

1.

cho list, animals = ['elephant', 'ant', 'dog'], thực hiện các thao tác sau:

- Thêm 'cat' vào cuối list.
- Thêm 'dolphin' vào vị trí thứ 2 trong list.
- Xóa 'ant' khỏi list.
- Xóa phần tử cuối cùng khỏi list.

Sau mỗi bước in list ra để kiểm tra.

2.

Cho nhập vào 2 list có cùng độ dài, merge 2 list theo vị trí tương ứng.

VD:

```
list1 = ["M", "na", "i", "Hoa"]
```

```
list2 = ["y", "me", "s", "ng"]
```

```
=> ['My', 'name', 'is', '']
```

VD:

```
list1 = ["1", "2"]
```

```
list2 = ["7", "8"]
```

```
=> ['17', '28']
```

3.

Xóa chuỗi trống từ list

```
input = ["Mike", "", "", "", "Emma", "", "Kelly", "", "", "Brad"] => output: ["Mike", "Emma", "Kelly", "Brad"] dùng 2 cách for và list comprehension
```

4.

Kiểm tra tất cả các giá trị trong tuple bất kì (n phần tử) trùng nhau hay không:

VD:(10, 10, 10, 10) -> True

(10, 10, 2) -> False

5.

Cho 2 set,

```
set1 = {10, 20, 30, 40, 50}
```

```
set2 = {30, 40, 50, 60, 70}
```

- Tìm set các phần tử có trong set1 hoặc set2 (phần tử có cả trong set1 và set2 vẫn thỏa mãn).
- Tìm set các phần tử có trong cả set1 và set2
- Tìm set các phần tử có trong set1 nhưng không trong set2

6.

Cho dict:

```
information = {  
    "name": "Tuan",  
    "age": 18,  
    "salary": 8000,  
    "city": "Ha Noi"  
}
```

- Thêm vào dict key: "job" với value: "IT"
- In ra salary từ dict
- Xóa key "age" ở dict
- Tăng salary lên 10000 (salary = 10000)

Sau mỗi bước in ra dict để kiểm tra.

7.

Merge 2 list thành 1 dict

keys = ['Ten', 'Twenty', 'Thirty']

values = [10, 20, 30]

=> {'Ten': 10, 'Twenty': 20, 'Thirty': 30}

8.

Cho 1 list các số:

a. tạo dict với key là số, value là số lần xuất hiện số đấy.

num = [1, 2, 3, 1, 2, 1, 1, 4] => dict = {1:4, 2:2, 3:1, 4:1}

b. in ra số mà có lần xuất hiện nhiều nhất và ít nhất trong dãy ( nhiều nhất: 1, ít nhất: 3,4)

c. in ra giá trị trung bình value của dict,  $(4+2+1+1)/4$

9.

Cho 1 dict điểm học sinh:

scores = {'Hoang': 8, 'Giang beo': 7, 'Giang gay': 9, 'meomeo': 10}

a. Cho người dùng nhập tên học sinh nếu học sinh có trong dict thì in ra điểm, nếu không thì in ra học sinh không có trong lớp.

b. Cho người dùng nhập tên và điểm của 1 học sinh, thêm thông tin vào dict

c. In ra các bạn có điểm cao nhất / thấp nhất trong lớp.

10. a. tên không có báo không có sản phẩm này.

b. In ra các sản phẩm có số lượng lớn nhất.

11. Input 1 chuỗi, output ra số kí tự hoa, số kí tự thường, số chữ số và số kí tự khác trong chuỗi.

12. Input 1 chuỗi, output ra 1 dict với key là ký tự, value là số lần chữ đấy xuất hiện. ví dụ: 'google.com' -> {'o': 3, 'g': 2, '.': 1, 'e': 1, 'l': 1, 'm': 1, 'c': 1}

13. Thay thế tất cả kí tự đầu tiên ( cả viết hoa cả viết thường) trong chuỗi bằng dấu \$ (trừ kí tự đầu tiên) ví dụ: "Nguyen Thanh Tuan Nga" -> "Nguye\$ Tha\$h Tua\$ \$ga" ( kí tự đầu tiên là N, thay thế tất cả các kí tự N, n ở trong chuỗi ban đầu mà có vị trí khác 0 bằng dấu \$)

14. Input động từ, output động từ thêm ing. Thêm theo các quy tắc sau

- Động từ tận cùng bằng e, bỏ e câm rồi thêm ing

- Động từ tận cùng bằng ie, ta biến ie thành Y rồi thêm ing

- Nhân đôi phụ âm cuối trước khi thêm ing

- Còn lại chỉ cần thêm ing.

15. Hàm find chỉ trả về vị trí xuất hiện lần đầu tiên của substring. Cho người dùng nhập string và substring, tìm vị trí xuất hiện của tất cả các substring. Nếu substring không có trong chuỗi in thông báo cho người dùng. (Mọi người tự viết lại hàm chứ không dùng luôn hàm find nhé)

16. Hàm replace trong python thay thế vị trí của tất cả các lần xuất hiện.

Cho người dùng nhập vào chuỗi

Cho người dùng nhập chuỗi con muốn thay thế

Cho người dùng nhập chuỗi được thay thế chuỗi con trên

Cho người dùng nhập vị trí thay thế (chỉ số bắt đầu từ 0)

Chỉ thay thế tại vị trí nhập, in ra chuỗi sau khi thay thế.

