[Bài tập Python](https://quantrimang.com/hon-100-bai-tap-python-co-loi-giai-code-mau-142456) level 1

Bài 01:

Câu hỏi:

Viết chương trình tìm tất cả các số chia hết cho 7 nhưng không phải bội số của 5, nằm trong đoạn 2000 và 3200 (tính cả 2000 và 3200). Các số thu được sẽ được in thành chuỗi trên một dòng, cách nhau bằng dấu phẩy.

Gợi ý:

Sử dụng range(#begin, #end)

Bài 02:

Câu hỏi:

Viết một chương trình có thể tính giai thừa của một số cho trước. Kết quả được in thành [chuỗi](https://quantrimang.com/gioi-thieu-qua-ve-chuoi-so-list-trong-python-140881) trên một dòng, phân tách bởi dấu phẩy. Ví dụ, số cho trước là 8 thì kết quả đầu ra phải là 40320.

Gợi ý:

Trong trường hợp dữ liệu đầu vào được cung cấp, bạn hãy chọn cách để người dùng nhập số vào.

Bài 03:

Câu hỏi:

Với số nguyên n nhất định, hãy viết chương trình để tạo ra một [dictionary](https://quantrimang.com/gioi-thieu-qua-ve-chuoi-so-list-trong-python-140881) chứa (i, i\*i) như là số nguyên từ 1 đến n (bao gồm cả 1 và n) sau đó in ra dictionary này. Ví dụ: Giả sử số n là 8 thì đầu ra sẽ là: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}.

Gợi ý:

Viết lệnh yêu cầu nhập số nguyên n.

Bài 04:

Câu hỏi:

Viết chương trình chấp nhận một chuỗi số, phân tách bằng dấu phẩy từ giao diện điều khiển, tạo ra một danh sách và một tuple chứa mọi số.

Ví dụ: Đầu vào được cung cấp là 34,67,55,33,12,98 thì đầu ra là:

['34', '67', '55', '33', '12', '98']  
('34', '67', '55', '33', '12', '98')

Gợi ý:

Viết lệnh yêu cầu nhập vào các giá trị sau đó dùng quy tắc chuyển đổi kiểu dữ liệu để hoàn tất.

Bài 05:

Câu hỏi:

Định nghĩa một class có ít nhất 2 method:

getString: để nhận một chuỗi do người dùng nhập vào từ giao diện điều khiển.

printString: in chuỗi vừa nhập sang chữ hoa.

Thêm vào các hàm hiểm tra đơn giản để kiểm tra method của class.

Ví dụ: Chuỗi nhập vào là quantrimang.com thì đầu ra phải là: QUANTRIMANG.COM

Gợi ý:

Sử dụng \_\_init\_\_ để xây dựng các tham số.

Bài 06:

Câu hỏi:

Viết một method tính giá trị bình phương của một số.

Gợi ý:

Sử dụng toán tử \*\*.

Bài 07:

Câu hỏi:

Python có nhiều hàm được tích hợp sẵn, nếu không biết cách sử dụng nó, bạn có thể đọc tài liệu trực tuyến hoặc tìm vài cuốn sách. Nhưng Python cũng có sẵn tài liệu về hàm cho mọi hàm tích hợp trong Python. Yêu cầu của bài tập này là viết một chương trình để in tài liệu về một số hàm Python được tích hợp sẵn như abs(), int(), input() và thêm tài liệu cho hàm bạn tự định nghĩa.

Gợi ý:

Sử dụng \_\_doc\_\_

Bài 08:

Câu hỏi:

Định nghĩa một lớp gồm có tham số lớp và có cùng tham số instance

Gợi ý:

Khi định nghĩa tham số instance, cần thêm nó vào \_\_init\_\_

Bạn có thể khởi tạo một đối tượng với tham số bắt đầu hoặc thiết lập giá trị sau đó.

Bài tập Python level 2

Bài 09:

Câu hỏi:

Viết chương trình và in giá trị theo công thức cho trước: Q = √([(2 \* C \* D)/H]) (bằng chữ: Q bằng căn bậc hai của [(2 nhân C nhân D) chia H]. Với giá trị cố định của C là 50, H là 30. D là dãy giá trị tùy biến, được nhập vào từ giao diện người dùng, các giá trị của D được phân cách bằng dấu phẩy.

Ví dụ: Giả sử chuỗi giá trị của D nhập vào là 100,150,180 thì đầu ra sẽ là 18,22,24.

