

Bài 1. Emojize

Vì biểu tượng cảm xúc không dễ nhập như văn bản, ít nhất là trên máy tính xách tay và máy tính để bàn nên một số chương trình hỗ trợ “mã”, nhờ đó bạn có thể nhập, chẳng hạn như, `:thumbs_up:` sẽ được tự động chuyển đổi thành 👍. Một số chương trình còn hỗ trợ thêm bí danh, nhờ đó bạn có thể nhập ngắn gọn hơn, chẳng hạn như `:thumbsup:` cũng sẽ được tự động chuyển đổi thành 👍.

Xem carpedm20.github.io/emoji/all.html?enableList=enable_list_alias để biết danh sách các mã có bí danh.

Trong một tệp có tên `emojize.py`, hãy triển khai một chương trình nhắc người dùng nhập a str bằng tiếng Anh và sau đó xuất ra phiên bản “biểu tượng cảm xúc” của đó str, chuyển đổi bất kỳ mã (hoặc bí danh) nào trong đó thành biểu tượng cảm xúc tương ứng của họ.

Gợi ý

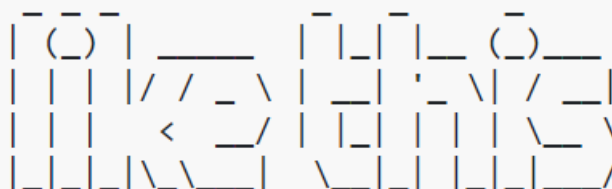
- Lưu ý rằng mô-đun emoji này có hai hàm, theo pypi.org/project/emoji/, một trong số đó là `emojize`, hàm này nhận một tham số có tên tùy chọn, được gọi là `language`. Bạn có thể cài đặt nó với:
- `pip install emoji`

Cách test:

- Chạy chương trình của bạn với `python emojize.py`. Nhập `:1st_place_medal:` và nhấn Enter. Chương trình của bạn sẽ xuất ra: Output: 🏆
- Chạy chương trình của bạn với `python emojize.py`. Nhập `:money_bag:` và nhấn Enter. Chương trình của bạn sẽ xuất ra: Output: 💰
- Chạy chương trình của bạn với `python emojize.py`. Nhập `:smile_cat:` và nhấn Enter. Chương trình của bạn sẽ xuất ra: Output: 😸

Bài 2. FIGlet

[FIGlet](#), được đặt theo tên [các chữ cái của Frank, Ian và Glen](#), là một chương trình từ đầu những năm 1990 để tạo các chữ cái lớn từ văn bản thông thường, một hình thức [nghệ thuật ASCII](#):



Trong số các phong chữ được FIGlet hỗ trợ có những phong chữ tại figlet.org/examples.html.

FIGlet kể từ đó đã được chuyển sang Python dưới dạng mô-đun có tên [pyfiglet](#).

Trong một tệp có tên `figlet.py`, hãy triển khai một chương trình:

- Yêu cầu không hoặc hai đối số dòng lệnh:
 - Bằng 0 nếu người dùng muốn xuất văn bản bằng phong chữ ngẫu nhiên.
 - Hai nếu người dùng muốn xuất văn bản bằng một phong chữ cụ thể, trong trường hợp đó chữ đầu tiên trong hai chữ phải là `-f` hoặc `-font` và chữ thứ hai trong hai chữ phải là tên của phong chữ.
- Nhắc người dùng về một str văn bản.
- Xuất văn bản đó theo phong chữ mong muốn.

Nếu người dùng cung cấp hai đối số dòng lệnh và đối số đầu tiên không phải `-f` hoặc `-font` hoặc đối số thứ hai không phải là tên của phong chữ thì chương trình sẽ thoát ra kèm theo `sys.exit` thông báo lỗi.

