Bài 1. Kiểm thử twttr.py

Trong một tệp có tên twttr.py, cài đặt lại Bài 3 twitter trong Bài tập Week 2, hãy cơ cấu lại mã của bạn theo cách bên dưới, trong đó hàm shorten mong đợi a str làm đầu vào và trả về xâu tương tự str nhưng với tất cả các nguyên âm (A, E, I, O và U) bị bỏ qua, cho dù được nhập bằng chữ hoa hoặc chữ thường.

```
def main():
    ...


def shorten(word):
    ...


if __name__ == "__main__":
    main()
```

Sau đó, trong tệp có tên test_twttr.py, hãy triển khai **một hoặc nhiều** hàm để kiểm tra tổng thể việc triển khai hàm shorten của bạn một cách kỹ lưỡng, mỗi tên phải bắt đầu bằng test_ để bạn có thể thực hiện kiểm tra của mình bằng:

pytest test_twttr.py

Gợi ý

- Hãy chắc chắn bao gồm
- **import** twttr hoặc **from** twttr **import** shorten

ở đầu file test_twttr.py để bạn có thể gọi shorten trong bài kiểm tra của mình.

- Hãy cẩn thận return, không in xâu trong hàm shorten. Chỉ main gọi print.

Cách test:

Để kiểm tra các bài kiểm tra của bạn, hãy chạy pytest test_twttr.py. Hãy chắc chắn rằng bạn có một bản sao của twttr.py tập tin trong cùng một thư mục. Cố gắng sử dụng các phiên bản đúng và không chính xác của twttr.py để xác định mức độ kiểm tra của bạn phát hiện ra lỗi:

- Đảm bảo bạn có phiên bản chính xác của twttr.py. Chạy thử nghiệm của bạn bằng cách thực thi pytest test_twttr.py. pytest sẽ cho thấy rằng tất cả các bài kiểm tra của bạn đã vượt qua.
- Sửa đổi phiên bản chính xác twttr.py theo cách có thể tạo ra lỗi. Ví dụ: chương trình của bạn có thể nhầm lẫn khi chỉ bỏ sót các nguyên âm viết thường! Chạy thử nghiệm của bạn bằng cách thực thi pytest test_twttr.py. pytest sẽ cho thấy rằng ít nhất một trong các thử nghiệm của bạn đã thất bại.

Bài 2. Kiểm thử bank.py

Trong một tệp có tên bank.py, triển khai lại Bài 2 Bank trong Bài tập Week 1, hãy tổ chức lại mã của bạn theo bên dưới, trong đó hàm value yêu cầu a str làm đầu vào và trả về 0 nếu str bắt đầu bằng "hello", 20 nếu mã đó str bắt đầu bằng "h" (nhưng không phải "hello" "), hoặc 100 với các trường hợp khác; xử lý str với các trường hợp không phân biệt chữ hoa chữ thường. Bạn có thể giả sử rằng chuỗi được truyền vào hàm value sẽ không chứa bất kỳ khoảng trắng nào ở đầu. Chỉ nên gọi print trong main.

```
def main():
    ...


def value(greeting):
    ...


if __name__ == "__main__":
    main()
```

Sau đó, trong một tệp có tên test_bank.py, hãy triển khai **ba hàm trở lên** để kiểm tra tổng thể việc triển khai value của bạn một cách kỹ lưỡng, mỗi hàm phải bắt đầu bằng tên của chúng test_ để bạn có thể thực hiện kiểm tra của mình bằng:

```
pytest test_bank.py
```

Gợi ý

- Hãy chắc chắn bao gồm
- `import bank`

  hoặc

  `from bank import value`

  ở đầu test_bank.pyđể bạn có thể gọi valuetrong bài kiểm tra của mình.

- Hãy cẩn thận return, không print một số nguyên trong value. Chỉ nên gọi print trong main.

Cách test:

Để kiểm tra các bài kiểm tra của bạn, hãy chạy pytest test_bank.py. Hãy chắc chắn rằng bạn có một bản sao của tập tin bank.py trong cùng một thư mục. Cố gắng sử dụng các phiên bản đúng và không chính xác của bank.py để xác định mức độ kiểm tra của bạn phát hiện ra lỗi:

- Đảm bảo bạn có phiên bản chính xác của bank.py. Chạy thử nghiệm của bạn bằng cách thực thi pytest test_bank.py. pytest sẽ cho thấy rằng tất cả các bài kiểm tra của bạn đã vượt qua.
- Sửa đổi phiên bản chính xác của bank.py, thay đổi các giá trị được cung cấp cho mỗi lời chào. Ví dụ: chương trình của bạn có thể cung cấp nhầm $100 cho khách hàng được chào bằng "Hello" và $0 cho khách hàng được chào bằng "What's up"! Bây giờ, hãy chạy thử nghiệm bằng cách thực thi pytest test_bank.py. pytest sẽ cho thấy rằng ít nhất một trong các thử nghiệm của bạn đã thất bại.

Bài 3. Tương tự như hai bài trên, viết chương trình để kiểm thử cho Bài 4 – Biển số trong Bài tập Week 2

Bài 4. Viết chương trình để kiểm thử cho Bài 1 – fuel trong Bài tập Week 3.