Gợi ý:

Nếu đầu ra nhận được là một số dạng thập phân, bạn cần làm tròn thành giá trị gần nhất, ví dụ 26.0 sẽ được in là 26.

Trong trường hợp dữ liệu đầu vào được cung cấp cho câu hỏi, nó được giả định là đầu vào do người dùng nhập từ giao diện điều khiển.

Bài 10:

Câu hỏi:

Viết một chương trình có 2 chữ số, X, Y nhận giá trị từ đầu vào và tạo ra một mảng 2 chiều. Giá trị phần tử trong hàng thứ i và cột thứ j của mảng phải là i\*j.

Lưu ý: i=0,1,...,X-1; j=0,1,...,Y-1.

Ví dụ: Giá trị X, Y nhập vào là 3,5 thì đầu ra là: [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]

Gợi ý:

Viết lệnh để nhận giá trị X, Y từ giao diện điều khiển do người dùng nhập vào.

Bài 11:

Câu hỏi:

Viết một chương trình chấp nhận chuỗi từ do người dùng nhập vào, phân tách nhau bởi dấu phẩy và in những từ đó thành chuỗi theo thứ tự bảng chữ cái, phân tách nhau bằng dấu phẩy.

Giả sử đầu vào được nhập là: without,hello,bag,world, thì đầu ra sẽ là: bag,hello,without,world.

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Bài 12:

Câu hỏi:

Viết một chương trình chấp nhận chuỗi là các dòng được nhập vào, chuyển các dòng này thành chữ in hoa và in ra màn hình. Giả sử đầu vào là:

Hello world  
Practice makes perfect

Thì đầu ra sẽ là:

HELLO WORLD  
PRACTICE MAKES PERFECT

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Bài 13:

Câu hỏi:

Viết một chương trình chấp nhận đầu vào là một chuỗi các từ tách biệt bởi khoảng trắng, loại bỏ các từ trùng lặp, sắp xếp theo thứ tự bảng chữ cái, rồi in chúng.

Giả sử đầu vào là: hello world and practice makes perfect and hello world again

Thì đầu ra là: again and hello makes perfect practice world

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Sử dụng set để loại bỏ dữ liệu trùng lặp tự động và dùng sorted() để sắp xếp dữ liệu.

Bài 14:

Câu hỏi:

Viết một chương trình chấp nhận đầu vào là chuỗi các số nhị phân 4 chữ số, phân tách bởi dấu phẩy, kiểm tra xem chúng có chia hết cho 5 không. Sau đó in các số chia hết cho 5 thành dãy phân tách bởi dấu phẩy.

Ví dụ đầu vào là: 0100,0011,1010,1001

Đầu ra sẽ là: 1010

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Bài 15:

Câu hỏi:

Viết một chương trình tìm tất cả các số trong đoạn 1000 và 3000 (tính cả 2 số này) sao cho tất cả các chữ số trong số đó là số chẵn. In các số tìm được thành chuỗi cách nhau bởi dấu phẩy, trên một dòng.

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Bài 16:

Câu hỏi:

Viết một chương trình chấp nhận đầu vào là một câu, đếm số chữ cái và chữ số trong câu đó. Giả sử đầu vào sau được cấp cho chương trình: hello world! 123

Thì đầu ra sẽ là:

Số chữ cái là: 10  
Số chữ số là: 3

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Bài 17:

Câu hỏi:

Viết một chương trình chấp nhận đầu vào là một câu, đếm chữ hoa, chữ thường.

Giả sử đầu vào là: Quản Trị Mạng

Thì đầu ra là:

Chữ hoa: 3

Chữ thường: 8

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Bài 18:

Câu hỏi:

Viết một chương trình tính giá trị của a+aa+aaa+aaaa với a là số được nhập vào bởi người dùng.

Giả sử a được nhập vào là 1 thì đầu ra sẽ là: 1234

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Code mẫu:

Bài 19:

Câu hỏi:

Sử dụng một danh sách để lọc các số lẻ từ danh sách được người dùng nhập vào.

Giả sử đầu vào là: 1,2,3,4,5,6,7,8,9 thì đầu ra phải là: 1,3,5,7,9

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Bài 20:

Câu hỏi:

Viết chương trình tính số tiền thực của một tài khoản ngân hàng dựa trên nhật ký giao dịch được nhập vào từ giao diện điều khiển.