Gợi ý

- Bạn có thể cài đặt `pyfiglet` với:
- `pip install pyfiglet`
- Tài liệu về `pyfiglet` không rõ ràng lắm, nhưng bạn có thể sử dụng mô-đun như sau:
- `from pyfiglet import Figlet`
-
- `figlet = Figlet()`

Sau đó, bạn có thể nhận được list các phong chữ có sẵn với mã như thế này:

```
figlet.getFonts()
```

Bạn có thể đặt phong chữ bằng mã như thế này, trong đó `f` tên phong chữ là str:

```
figlet.setFont(font=f)
```

Và bạn có thể xuất văn bản bằng phong chữ đó với mã như thế này, trong đó `s` văn bản đó dưới dạng str:

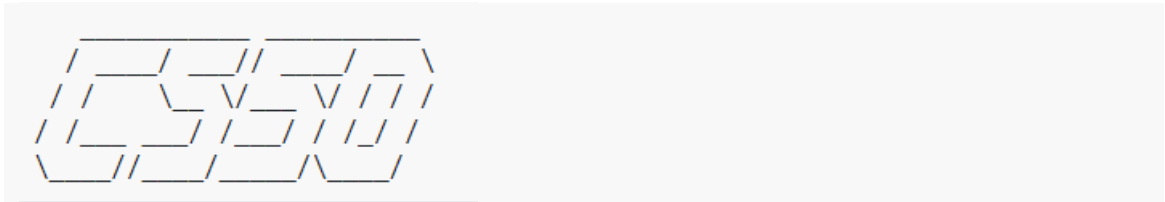
```
print(figlet.renderText(s))
```

- Lưu ý rằng mô-đun `random` này có khá nhiều chức năng, theo docs.python.org/3/library/random.html.

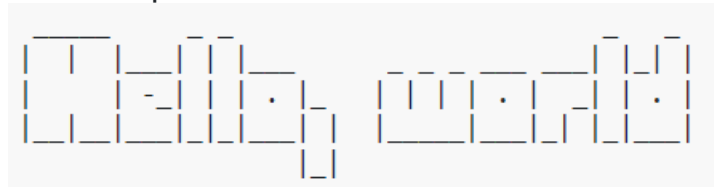
Cách test:

- Chạy chương trình của bạn với `python figlet.py test`. Chương trình của bạn sẽ thoát qua `sys.exit` và in thông báo lỗi: `Invalid usage`
- Chạy chương trình của bạn với `python figlet.py -a slant`. Chương trình của bạn sẽ thoát qua `sys.exit` và in thông báo lỗi: `Invalid usage`
- Chạy chương trình của bạn với `python figlet.py -f invalid_font`. Chương trình của bạn sẽ thoát qua `sys.exit` và in thông báo lỗi: `Invalid usage`

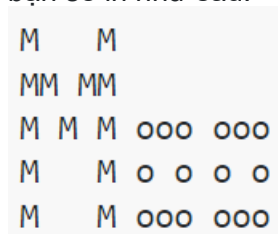
- Chạy chương trình của bạn với `python figlet.py -f slant`. Nhập CS50. Chương trình của bạn sẽ in như sau:



- Chạy chương trình của bạn với `python figlet.py -f rectangles`. Nhập Hello, world. Chương trình của bạn sẽ in như sau:



- Chạy chương trình của bạn với `python figlet.py -f alphabet`. Nhập Moo. Chương trình của bạn sẽ in như sau:



Bài 3. Adieu

Trong [The Sound of Music](#), có một bài hát được hát phần lớn bằng tiếng Anh, [So Long, Farewell](#), với [lời bài hát](#) này, trong đó “adieu” có nghĩa là “goodbye” trong tiếng Pháp: **Adieu, adieu, to yieu and yieu and yieu**

Tất nhiên, dòng này không đúng ngữ pháp vì nó thường được viết (với [Oxford comma](#)) là: **Adieu, adieu, to yieu, yieu, and yieu**

Công bằng mà nói, “yieu” thậm chí không phải là một từ; nó chỉ vần với “you”!

