Bài tập vận dụng.

Bài 3.1. Anh/Chị hãy viết chương trình Numpy để lấy phiên bản Numpy hiện tại và hiển thị cấu hình xây dựng của Numpy.

Bài 3.2. Viết chương trình với các yêu cầu sau:

1. Tạo numpy array arr có giá trị từ 0-9. Hiển thị các phần tử có trong arr, xem kiểu dữ liệu và kích thước của arr.
2. Từ array arr ở câu 1 => tạo arr_odd và arr_even.
3. Từ array arr ở câu 1 => tạo arr_update_1 với các phần tử chẵn giữ nguyên, các phần tử lẻ thay bằng 100.

Bài 3.3. Cho 2 array arr_a = [1,2,3,2,3,4,3,4,5,6] và arr_b = [7,2,10,2,7,4,9,4,9,8], thực hiện các yêu cầu sau :

1. Tạo array arr_c chỉ lấy duy nhất các phần tử xuất hiện ở cả arr_a và arr_b.
2. Từ arr_a và arr_b ở câu 1 => Tạo arr_d chứa các phần tử chỉ xuất hiện ở arr_a.
3. Cho arr_e = np.array([2, 6, 1, 9, 10, 3, 27, 8, 6, 25, 16]), hãy tạo arr_f chỉ chứa các phần tử có giá trị từ 5 đến 10 của arr_e.

Bài 3.4. Viết chương trình với các yêu cầu sau :

1. Tạo arr_zeros có 10 phần tử 0, cập nhật phần tử ở vị trí thứ 5 là 1.
2. Tạo arr_h có giá trị từ 10 đến 24. In danh sách các phần tử theo tứ tự đảo ngược của arr_h.
3. Cho arr_k = np.array([1, 2, 0, 8, 2, 0, 1, 3, 0, 5, 0]), tạo arr_l từ arr_k với các phần tử khác 0.
4. Từ arr_l của câu 3, thêm 2 phần tử có giá trị là 10 và 20 vào cuối array.
5. Từ array của câu 4, thêm phần tử có giá trị 100 vào vị trí có index = 5.
6. Từ array của câu 5, xóa các phần tử tại vị trí có index = 0, 1, 2.

Bài 3.5. Biết rằng : Major League Baseball (MLB) là giải đấu bóng chày chuyên nghiệp. Major League Baseball có tổng cộng 30 đội bóng đến từ nhiều bang khác nhau của Mỹ và Canada (29 đội từ Mỹ và 1 đội từ Canada). MLB luôn được sự quan tâm lớn của hầu hết fan bóng chày trên toàn thế giới, và cũng được xem là giải đấu nổi tiếng và uy tín nhất, tập hợp những cầu thủ có trình độ cao nhất trong bộ môn này. Dữ liệu heights (tính theo inches) và weights (tính theo pounds) là chiều cao và cân nặng của các cầu thủ có tham gia 1 số giải của MLB.

Dữ liệu được trích xuất từ địa chỉ :

http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_MLB_HeightsWeights

Cho tập tin dữ liệu **heights_1.txt**, **weights_1.txt** => hãy chép dữ liệu từ tập tin này vào list là height, weight, và thực hiện các yêu cầu sau:

1. Tạo numpy array arr_height từ list height.
2. Tạo numpy array arr_weight từ list weight.
3. Cho hệ số quy đổi từ inch sang m là 0.0254, tạo arr_height_m dựa trên công thức: arr_height * hệ số quy đổi.
4. Cho hệ số quy đổi từ pound sang kg là 0.453592, tạo arr_weight_kg dựa trên công thức: arr_weight * hệ số quy đổi.
5. Tính BMI của arr_height_m và arr_weight_kg theo công thức BMI = Cân nặng / (Chiều cao * Chiều cao), và lưu vào arr_bmi.
6. Cho biết giá trị cân nặng ở vị trí index = 50 trong arr_weight_kg.
7. Tạo arr_height_m_100 gồm các phần tử có vị trí index từ 100 đến 110 (lấy cả index 110) trong arr_height_m.
8. Cho biết các cầu thủ bóng chày có bmi < 21 trong arr_bmi.
9. Cho biết chiều cao trung bình và cân nặng trung bình của các cầu thủ.
10. Cho biết chiều cao và cân nặng lớn nhất của các cầu thủ.
11. Cho biết chiều cao và cân nặng nhỏ nhất của các cầu thủ.