Định dạng nhật ký được hiển thị như sau:

D 100  
W 200

(D là tiền gửi, W là tiền rút ra).

Giả sử đầu vào được cung cấp là:

D 300

D 300

W 200

D 100

Thì đầu ra sẽ là:

500

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

5. Bài tập Python level 3

Bài 21:

Câu hỏi:

Một website yêu cầu người dùng nhập tên người dùng và mật khẩu để đăng ký. Viết chương trình để kiểm tra tính hợp lệ của mật khẩu mà người dùng nhập vào.

Các tiêu chí kiểm tra mật khẩu bao gồm:

1. Ít nhất 1 chữ cái nằm trong [a-z]  
2. Ít nhất 1 số nằm trong [0-9]  
3. Ít nhất 1 kí tự nằm trong [A-Z]  
4. Ít nhất 1 ký tự nằm trong [$ # @]  
5. Độ dài mật khẩu tối thiểu: 6  
6. Độ dài mật khẩu tối đa: 12

Chương trình phải chấp nhận một chuỗi mật khẩu phân tách nhau bởi dấu phẩy và kiểm tra xem chúng có đáp ứng những tiêu chí trên hay không. Mật khẩu hợp lệ sẽ được in, mỗi mật khẩu cách nhau bởi dấu phẩy.

Ví dụ mật khẩu nhập vào chương trình là: ABd1234@1,a F1#,2w3E\*,2We3345

Thì đầu ra sẽ là: ABd1234@1

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Bài 22:

Câu hỏi:

Viết chương trình sắp xếp tuple (name, age, score) theo thứ tự tăng dần, name là string, age và height là number. Tuple được nhập vào bởi người dùng. Tiêu chí sắp xếp là:

Sắp xếp theo name sau đó sắp xếp theo age, sau đó sắp xếp theo score. Ưu tiên là tên > tuổi > điểm.

Nếu đầu vào là:

Tom,19,80  
John,20,90  
Jony,17,91  
Jony,17,93  
Json,21,85

Thì đầu ra sẽ là:

[('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json', '21', '85'), ('Tom', '19', '80')]

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Sử dụng itemgetter để chấp nhận nhiều key sắp xếp.

Bài 23:

Câu hỏi:

Xác định một class với generator có thể lặp lại các số nằm trong khoảng 0 và n, và chia hết cho 7.

Gợi ý:

Sử dụng yield.

Bài 24:

Câu hỏi:

Một Robot di chuyển trong mặt phẳng bắt đầu từ điểm đầu tiên (0,0). Robot có thể di chuyển theo hướng UP, DOWN, LEFT và RIGHT với những bước nhất định. Dấu di chuyển của robot được đánh hiển thị như sau:

UP 5

DOWN 3

LEFT 3

RIGHT 3

Các con số sau phía sau hướng di chuyển chính là số bước đi. Hãy viết chương trình để tính toán khoảng cách từ vị trí hiện tại đến vị trí đầu tiên, sau khi robot đã di chuyển một quãng đường. Nếu khoảng cách là một số thập phân chỉ cần in só nguyên gần nhất.

Ví dụ: Nếu tuple sau đây là input của chương trình:

UP 5  
DOWN 3  
LEFT 3  
RIGHT 2

thì đầu ra sẽ là 2.

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Bài 25:

Câu hỏi:

Viết chương trình tính tần suất các từ từ input. Output được xuất ra sau khi đã sắp xếp theo bảng chữ cái.

Giả sử input là: New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3.

Thì output phải là:

2:2  
3.:1  
3?:1  
New:1  
Python:5  
Read:1  
and:1  
between:1  
choosing:1  
or:2  
to:1

Gợi ý:

Trong trường hợp dữ liệu đầu vào được cung cấp cho câu hỏi, nó phải được giả định là một input được nhập từ giao diện điều khiển.

6. Bài tập Python khác

Bài 26:

Câu hỏi:

Định nghĩa 1 hàm có thể tính tổng hai số.

Gợi ý:

Định nghĩa 1 hàm với 2 số là đối số. Bạn có thể tính tổng trong hàm và trả về giá trị.

Bài 27:

Câu hỏi:

Định nghĩa một hàm có thể chuyển số nguyên thành chuỗi và in nó ra giao diện điều khiển

Gợi ý:

Sử dụng str() để chuyển đổi một số thành chuỗi.