Trong tệp có tên `adieu.py`, hãy triển khai chương trình nhắc người dùng đặt tên, mỗi tên một dòng, cho đến khi người dùng nhập **control-d (trên Linux, MacOS) hoặc control-z (trên Windows)**. Giả sử người dùng sẽ nhập ít nhất một tên. Sau đó tạm biệt những cái tên đó, tách hai tên bằng một `and`, ba tên bằng hai dấu phẩy và một `and`, và `n` tên với `n-1` dấu phẩy và một `and`, như dưới đây:

Adieu, adieu, to Liesl
Adieu, adieu, to Liesl and Friedrich
Adieu, adieu, to Liesl, Friedrich, and Louisa

Adieu, adieu, to Liesl, Friedrich, Louisa, and Kurt
Adieu, adieu, to Liesl, Friedrich, Louisa, Kurt, and Brigitta
Adieu, adieu, to Liesl, Friedrich, Louisa, Kurt, Brigitta, and Marta
Adieu, adieu, to Liesl, Friedrich, Louisa, Kurt, Brigitta, Marta, and Gretl

Gợi ý:

- Lưu ý rằng mô-đun inflect này đi kèm với khá nhiều phương thức, theo pypi.org/project/inflect. Bạn có thể cài đặt nó với:
- `pip install inflect`

Cách test:

- Chạy chương trình của bạn với python adieu.py. Nhập Liesl và nhấn Enter, theo sau là control-d. Chương trình của bạn sẽ xuất ra: Adieu, adieu, to Liesl
- Chạy chương trình của bạn với python adieu.py. Nhập Liesl và nhấn Enter, sau đó nhập Friedrich và nhấn Enter, theo sau là control-d. Chương trình của bạn sẽ xuất ra: Adieu, adieu, to Liesl and Friedrich
- Chạy chương trình của bạn với python adieu.py. Nhập Liesl và nhấn Enter, sau đó nhập Friedrich và nhấn Enter. Bây giờ gõ Louisa và nhấn Enter, theo sau là control-d. Chương trình của bạn sẽ xuất ra: Adieu, adieu, to Liesl, Friedrich, and Louisa

Bài 4. Guessing Game

Trong một tệp có tên `game.py`, hãy triển khai một chương trình:

- Nhắc người dùng về một cấp độ, `n`. Nếu người dùng không nhập số nguyên dương, chương trình sẽ nhắc lại.
- Tạo ngẫu nhiên một số nguyên từ **1 đến `n`**, bao gồm cả **1** và **`n`**, sử dụng mô-đun `random`.
- Nhắc người dùng đoán số nguyên đó. Nếu dự đoán không phải là số nguyên dương, chương trình sẽ nhắc lại người dùng.
 - Nếu dự đoán nhỏ hơn số nguyên đó, chương trình sẽ xuất ra **Too small!** và nhắc lại người dùng.
 - Nếu dự đoán lớn hơn số nguyên đó, chương trình sẽ xuất ra **Too large!** và nhắc lại người dùng.
 - Nếu dự đoán giống với số nguyên đó, chương trình sẽ xuất **Just right!** và thoát.

Gợi ý

- Lưu ý rằng mô-đun `random` này có khá nhiều chức năng, theo docs.python.org/3/library/random.html.

Bài 5. Little professor

Một trong những món đồ chơi đầu tiên của David khi còn nhỏ, khá buồn cười, là [Giáo sư nhỏ](#), một “máy tính” có thể tạo ra mười bài toán khác nhau để David giải. Ví dụ: nếu đồ chơi được hiển thị $4 + 0 =$, (hy vọng) David sẽ trả lời bằng 4. Nếu món đồ chơi được trưng bày $4 +$

1 = , (hy vọng) David sẽ trả lời là 5. Nếu David trả lời sai, đồ chơi sẽ hiển thị EEE. Và sau ba câu trả lời sai cho cùng một bài toán, đồ chơi sẽ chỉ hiển thị câu trả lời đúng (ví dụ: $4 + 0 = 4$ hoặc $4 + 1 = 5$).