VD:

Input: "The apple the apple",

replace: "apple" -> "banana",

pos: 1 (index từ 0)

-> output: "The apple the banana"

( Mọi người tự viết lại hàm chứ không dùng luôn hàm replace nhé )

17. Input các từ cách nhau bởi dấu phẩy, output trả về các từ nhưng không trùng nhau, và sắp xếp theo thứ tự bảng chữ cái.

input: "red, white, black, red, green, black" ->

output: "black, green, red, white"

18. Convert tên về dạng chuẩn, các từ cách nhau 1 dấu cách, chữ đầu viết hoa.

VD: input: " Nguyen ngOc HuY hOANG" -> "Nguyen Ngoc Huy Hoang"

19. Input string dạng "dd-mm-yyyy" in ra ngày tháng năm bằng chữ, ví dụ "02-12-2020" -> ngày hai tháng mười hai năm hai nghìn hai mươi.

20. Cho chuỗi

```
'''Interface ethernet0 is up
```

```
Interface ethernet1 is down
```

```
Interface serial0 is down
```

```
Interface serial1 is up '''
```

a. Chữ interface bị viết nhầm thành inteface, thay thế tất cả inteface thành interface.

b. Đếm xem từ "up" xuất hiện bao nhiêu lần.

c. In ra tên các interface đang up -> ["ethernet0", "serial1"]

21.

a. Tạo 1D array các số nguyên,  $x = \text{np.array}([\dots])$

b. Thay thế các số lẻ trong array bằng -1

Hint: Thử  $x \% 2 == 1$  và  $x[[\text{True}, \text{False}, \text{True}]]$  (vector x có 3 phần tử)

22.

a. Tạo 2D array (matrix) A kích thước  $3 \times 2$  gồm toàn số 1, và B kích thước  $3 \times 2$  với các số nguyên dương từ 1->6.

b. Thực hiện phép tính cộng, trừ, nhân (các phần tử tương ứng), chia (các phần tử tương ứng)

c. Thực hiện phép tính nhân ma trận A và B.T (B transpose)

23.

a. Tạo 2D array (matrix) A kích thước  $3 \times 4$  ngẫu nhiên các số nguyên.

b. Tính tổng ma trận theo hàng. Có thuộc tính keepdims mọi người thử xem.

c. Tính tổng ma trận theo cột.

d. Lấy ra ma trận gồm 2 hàng đầu tiên và 2 cột cuối cùng.

24

a. Cho 1D array kích thước 10, convert về 2D array (matrix) x kích thước  $5 \times 2$  (Hint: reshape)

b. Đảo ngược thứ tự các cột trong x ( $[[1,2],[3,4]] \rightarrow [[2,1],[4,3]]$ )

25.

Cho array:

```
dataset = np.array([
```

```
['Quần thun', 10, 50000],
```

```
['Áo gió', 5, 90000],
```

```
['Tất', 50, 8000],
```

```
['Giày', 1, 200000],
```

```
['Mũ', 1, 30000],
```

```
['Quần jean', 3, 100000],  
['Áo len', 10, 100000],  
])
```

có 3 cột tương ứng: tên sản phẩm, số lượng, giá mỗi sản phẩm.

- Tính số lượng sản phẩm trong dataset.
- Lấy ra các sản phẩm có giá lớn hơn 50000.
- Lấy ra các sản phẩm đã có doanh thu hơn 300000.

26. Cho 2 list tháng và lợi nhuận tương ứng. Vẽ biểu đồ line thể hiện dữ liệu với đầy đủ các label  
27.

- Vẽ biểu đồ 2 line thể hiện lợi nhuận của 2 công ty A và B, có legend để phân biệt đường nào của công ty nào.
- Vẽ 4 subplots cho 4 công ty A, B, C, D (tương tự bài 1) trên cùng 1 figure.

28. Sửa bài 2 sao cho mỗi subplot có 1 màu và cách hiển thị đồ thị riêng (b-, ro, ..)

29. Cho dict: {'10-20':[10, 15], '20-30':[50, 75], '30-40':[40, 55], '40-50':[30, 25]} với key là nhóm tuổi, value là 1 list thể hiện số người đi xe đạp nam và nữ theo thứ tự lần lượt:

- Vẽ biểu đồ bar thể hiện số liệu
- Vẽ biểu đồ bar chồng số liệu nam và nữ lên nhau ở mỗi cột

30. Cho dict tỉ lệ sử dụng phương tiện giao thông: transport = {'car': 0.2, 'bus': 0.5, 'bike': 0.2, 'motorbike': 0.1}. Vẽ pie chart cho số liệu trên.

31. Cho 1 list điểm thi của sinh viên, vẽ histogram với bin = 10 cho dữ liệu.

32. Cho 1 list điểm thi của sinh viên, vẽ boxplot để xem phân bố của điểm thi

33. Tải [https://archive.ics.uci.edu/ml/machine-learning-databases/housing/?fbclid=IwAR2-8JhUaUHOTTshQUrFfvN4\\_bGCQVqMfs3g2yquJxpU08F9S6lxWGnCrEw](https://archive.ics.uci.edu/ml/machine-learning-databases/housing/?fbclid=IwAR2-8JhUaUHOTTshQUrFfvN4_bGCQVqMfs3g2yquJxpU08F9S6lxWGnCrEw) đọc file housing.data (nhớ thuộc tính sep='\\s+')

- Kiểm tra số hàng, cột
- Dựa vào file housing.names thêm tên các column cho dữ liệu
- Tính mean, min, max của từng cột dữ liệu (hàm describe)
- Kiểm tra dữ liệu có bị missing không, nên bỏ hàng/cột missing hay thay thế?
- Tính correlation giữa các cột dữ liệu ( [https://www.youtube.com/watch?v=ugd4k3dC\\_8Y](https://www.youtube.com/watch?v=ugd4k3dC_8Y) )