Bài 28:

Câu hỏi:

Định nghĩa hàm có thể nhận hai số nguyên trong dạng chuỗi và tính tổng của chúng, sau đó in tổng ra giao diện điều khiển.

Gợi ý:

Sử dụng int() để chuyển đổi một chuỗi thành số nguyên.

Bài 29:

Câu hỏi:

Định nghĩa hàm có thể nhận 2 chuỗi từ input và nối chúng sau đó in ra giao diện điều khiển

Gợi ý:

Sử dụng + để nối các chuỗi.

Bài 30:

Câu hỏi:

Định nghĩa một hàm có input là 2 chuỗi và in chuỗi có độ dài lớn hơn trong giao diện điều khiển. Nếu 2 chuỗi có chiều dài như nhau thì in tất cả các chuỗi theo dòng.

Gợi ý:

Sử dụng hàm len() để lấy chiều dài của một chuỗi

Bài 31:

Câu hỏi:

Định nghĩa hàm có thể chấp nhận input là số nguyên và in "Đây là một số chẵn" nếu nó chẵn và in "Đây là một số lẻ" nếu là số lẻ.

Gợi ý:

Sử dụng toán tử % để kiểm tra xem số đó chẵn hay lẻ.

Bài 32:

Câu hỏi:

Định nghĩa một hàm có thể in dictionary chứa key là các số từ 1 đến 3 (bao gồm cả hai số) và các giá trị bình phương của chúng.

Gợi ý:

Sử dụng dict[key]=value để nhập mục vào dictionary.

Sử dụng toán từ \*\* để lấy bình phương của một số.

Bài 33:

Câu hỏi:

Định nghĩa một hàm có thể in dictionary chứa các key là số từ 1 đến 20 (bao gồm cả 1 và 20) và các giá trị bình phương của chúng.

Gợi ý:

Sử dụng dict[key]=value để nhập mục vào dictionary.

Sử dụng toán từ \*\* để lấy bình phương của một số.

Sử dujnng range() cho các vòng lặp.

Bài 34:

Câu hỏi:

Định nghĩa một hàm có thể tạo dictionary, chứa các key là số từ 1 đến 20 (bao gồm cả 1 và 20) và các giá trị bình phương của chúng. Hàm chỉ in các giá trị mà thôi.

Gợi ý:

Sử dụng dict[key]=value để nhập mục vào dictionary.

Sử dụng toán từ \*\* để lấy bình phương của một số.

Sử dụng range() cho các vòng lặp.

Sử dụng keys() để di lặp các key trong dictionary. Có thể sử dụng item() để nhận cặp key/value.

Bài 35:

Câu hỏi:

Định nghĩa một hàm có thể tạo ra một dictionary chứa key là những số từ 1 đến 20 (bao gồm cả 1 và 20) và các giá trị bình phương của key. Hàm chỉ cần in các key.

Gợi ý:

Tương tự như bài 34.

Bài 36:

Câu hỏi:

Định nghĩa một hàm có thể tạo và in list chứa các giá trị bình phương của các số từ 1 đến 20 (tính cả 1 và 20).

Gợi ý:

Sử dụng toán tử \*\* để lấy giá trị bình phương.

Sử dụng range() cho vòng lặp.

Sử dụng list.append() để thêm giá trị vào list.

Bài 37:

Câu hỏi:

Định nghĩa một hàm có thể tạo list chứa các giá trị bình phương của các số từ 1 đến 20 (bao gồm cả 1 và 20) và in 5 mục đầu tiên trong list.

Gợi ý:

Sử dụng toán tử \*\* để lấy giá trị bình phương.

Sử dụng range() cho vòng lặp.

Sử dụng list.append() để thêm giá trị vào list.

Sử dụng [n1:n2] để cắt list

Bài 38:

Câu hỏi:

Định nghĩa một hàm có thể tạo ra list chứa các giá trị bình phương của các số từ 1 đến 20 (bao gồm cả 1 và 20), rồi in 5 mục cuối cùng trong list.

Gợi ý:

Tương tự bài 37.

Bài 39:

Câu hỏi:

Định nghĩa một hàm có thể tạo list chứa giá trị bình phương của các số từ 1 đến 20 (bao gồm cả 1 và 20). Sau đó in tất cả các giá trị của list, trừ 5 mục đầu tiên.

Gợi ý:

Tương tư bài 37, 38.