Trong một tệp có tên professor.py, hãy triển khai một chương trình:

- Nhắc người dùng về một cấp độ, **n**. Nếu người dùng không nhập **1**, **2**, hoặc **3**, chương trình sẽ nhắc lại.
- Tạo ngẫu nhiên mười (10) bài toán có định dạng là $X + Y =$, trong đó mỗi bài X và Y là một số nguyên không âm với **n** chữ số. Không cần hỗ trợ các thao tác khác ngoài phép cộng (+).
- Nhắc người dùng giải quyết từng vấn đề đó. Nếu một câu trả lời không đúng (hoặc thậm chí không phải là một con số), chương trình sẽ xuất ra EEE và nhắc lại người dùng, cho phép người dùng tổng cộng tối đa ba lần thử cho vấn đề đó. Nếu người dùng vẫn không trả lời đúng sau ba lần thử, chương trình sẽ đưa ra câu trả lời đúng.
- Cuối cùng, chương trình sẽ xuất ra điểm của người dùng: số câu trả lời đúng trên 10.

Cấu trúc chương trình của bạn như sau, trong đó **get_level** nhắc nhở (và, nếu cần, nhắc lại) người dùng về một cấp độ và trả về **1**, **2**, hoặc **3**, và **generate_integer** trả về một số nguyên không âm được tạo ngẫu nhiên có **level** các chữ số hoặc đưa ra một ngoại lệ **ValueError** nếu **level** không phải 1, 2, hoặc 3:

```
import random

def main():
    ...

def get_level():
    ...

def generate_integer(level):
    ...

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

Gợi ý

- Lưu ý rằng bạn có thể đưa ra một ngoại lệ giống như mã **ValueError** như:
- **raise ValueError**
- Lưu ý rằng mô-đun random này có khá nhiều chức năng, theo docs.python.org/3/library/random.html .

Cách test:

- Chạy chương trình của bạn với `python professor.py`. Nhập `-1` và nhấn Enter. Chương trình của bạn sẽ nhắc nhở bạn: **Level:**
- Chạy chương trình của bạn với `python professor.py`. Nhập `4` và nhấn Enter. Chương trình của bạn sẽ nhắc nhở bạn: **Level:**
- Chạy chương trình của bạn với `python professor.py`. Nhập `1` và nhấn Enter. Chương trình của bạn sẽ bắt đầu đặt ra các bài toán cộng với các số nguyên dương, có một chữ số. Ví dụ:

`6 + 6 =`

Chương trình của bạn sẽ (lần lượt) xuất ra 10 bài toán riêng biệt (và nhận câu trả lời) trước khi in số câu hỏi mà bạn đã trả lời đúng và thoát ra.

- Chạy chương trình của bạn với `python professor.py`. Nhập `1` và nhấn Enter. Trả lời sai câu hỏi đầu tiên. Chương trình của bạn sẽ xuất ra: **EEE** trước khi nhắc lại câu hỏi tương tự với bạn.
- Chạy chương trình của bạn với `python professor.py`. Nhập `1` và nhấn Enter. Trả lời sai câu hỏi đầu tiên ba lần. Chương trình của bạn sẽ đưa ra câu trả lời đúng. Ví dụ: **6 + 6 = 12** rồi chuyển sang câu hỏi khác. Trả lời đúng các câu hỏi còn lại. Chương trình của bạn sẽ xuất ra điểm **9**.
- Chạy chương trình của bạn với `python professor.py`. Nhập `1` và nhấn Enter. Trả lời đúng tất cả **10** câu hỏi. Chương trình của bạn sẽ xuất ra điểm **10**.

Bài 6. Chỉ số giá Bitcoin

Bitcoin là một dạng tiền kỹ thuật số, hay còn gọi là tiền điện tử. Thay vì dựa vào cơ quan trung ương như ngân hàng, Bitcoin dựa vào mạng phân tán, hay còn gọi là blockchain, để ghi lại các giao dịch.

Bởi vì có nhu cầu về Bitcoin (tức là người dùng muốn nó), nên người dùng sẵn sàng mua nó, bằng cách đổi một loại tiền tệ (ví dụ: USD) lấy Bitcoin.

Trong một tệp có tên `bitcoin.py`, hãy triển khai một chương trình:

- Yêu cầu người dùng chỉ định số lượng Bitcoin làm đối số dòng lệnh, `n`, mà họ muốn mua. Nếu đối số đó không thể được chuyển đổi thành a float, chương trình sẽ thoát ra `sys.exit` với thông báo lỗi.
- Truy vấn API cho Chỉ số giá Bitcoin của CoinDesk tại <https://api.coindesk.com/v1/bpi/currentprice.json>, trả về một đối tượng JSON, trong đó các khóa lồng nhau của nó là giá hiện tại của Bitcoin dưới dạng số thực float. Hãy chắc chắn bắt mọi trường hợp ngoại lệ, chẳng hạn như với mã như:

- `import requests`
- `try:`
- `...`
- `except requests.RequestException:`
- `...`
- Xuất ra chi phí hiện tại của `n` Bitcoin bằng USD đến bốn chữ số thập phân, sử dụng `''` làm dấu phân cách hàng nghìn.

Gợi ý

- Hãy nhớ lại rằng mô-đun sys đi kèm với argv, theo docs.python.org/3/library/sys.html#sys.argv.
- Lưu ý rằng mô-đun requests này đi kèm với khá nhiều phương thức, theo request.readthedocs.io/en/latest, trong số đó có `get`, theo request.readthedocs.io/en/latest/user/quickstart/#make-a-request và `json`, theo <https://request.readthedocs.io/en/latest/user/quickstart/#json-response-content>. Bạn có thể cài đặt nó với:

- `pip install requests`
- Lưu ý rằng API của CoinDesk trả về phản hồi JSON như:

