Bài 1. Deep thought

Hãy tạo tệp có tên **deep.py**, triển khai chương trình nhắc người dùng trả lời câu hỏi "`What is the answer to the Great Question of Life, the Universe, and Everything?`", xuất ra **Yes** nếu người dùng nhập **42** hoặc (không phân biệt chữ hoa chữ thường) **forty-two** hoặc **forty two**. Nếu không thì xuất ra **No**.

Bài 2. Bank

Hãy tạo tệp có tên **bank.py**, hãy triển khai một chương trình nhắc người dùng chào hỏi. Nếu lời chào bắt đầu bằng "`hello`", xuất ra `$0`. Nếu lời chào bắt đầu bằng chữ "`h`" (chứ không phải "`hello`"), hãy xuất ra `$20`. Ngược lại, xuất ra `$100`. Bỏ qua mọi khoảng trắng ở đầu trong lời chào của người dùng và xử lý lời chào của người dùng không phân biệt chữ hoa chữ thường.

Bài 3. File extension

Mặc dù Windows và macOS đôi khi ẩn phần mở rộng của tệp nhưng hầu hết các tệp đều có phần mở rộng tệp , hậu tố bắt đầu bằng dấu chấm (.) ở cuối tên của chúng. Ví dụ: tên tệp cho ảnh GIF kết thúc bằng .gif, và tên tệp cho ảnh JPEG kết thúc bằng .jpg hoặc .jpeg. Khi bạn bấm đúp vào một tệp để mở tệp đó, máy tính của bạn sẽ sử dụng phần mở rộng tệp của tệp đó để xác định chương trình nào sẽ khởi chạy.

Ngược lại, các trình duyệt web dựa vào các loại phương tiện , trước đây gọi là loại MIME, để xác định cách hiển thị các tệp trực tuyến trên web. Khi bạn tải xuống tệp từ máy chủ web, máy chủ đó sẽ gửi tiêu đề HTTP , cùng với chính tệp đó, cho biết loại phương tiện của tệp. Ví dụ: loại phương tiện cho ảnh GIF là image/gif, và loại phương tiện cho JPEG là image/jpeg. Để xác định loại phương tiện cho một tệp, máy chủ web thường xem xét phần mở rộng của tệp, ánh xạ phần mở rộng này với phần mở rộng khác.

Xem https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types để biết các loại phổ biến.

Hãy viết tệp có tên extensions.py, hãy triển khai chương trình nhắc người dùng nhập tên tệp rồi xuất ra loại phương tiện của tệp đó nếu tên tệp kết thúc, không phân biệt chữ hoa chữ thường, với bất kỳ hậu tố nào sau đây:

- .gif
- .jpg

- .jpeg
- .png
- .pdf
- .txt
- .zip

Nếu tên của tệp kết thúc bằng một số hậu tố khác hoặc không có hậu tố nào cả, hãy xuất ra application/octet-stream, đây là một mặc định phổ biến.

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 .

Kiểm tra / testing:

- Chạy chương trình và gõ happy.jpg và nhấn Enter -> chương trình in ra: image/jpeg
- Chạy chương trình và gõ document.pdf và nhấn Enter -> chương trình in ra: application/pdf

Bài 4. Interpreter

Python đã hỗ trợ toán học, nhờ đó *bạn* có thể viết mã để cộng, trừ, nhân hoặc chia các giá trị và thậm chí cả các biến. Nhưng hãy viết một chương trình cho phép *người dùng* làm toán mà không cần biết Python.

Hãy viết tệp có tên interpreter.py, hãy triển khai một chương trình nhắc người dùng nhập biểu thức số học, sau đó tính toán và xuất kết quả dưới dạng giá trị dấu phẩy động được định dạng đến một chữ số thập phân. Giả sử rằng đầu vào của người dùng sẽ được định dạng là x y z, với một khoảng trắng ở giữa x và y một khoảng trắng ở giữa y và z, trong đó:

- x là một số nguyên
- y là +, -, * hoặc /
- z là một số nguyên

Chẳng hạn, nếu người dùng nhập 1 + 1, chương trình của bạn sẽ xuất ra 2.0. Giả sử nếu y có / thì z không 0.

Lưu ý rằng, giống như python bản thân nó là một trình thông dịch cho Python, thì bạn cũng sẽ interpreter.py là một trình thông dịch cho môn Toán!

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 , bao gồm split, phân tách a str thành một chuỗi các giá trị, tất cả đều có thể được gán cho các biến cùng một lúc . Ví dụ: nếu expression là 1 + 1 thì

x, y, z = expression.split(" ")

sẽ gán 1 cho x, + cho y, và 1 cho z.

Bài 5. Meal time

Giả sử bạn đang ở một đất nước có phong tục ăn sáng từ 7 giờ đến 8 giờ, bữa trưa từ 12 giờ đến 13 giờ và bữa tối từ 18 giờ đến 19 giờ. Sẽ thật tuyệt nếu bạn có một chương trình có thể cho bạn biết nên ăn gì khi nào?

Trong meal.py, triển khai một chương trình nhắc người dùng về thời gian và đưa ra kết quả là breakfast time, lunch time, hay dinner time. Nếu chưa đến giờ ăn thì đừng nói gì cả. Giả sử rằng đầu vào của người dùng sẽ được định dạng trong thời gian 24 giờ dưới dạng #:## hoặc ##:##. Và giả sử rằng khoảng thời gian của mỗi bữa ăn đều được bao gồm. Ví dụ: cho dù lúc đó là 7:00, 7:01, 7:59 hay 8:00 hay bất cứ lúc nào trong khoảng thời gian đó thì đã đến giờ ăn sáng.

Cấu trúc chương trình của bạn theo bên dưới, trong đó convert là một hàm (có thể được gọi bằng main) chuyển đổi thời gian ở định dạng 24 giờ, thành số giờ tương ứng dưới dạng float. Ví dụ: cho một time như "7:30" (tức là 7 giờ 30 phút), convert sẽ trả về 7.5 (tức là 7,5 giờ).

```
def main():
    ...


def convert(time):
    ...


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

Gợi ý

- Hãy nhớ lại rằng a strđi kèm với khá nhiều phương thức, theo docs.python.org/3/library/stdtypes.html#string-methods , bao gồm split, phân tách a str thành một chuỗi các giá trị, tất cả đều có thể được gán cho các biến cùng một lúc . Ví dụ: nếu time là str giống như "7:30" thì

- hours, minutes = time.split(":")

sẽ gán "7" cho hours và "30" cho minutes.

- Hãy nhớ rằng có 60 phút trong 1 giờ