Kết quả:

[36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400]

Bài 40:

Câu hỏi:

Định nghĩa 1 hàm có thể tạo và in một tuple chứa các giá trị bình phương của các số từ 1 đến 20 (tính cả 1 và 20).

Gợi ý:

Sử dụng toán tử \*\* để lấy giá trị bình phương.

Sử dụng range() cho vòng lặp.

Sử dụng list.append() để thêm giá trị vào list.

Sử dụng tuple() để lấy giá tuple từ list.

Kết quả:

(1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400)

Bài 41:

Câu hỏi:

Với tuple (1,2,3,4,5,6,7,8,9,10) cho trước, viết một chương trình in một nửa số giá trị đầu tiên trong 1 dòng và 1 nửa số giá trị cuối trong 1 dòng.

Gợi ý:

Sử dụng [n1:n2] để lấy một phần từ tuple.

Kết quả:

(1, 2, 3, 4, 5)  
(6, 7, 8, 9, 10)

Bài 42:

Câu hỏi:

Viết một chương trình để tạo tuple khác, chứa các giá trị là số chẵn trong tuple (1,2,3,4,5,6,7,8,9,10) cho trước.

Gợi ý:

Sử dụng for để lặp tuple.

Sử dụng tuple() để tạo tuple từ list.

Kết quả:

(2, 4, 6, 8, 10)

Bài 43:

Câu hỏi:

Viết một chương trình để tạo ra và in tuple chứa các số chẵn được lấy từ tuple (1,2,3,4,5,6,7,8,9,10).

Gợi ý:

Sử dụng "for" để lặp lại tuple.

Sử dụng tuple() để tạo ra một tuple từ một danh sách.

Kết quả:

(2, 4, 6, 8, 10)

Bài 44:

Yêu cầu:

Viết một chương trình Python nhận chuỗi nhập vào bởi người dùng, in "Yes" nếu chuỗi là "yes" hoặc "YES" hoặc "Yes", nếu không in "No".

Gợi ý:

Sử dụng lệnh if để kiểm tra điều kiện.

Bài 45:

Yêu cầu:

Viết chương trình Python có thể lọc các số chẵn trong danh sách sử dụng hàm filter. Danh sách là [1,2,3,4,5,6,7,8,9,10].

Gợi ý:

Sử dụng filter() để lọc các yếu tố trong một list.

Sử dụng lambda để định nghĩa hàm chưa biết.

Kết quả:

[2, 4, 6, 8, 10]

Lưu ý: Trong các phiên bản Python trước, bạn chỉ cần dùng hàm filter sẽ được trả kết quả đầu ra là một danh sách. Nhưng từ Python 3, phải dùng list(filter()) thì kết quả trả về mới là list. Điều này cũng áp dụng với map().

Bài 46:

Yêu cầu:

Viết chương trình Python dùng map() để tạo list chứa các giá trị bình phương của các số trong [1,2,3,4,5,6,7,8,9,10].

Gợi ý:

Sử dụng map() để tạo list.

Sử dụng lambda để định nghĩa hàm chưa biết.

Kết quả:

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

Bài 47:

Yêu cầu:

Viết chương trình Python dùng map() và filter() để tạo list chứa giá trị bình phương của các số chẵn trong [1,2,3,4,5,6,7,8,9,10].

Gợi ý:

Dùng map() để tạo list.

Dùng filter() để lọc thành phần trong list.

Dùng lambda để định nghĩa hàm chưa biết.

Kết quả:

[4, 16, 36, 64, 100]

Bài 48:

Yêu cầu:

Viết chương trình Python dùng filter() để tạo danh sách chứa các số chẵn trong đoạn [1,20].

Gợi ý:

Giống bài 45.

Kết quả:

[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

Bài 49:

Yêu cầu:

Viết chương trình Python sử dụng map() để tạo list chứa giá trị bình phương của các số trong đoạn [1,20].

Gợi ý:

Giống bài 46.

Kết quả:

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400]

Bài 50:

Yêu cầu:

Định nghĩa một class có tên là Vietnam, với static method là printNationality.

Gợi ý:

Sử dụng @staticmethod để định nghĩa class với static method.

Bài 51:

Yêu cầu:

Định nghĩa một class tên Vietnam và class con của nó là Hanoi.

Gợi ý:

Sử dụng Subclass(ParentClass) để định nghĩa một class con.

Bài 52:

Yêu cầu:

Định nghĩa một class có tên là Circle có thể được xây dựng từ bán kính. Circle có một method có thể tính diện tích.