```
{
  "time":{
    "updated":"May 2, 2022 15:27:00 UTC",
    "updatedISO":"2022-05-02T15:27:00+00:00",
    "updateduk":"May 2, 2022 at 16:27 BST"
  },
  "disclaimer":"This data was produced from the CoinDesk Bitcoin Price Index (USD). Non-USD
currency data converted using hourly conversion rate from openexchangerates.org",
  "chartName":"Bitcoin",
  "bpi":{
    "USD":{
      "code":"USD",
      "symbol":"&#36;",
      "rate":"38,761.0833",
      "description":"United States Dollar",
      "rate_float":38761.0833
    },
    "GBP":{
      "code":"GBP",
      "symbol":"&pound;",
      "rate":"30,827.6198",
      "description":"British Pound Sterling",
      "rate_float":30827.6198
    },
    "EUR":{
      "code":"EUR",
      "symbol":"&euro;",
      "rate":"36,800.2764",
      "description":"Euro",
      "rate_float":36800.2764
    }
  }
}
```

- Hãy nhớ rằng bạn có thể định dạng USD thành bốn chữ số thập phân bằng dấu [phân cách hàng nghìn](#) với mã như:
- `print(f'${amount:,.4f}')`

Cách test:

- Chạy chương trình của bạn với `python bitcoin.py`. Chương trình của bạn nên sử dụng `sys.exit` để thoát với thông báo lỗi: Missing command-line argument
- Chạy chương trình của bạn với `python bitcoin.py cat`. Chương trình của bạn nên sử dụng `sys.exit` để thoát với thông báo lỗi: Command-line argument is not a number
- Chạy chương trình của bạn với `python bitcoin.py 1`. Chương trình của bạn phải xuất giá của một Bitcoin đến bốn chữ số thập phân, sử dụng `'` làm [dấu phân cách hàng nghìn](#) .