AI VIET NAM - COURSE 2022
Bài kiểm tra đầu vào Python

(**Lưu ý:** không cần sử dụng hàm *input()*, các bạn hãy gán giá trị (theo ví dụ) trực tiếp cho các biến Input của bài toán.)

- **Câu 1 (Basic Operation):** Viết chương trình in ra phần dư của phép chia hai số nguyên.
  - **Input:** Hai số nguyên a, b.
  - **Output:** Phần dư của phép chia $\frac{a}{b}$.
  - **Ví dụ:**
    - Input: a = 2 b = 3
    - Output: 2
- **Câu 2 (Conditional Sentence):** Viết chương trình kiểm tra tính chẵn lẻ của một số.
  - **Input:** Một số nguyên n.
  - **Output:** Trả về True nếu n là số chẵn và ngược lại.
  - **Ví dụ:**
    - Input: n = 3
    - Output: False
- **Câu 3 (List Comprehension):** Viết chương trình nhân tất cả các phần tử trong một list các số nguyên bởi một số k và loại bỏ các phần tử ≤ 0.
  - **Input:** Một list các số nguyên.
  - **Output:** Một list các số nguyên dương.
  - **Ví dụ:**
    - Input: lst = [-1, -2, -3, 0, 1, 2, 3], k = 2
    - Output: [2, 4, 6]
- **Câu 4 (Loops):** Dãy Fibonacci là một dãy số có dạng $0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...$; các con số trong dãy được tính dựa trên công thức như sau:

$$fibo(k) = fibo(k-1) + fibo(k-2)$$

Trong đó k là vị trí trong dãy Fibonacci. Dựa vào định nghĩa trên, các bạn hãy viết chương trình in ra số tại vị trí thứ k trong dãy Fibonacci.

- **Input:** Một số nguyên k ($k \geq 0$).
- **Output:** Số Fibonacci tại vị trí thứ k.
- **Ví dụ:**
  - Input: k = 6
  - Output: 8

- **Câu 5 (String Manipulation):** Viết chương trình đếm số lần xuất hiện của các chữ cái trong 1 chuỗi cho trước. Lưu ý: cần loại bỏ các dấu câu (dấu chấm, dấu phẩy, khoảng trắng...) và đổi toàn bộ chữ viết hoa thành chữ viết thường nếu có.
  - **Input:** Một chuỗi s.
  - **Output:** Một dictionary chứa số lần xuất hiện của các chữ cái có trong chuỗi s.
  - **Ví dụ:**
    - Input: s = "I'm learning Artificial Intelligence."
    - Output: {'i': 7, 'm': 1, 'l': 4, 'e': 4, 'a': 3, 'r': 2, 'n': 4, 'g': 2, 't': 2, 'f': 1, 'c': 2}

- **Câu 6 (Function):** Viết chương trình khai báo một hàm có tên *loss()*, nhận ba tham số đầu vào là y, y_hat và δ, trả về giá trị là kết quả của phương trình sau:

$$loss(y, y\_hat, \delta) = \begin{cases} \frac{1}{2}(y - y\_hat)^2, & |y - y\_hat| \leq \delta \\ \delta(|y - y\_hat| - \frac{1}{2}\delta), & |y - y\_hat| > \delta \end{cases}$$

  - **Input:** Ba giá trị y, y_hat và δ.
  - **Output:** Kết quả lời gọi hàm loss(y, y_hat, δ).
  - **Ví dụ:**
    - Input: $y = 1.5, y\_hat = 2, \delta = 0.5$
    - Output: $loss(1.5, 2, 0.5) = 0.125$

- **Câu 7 (Files I/O):** Khi làm việc với các mã nguồn liên quan đến mô hình Machine Learning/Deep Learning, ta có thể sẽ bắt gặp yêu cầu chuyển đổi bộ dữ liệu thành một dạng file .json chứa các trường thông tin của bộ dữ liệu mà tác giả mã nguồn đã quy định sẵn từ đó mới có thể thực hiện huấn luyện, đánh giá mô hình. Vì vậy, kĩ năng thao tác và xử lý file có thể coi là rất quan trọng.