Gợi ý:

Sử dụng def methodName(self) để định nghĩa method.

Bài 53:

Yêu cầu:

Định nghĩa class có tên là Hinhchunhat được xây dựng bằng chiều dài và chiều rộng. Class Hinhchunhat có method để tính diện tích.

Gợi ý:

Như bài 52.

Trong code trên chiều dài hình nhữ nhật là 10, chiều rộng là 2. Chạy code ta được kết quả là 20.

Bài 54:

Yêu cầu:

Định nghĩa một class có tên là Shape và class con là Square. Square có hàm init để lấy đối số là chiều dài. Cả 2 class đều có hàm area để in diện tích của hình, diện tích mặc định của Shape là 0.

Gợi ý:

Để ghi đè một method trong super class, chúng ta có thể định nghĩa một method có cùng tên trong super class.

Với chiều dài là 3, khi chạy code trên ta được kết quả là 9.

Bài 55:

Yêu cầu:

Đưa ra một RuntimeError exception.

Gợi ý:

Sử dụng raise() để đưa ra exception.

Code mẫu:

Bài 56:

Yêu cầu:

Viết hàm để tính 5/0 và sử dụng try/exception để bắt lỗi.

Gợi ý:

Sử dụng try/exception để bắt lỗi.

Kết quả khi chạy code trên ta nhận được như sau:

Bài 57:

Yêu cầu:

Định nghĩa một class exception tùy chỉnh, nhận một thông báo là thuộc tính.

Gợi ý:

Để định nghĩa một class exception tùy chỉnh, chúng ta phải định nghĩa một class kế thừa từ Exception.

Bài 58:

Yêu cầu:

Giả sử rằng chúng ta có vài địa chỉ email dạng username@companyname.com, hãy viết một chương trình để in username của địa chỉ email cụ thể. Cả username và companyname chỉ bao gồm chữ cái.

Ví dụ: Nếu cung cấp địa chỉ email QTM@quantrimang.com thì đầu ra sẽ là: QTM.

Trong trường hợp dữ liệu đầu vào không có sẵn, ta giả định nó được người dùng nhập vào từ giao diện điều khiển.

Gợi ý:

Sử dụng \w để kiểm tra chữ cái.

Bài 59:

Yêu cầu:

Tương tự như bài 58, nhưng lần này ta sẽ viết hàm để lấy companyname.

Gợi ý:

Giống bài 58.

Bài 60:

Yêu cầu:

Viết một chương trình chấp nhận chuỗi từ được phân tách bằng khoảng trống và in các từ chỉ gồm chữ số.

Ví du: Nếu những từ sau đây là đầu vào của chương trình: 3 quantrimang.com và 2 python. Đầu ra sẽ là ['3', '2']

Gợi ý:

Sử dụng re.findall() để tìm tất cả chuỗi con sử dụng regex (biểu thức tiêu chuẩn).

Bài 61:

Yêu cầu:

In chuỗi Unicode "Hello world".

Gợi ý:

Sử dụng định dạng u'string' để định nghĩa chuỗi Unicode.

Bài 62:

Yêu cầu:

Viết chương trình để đọc chuỗi ASCII và chuyển đổi nó sang một chuỗi Unicode được mã hóa bằng UTF-8.

Gợi ý:

Sử dụng hàm encode() để chuyển đổi.

Bài 63:

Yêu cầu:

Viết comment đặc biệt để chỉ định file code nguồn Python ở Unicode.

Bài 64:

Yêu cầu:

Viết một chương trình tính 1/2 + 2/3 + 3/4 + ... + n/(n + 1) với một n là số được nhập vào (n> 0).

Ví dụ, nếu n là số sau đây được nhập vào:

5

Thì đầu ra phải là:

3.55

Gợi ý:

Sử dụng float() để chuyển số nguyên sang số thập phân.

Bài 65:

Yêu cầu:

Viết chương trình tính:  f(n)=f(n-1)+100 khi n>0 và f(0)=1, với n là số được nhập vào (n>0).

Ví dụ: Nếu n được nhập vào là 5 thì đầu ra phải là 500.

Gợi ý:

Chúng ta có thể định nghĩa [hàm đệ quy trong Python](https://quantrimang.com/ham-de-quy-trong-python-145807).