Bài 3.6. Thao tác trên mảng nhiều chiều :

Viết chương trình thực hiện các yêu cầu sau :

1. Tạo array arr có kích thước 3x3 với các giá trị True
2. Cho arr_1D = np.array([0 1 2 3 4 5 6 7 8]). Tạo array 2 chiều có kích thước 3x3 từ arr_1D, và lưu vào arr_2D Trong arr_2D, chuyển cột 1 sang cột 3 và ngược lại.
3. Từ arr_2D của câu 2 (sau khi đổi thứ tự cột), chuyển dòng 1 sang dòng 2 và ngược lại.
4. Từ arr_2D của câu 3, đảo ngược các dòng của arr_2D.
5. Từ arr_2D của câu 4, đảo ngược các cột của arr_2D.
6. Cho arr_2D_null = np.array([[1, 2, 3], [np.NaN, 5, 6], [7, np.NaN, 9], [4, 5, 6]]), Kiểm tra trong array có giá trị rỗng không?
7. Từ arr_2D_null của câu 6, thay thế giá trị null bằng 0.

Bài 3.7. Thao tác dữ liệu mảng dữ liệu baseball

Cho tập tin baseball_2D.txt => chép dữ liệu từ tập tin vào list là baseball

Dữ liệu baseball cho biết chiều cao (cột 1) tính theo inch và cân nặng (cột 2) tính theo pounds của các cầu thủ

Viết chương trình với các yêu cầu sau:

1. Tạo 2D numpy array tên np_baseball từ baseball. Xem kiểu dữ liệu và kích thước của np_baseball.
2. In các giá trị của dòng thứ 50 trong np_baseball.
3. Tạo numpy array np_weight với dữ liệu được lấy từ cột hai của np_baseball.
4. Cho biết chiều cao của vận động viên thứ 124.
5. Cho biết chiều cao trung bình, cân nặng trung bình của các cầu thủ.
6. Bạn nhận xét gì về mối tương quan giữa chiều cao và cân nặng của các cầu thủ: có/ không có tương quan, tương quan thuận/nghịch.

Bài 3.8. Tính trung bình của chiều cao (height) dựa vào vị trí (position)

Cho 2 tập tin heights.txt và positions.txt => chép dữ liệu từ 2 tập tin vào 2 list là heights và positions :

'GK' (goalkeeper), 'M' (midfield), 'A' (attack) and 'D' (defense)

Hãy viết chương trình với các yêu cầu sau:

1. a) Tạo numpy array np_positions từ list positions. Xem kiểu dữ liệu của np_positions

   b) Tạo numpy array np_heights từ list heights. Xem kiểu dữ liệu của np_heights
2. Tính chiều cao trung bình của các GK.
3. Tính chiều cao trung bình của những vị trí khác (Không phải là GK).
4. Tạo mảng dữ liệu có cấu trúc tự định nghĩa players gồm 'position' kiểu văn bản (U5) và 'height' kiểu 'float'.
5. Sắp mảng players theo height, cho biết vị trí có chiều cao cao nhất và chiều cao thấp nhất.

Bài 3.9. Lọc và sắp xếp dữ liệu UEFA_European_Championship/Euro 2012. (file euro2012.csv)

Thực hiện các thao tác lọc và sắp xếp dữ liệu về Euro 2012 theo những yêu cầu sau:

Tạo data frame euro12 từ dữ liệu trên. In euro2: type, shape, danh sách các cột

1. In giá trị cột Goals
2. Có bao nhiêu đội tham gia Euro2012?
3. In thông tin của Euro2012.
4. Tạo 1 data frame mới từ euro12 có tên là discipline chỉ chứa 3 cột 'Team', 'Yellow Cards', 'Red Cards'.
5. Sắp xếp discipline giảm dần theo 2 cột 'Red Cards', 'Yellow Cards'.
6. a) Tính trung bình Yellow Cards.

   b) Lọc các đội đã ghi hơn 6 bàn thắng.
7. In các đội mà tên bắt đầu bằng 'G' (gợi ý: dùng str.startswith()).

8. In 7 cột đầu của euro12.
9. In tất cả các cột, trừ 3 cột cuối.
10. In các cột Team, Goals, Shooting Accuracy, Yellow Cards, Red Cards.
11. In các cột chỉ hiển thị 'Team','Shooting Accuracy' từ 'England', 'Italy', 'Russia'.

Bài 3.10. *Thống kê dữ liệu - Groupby*

Cho **drinks.csv** là tập tin cung cấp dữ liệu về tình hình tiêu thụ rượu bia ở các quốc gia theo từng châu lục.