  Cho bộ dữ liệu VIVOS (các bạn sẽ sử dụng phiên bản rút gọn tại đây) về Nhận diện giọng nói (Speech Recognition). Tổng quan về folder bộ dữ liệu sẽ bao gồm một số những thành phần sau:

- **Folder waves:** chứa folder có các file âm thanh .wav.
- **File prompts.txt:** thông tin nhãn về bản dịch của từng file âm thanh.
- **File genders.txt:** thông tin nhãn về giới tính của người nói thuộc folder chứa file âm thanh.

Các bạn hãy sử dụng sử dụng các kĩ thuật thao tác file trong Python, viết chương trình tạo một file .json chứa thông tin về ba mẫu dữ liệu trong bộ dữ liệu VIVOS rút gọn. Cấu trúc thông tin của một mẫu dữ liệu trong file .json sẽ có dạng như sau:

```
{
    "filepath": "...",
    "gender": "...",
    "prompt": "..."
}
```

Listing 1: Trường thông tin của một mẫu dữ liệu trong file json

Khi đọc toàn bộ ba mẫu dữ liệu, ta sẽ có một file .json hoàn chỉnh như sau:

```
[
    {
        "filepath": "VIVOSDEV01_R012.wav",
        "gender": "m",
        "prompt": "NHỮNG CƠN GIÓ MẠNH VÀ MƯA ĐÓNG BĂNG GÂY
        TRƠN TRỰT"
    },
    {
        "filepath": "VIVOSDEV01_R002.wav",
        "gender": "m",
        "prompt": "TIẾNG CỌC CẠCH KHỰNG LẠI CỦA NHỮNG
        KHỚP SẮT"
    },
    {
        "filepath": "VIVOSDEV01_R003.wav",
        "gender": "m",
        "prompt": "CŨNG LÊN TIẾNG ỦNG HỘ CÁC KIẾN NGHỊ NÀY"
    }
]
```

Listing 2: File json hoàn chỉnh

- **Input:** Bộ dữ liệu VIVOS rút gọn.
- **Output:** File .json chứa các trường thông tin của bộ dữ liệu VIVOS rút gọn.

- **Câu 8 (Python library, Linear Regresison):** Cho một bảng dữ liệu về thông tin cá nhân của những người tham gia bảo hiểm. Đây là ba hàng đầu tiên của bảng dữ liệu:

age	**sex**	**bmi**	**children**	**smoker**	**region**	**charges**
19	female	27.9	0	yes	southwest	16884.924
18	male	33.77	1	no	southeast	1725.5523
28	male	33	3	no	southeast	4449.462

Bảng 1: Bảng dữ liệu về thông tin cá nhân người tham gia bảo hiểm

Dựa vào đây, người ta có thể áp dụng mô hình máy học để dự đoán được số tiền chi trả bảo hiểm (charges) của một người tham gia bảo hiểm bất kì dựa trên các thông tin cá nhân của họ (age, sex, bmi, children, smoker, region). Các bạn hãy viết chương trình sử dụng module Linear Regression trong thư viện *scikit-learn* để huấn luyện mô hình Linear Regression trên bộ dữ liệu đầu vào.

Gợi ý: các bạn hãy truy cập đường link này để tham khảo code mẫu và áp dụng vào bộ dữ liệu của mình.

- **Input:** Bảng dữ liệu insurance.csv.
- **Output:** Mô hình Linear Regression đã được huấn luyện trên bộ dữ liệu từ Input (chỉ cần làm tới bước fit thành công dữ liệu vào mô hình).

