát ra kèm theo `sys.exit` thông báo lỗi.

### Gợi ý

- Lưu ý rằng csv mô-đun đi kèm với khá nhiều phương thức, theo [docs.python.org/3/library/csv.html](https://docs.python.org/3/library/csv.html), trong đó có DictReader, theo [docs.python.org/3/library/csv.html#csv.DictReader](https://docs.python.org/3/library/csv.html#csv.DictReader) và DictWriter, theo [docs.python.org/3/library/csv.html#csv.DictWriter](https://docs.python.org/3/library/csv.html#csv.DictWriter).
- Lưu ý rằng bạn có thể yêu cầu a DictWriter ghi nó fieldnames vào một tệp writeheader mà không cần đối số, theo [docs.python.org/3/library/csv.html#csv.DictWriter.writeheader](https://docs.python.org/3/library/csv.html#csv.DictWriter.writeheader).

### Cách test:

- Chạy chương trình của bạn với `python scourgify.py`. Chương trình của bạn sẽ thoát bằng cách sử dụng `sys.exit` và đưa ra thông báo lỗi: **Too few command-line arguments**
- Tạo các tập tin trống `1.csv`, `2.csv` và `3.csv`. Chạy chương trình của bạn với `python scourgify.py 1.csv 2.csv 3.csv`. Chương trình của bạn sẽ xuất ra: **Too many command-line arguments**
- Chạy chương trình của bạn với `python scourgify.py invalid_file.csv output.csv`. Giả sử `invalid_file.csv` không tồn tại, chương trình của bạn sẽ thoát bằng cách sử dụng `sys.exit` và đưa ra thông báo lỗi: **Could not read invalid\_file.csv**

- Chạy chương trình của bạn với `python scourgify.py before.csv after.csv`. Giả sử `before.csv` tồn tại, chương trình của bạn sẽ tạo một tệp mới, `after.csv`, có các cột theo thứ tự, `first`, `last`, và `house`. Dưới đây là một số dòng đầu tiên của `after.csv`:

`first,last,house`

`Hannah,Abbott,Hufflepuff`

`Katie,Bell,Gryffindor`

`Susan,Bones,Hufflepuff`

`Terry,Boot,Ravenclaw`