Hãy hoàn tất file đính kèm cùng các yêu cầu sau:

1. Đọc dữ liệu từ tập tin drinks.csv với index_col là cột đầu tiên của dữ liệu, và lưu vào biến drink.
   - Cho biết kiểu dữ liệu (type), kích thước (shape) của drink,
   - Hiển thị tên các cột (columns) của drink.
   - Xem 5 dòng dữ liệu đầu tiên (head) và cuối cùng (tail) của drink.
2. Cho biết số lượng bia tiêu thụ trung bình ở mỗi châu lục.
3. Cho biết thông tin thống kê tổng quát (describe) số lượng rượu vang được tiêu thụ ở mỗi châu lục.
4. Cho biết số lượng các loại bia và rượu tiêu thụ trung bình (mean) ở mỗi châu lục.
5. Cho biết giá trị trung vị (median) cho các loại bia và rượu tiêu thụ ở mỗi châu lục.
6. Cho biết số lượng rượu mạnh (spirit_servings) tiêu thụ trung bình, lớn nhất và nhỏ nhất ở mỗi châu lục
7. Sắp xếp dữ liệu tăng dần (sort_values) theo số lượng bia tiêu thụ.
   - Cho biết 5 quốc gia có lượng tiêu thụ bia nhiều nhất,
   - Cho biết 5 quốc gia có lượng tiêu thụ bia ít nhất.

Bài 3.11. *Giao dịch chứng khoán (Stock Trading)*

Cho 3 file .csv sau:

• **stocks1.csv** : date, symbol, open, high, low, close, volume : chứa thông tin giao dịch chứng khoán các công ty khác nhau

• **stocks2.csv** : date, symbol, open, high, low, close, volume : chứa thông tin giao dịch chứng khoán các công ty khác nhau

• **companies.csv**: name, employees, headquarters_city, headquarters_state : chứa thông tin về trụ sở và số lượng nhân viên cho một công ty cụ thể

Viết chương trình thực hiện các yêu cầu sau:

1. a) Đọc file stocks1.csv => đưa dữ liệu vào stocks1.
   - Hiển thị 5 dòng dữ liệu đầu và cuối của stocks1.

- Cho biết kiểu dữ liệu (dtype) của các cột của stocks1.
- Xem thông tin (info) của stocks1.

b) Đọc file stocks2.csv => đưa dữ liệu vào stocks2.

- Hiển thị 5 dòng dữ liệu đầu và cuối của stocks2.
- Cho biết kiểu dữ liệu (dtype) của các cột của stocks2.
- Xem thông tin (info) của stocks2.

c) Đọc file companies.csv => đưa dữ liệu vào companies

- Xem dữ liệu của companies.
- Cho biết kiểu dữ liệu (dtype) của các cột của companies.
- Xem thông tin (info) của companies.

2. Cho biết trong stocks1 có dữ liệu Null hay không? Nếu có, hãy thay thế với quy tắc sau:
   - Nếu Null cột high thì thay bằng giá trị max trên cột high của mã chứng khoán đó.
   - Nếu Null cột low thì thay bằng giá trị min trên cột low của mã chứng khoán đó.
3. Tạo dataframe stocks bằng cách gộp stocks1 và stocks2 theo dòng. Xem 15 dòng dữ liệu cuối của stocks.
4. Tạo dataframe stocks_companies bằng cách gộp stocks và companies.

   Xem 5 dòng dữ liệu đầu của stocks_companies.
5. Cho biết giá (open, high, low, close) trung bình và volume trung bình của mỗi công ty.
6. Cho biết giá đóng cửa (close) trung bình, lớn nhất và nhỏ nhất ở mỗi công ty
7. Tạo cột parsed_time trong stocks_companies bằng cách đổi thời gian sang định dạng DateTime. Cho biết kiểu dữ liệu của cột parsed_time. Hiển thị 5 dòng dữ liệu đầu của stocks_companies.
8. Thêm cột result, nếu giá 'close' > 'open' thì cột result có giá trị 'up', ngược lại 'down'.

Bài 3.12. Phân tích dữ liệu Movies.

Dữ liệu được lấy từ MovieLens website. Download the Dataset theo link:

- Data Source: MovieLens web site (filename: ml-latest-small.zip)
- Location: https://grouplens.org/datasets/movielens/latest/

Hãy sử dụng các dữ liệu về movies được cung cấp để viết chương trình với các yêu cầu sau:

- Đọc dữ liệu & Data Structures.
- Xử lý dữ liệu bị thiếu/ không hợp lệ.
- Gộp DataFrame.
- Lọc dữ liệu theo yêu cầu.
- Thống kê dữ liệu.
- Parsing Timestamps.