- **Câu 9 (Python library, Image Processing):** Trong xử lý ảnh, người ta thường sử dụng ảnh nhị phân (binary image), một loại ảnh chỉ gồm có các pixel trắng và đen, trong việc nhận diện các vật thể cụ thể có trong ảnh, đóng vai trò là bước tiền xử lý dữ liệu trong một số bài toán liên quan đến dữ liệu ảnh và còn nhiều ứng dụng khác...

Các bạn hãy viết chương trình chuyển đổi ảnh đầu vào thành ảnh nhị phân. Lưu ý: chỉ được sử dụng các thư viện xử lý ảnh như *cv2*, *PIL*... để đọc ảnh, bước nhị phân hóa ảnh có thể thực hiện bằng *Python* hoặc *NumPy*.

- **Input:** Một ảnh bất kì.
- **Output:** Ảnh nhị phân của ảnh đầu vào.
- **Ví dụ:**

Hình 1: Ảnh Input

Hình 2: Ảnh Output

- **Câu 10 (Machine Learning Metric):** Trong lĩnh vực truy vấn thông tin (Information Retrieval), để đánh giá được hệ thống truy vấn văn bản nào cho hiệu suất tốt hơn, người ta đã sử dụng một độ đo có tên gọi là Interpolated Precision.

  Một số ký hiệu/khái niệm:

  - q: câu truy vấn (query).
  - c: danh sách các tài liệu liên quan đến câu truy vấn q.
  - s: danh sách các tài liệu trả về với câu truy vấn q.
  - *precision*: độ đo được tính bằng công thức:

$$p = \frac{\text{Tổng số tài liệu liên quan đến câu truy vấn được trả về}}{\text{Tổng số tài liệu được trả về}}$$

- *recall*: độ đo được tính bằng công thức:

$$r = \frac{\text{Tổng số tài liệu liên quan đến câu truy vấn được trả về}}{\text{Tổng số tài liệu liên quan của câu truy vấn}}$$

- **Lưu ý:** các tài liệu sẽ được biểu diễn dưới dạng là một số nguyên (còn được gọi là *chỉ mục* của tài liệu).

Cách bước tính Interpolated Precision khi xét trên một câu truy vấn q là như sau:

- **Bước 1:** Tính giá trị precision của hệ thống với câu truy vấn q tại các vị trí mà hệ thống trả về tài liệu có nội dung liên quan đến câu truy vấn trong tập tất cả tài liệu mà hệ thống trả về. Lưu ý: theo công thức tính precision đã nêu ở trên, *tổng số tài liệu được trả về* sẽ tính từ vị trí tài liệu trả về đầu tiên đến vị trí tài liệu trả về đang xét trong danh sách các tài liệu trả về.
- **Bước 2:** Dựa vào danh sách kết quả precision của một số mốc recall tìm được ở **Bước 1**. Tính Interpolated Precision trên 11 mốc recall (từ 0% đến 100%) bằng cách lấy giá trị precision cao nhất trong số các precision của các mốc recall lớn hơn hoặc bằng mốc recall đang xét. Cách tính toán trên được diễn tả thành công thức như sau:

$$p_{interpolated}(r) = \max_{r' \geq r} p(r')$$

Dựa trên các bước tính toán trên, các bạn hãy viết chương trình tính **trung bình các Interpolated Precision** trên 1 câu truy vấn của một hệ thống truy vấn văn bản khi đã biết được tập văn bản có liên quan đến câu truy vấn và tập văn bản mà hệ thống trả về sau khi tìm kiếm.

- **Input:** Danh sách tài liệu liên quan đến câu truy vấn và danh sách tài liệu trả về.
- **Output:** Kết quả Interpolated Precision.
- **Ví dụ:**
  - Input: c = [3, 15, 74, 320], s = [1, 3, 15, 30, 58, 74, 100, 129, 190, 241, 320]
  - Output: 0.55372 (kết quả làm tròn đến phần thập phân thứ 5).

– Hết –