Bài 66:

Yêu cầu:

Dãy Fibonacci được tính dựa trên công thức sau:

f(n)=0 nếu n=0

f(n)=1 nếu n=1

f(n)=f(n-1)+f(n-2) nếu n>1

Hãy viết chương trình tính giá trị của f(n) với n là số được người dùng nhập vào. Ví dụ: Nếu n được nhập vào là 7 thì đầu ra của chương trình sẽ là 13.

Gợi ý:

Tương tự như bài 65, ta cũng sử dụng hàm đệ quy trong Python.

Bài 67:

Yêu cầu:

Dãy Fibonacci được tính dựa trên công thức sau:

f(n)=0 nếu n=0

f(n)=1 nếu n=1

f(n)=f(n-1)+f(n-2) nếu n>1

Hãy viết chương trình sử dụng list comprehension để in dãy Fibonacci dưới dạng tách biệt bằng dấu ",", n được người dùng nhập vào.

Ví dụ: Nếu n được nhập vào là 7 thì đầu ra của chương trình sẽ là: 0,1,1,2,3,5,8,13

Gợi ý:

Chúng ta có thể định nghĩa hàm đệ quy trong Python.

Sử dụng list comprehension để tạo ra list từ list hiện có.

Sử dụng string.join() để nối danh sách các chuỗi.

Bài 68:

Yêu cầu:

Viết chương trình sử dụng generator để in số chẵn trong khoảng từ 0 đến n, cách nhau bởi dấu phẩy, n là số được nhập vào.

Ví dụ nếu n=10 được nhập vào thì đầu ra của chương trình là: 0,2,4,6,8,10

Gợi ý:

Sử dụng yield để tạo ra giá trị kết tiếp trong generator.

Bài 69:

Yêu cầu:

Viết chương trình sử dụng generator để in số chia hết cho 5 và 7 giữa 0 và n, cách nhau bằng dấu phẩy, n được người dùng nhập vào.

Ví dụ: Nếu n=100 được nhập vào thì đầu ra của chương trình là: 0,35,70.

Gợi ý:

Như bài 68.

Bài 70:

Yêu cầu:

Viết các lệnh assert để xác minh rằng tất cả các số trong list [2,4,6,8] là chẵn.

Gợi ý:

Sử dụng assert để khẳng định.

Bài 71:

Yêu cầu:

Viết chương trình chấp nhận biểu thức toán học cơ bản do người dùng nhập vào từ bảng điều khiển và in kết quả ước lượng ra ngoài màn hình.

Ví dụ: Nếu chuỗi sau là đầu vào của chương trình:

35 + 3

thì đầu ra sẽ lả:

38

Gợi ý:

Sử dụng eval() để ước lượng biểu thức

Bài 72:

Yêu cầu:

Viết hàm tìm kiếm nhị phân để tìm các item trong một list đã được sắp xếp. Hàm sẽ trả lại chỉ số của phần tử được tìm thấy trong list.

Gợi ý:

Sử dụng[if/elif](https://quantrimang.com/lenh-if-ifelse-trong-python-141111) để giải quyết các điều kiện.

Bài 73:

Yêu cầu:

Tạo một số thập phân ngẫu nhiên, có giá trị nằm trong khoảng từ 10 đến 100 bằng cách sử dụng module math của Python.

Gợi ý:

Sử dụng random.random() để tạo float ngẫu nhiên trong [0,1].

Bài 74:

Yêu cầu:

Tạo một số thập phân ngẫu nhiên, có giá trị nằm trong khoảng 5 đến 95, sử dụng module math của Python.

Gợi ý:

Giống bài 73.

Bài 75:

Yêu cầu:

Viết chương trình xuất ra một số chẵn ngẫu nhiên trong khoảng 0 đến 10 (bao gồm cả 0 và 10), sử dụng module random và list comprehension.

Gợi ý:

Sử dụng random.choice() để tạo một phần tử ngẫu nhiên từ list.

Bài 76:

Yêu cầu:

Vui lòng viết chương trình để xuất một số ngẫu nhiên, chia hết cho 5 và 7, từ 0 đến 200 (gồm cả 0 và 200), sử dụng module random và list comprehension.

Gợi ý:

Giống bài 75.

Bài 77:

Yêu cầu:

Vui lòng viết chương trình để tạo một list với 5 số ngẫu nhiên từ 100 đến 200.

Gợi ý:

Sử dụng random.sample() để tạo list chứa các giá trị ngẫu nhiên.

Bài 78:

Yêu cầu:

Viết chương trình tạo ngẫu nhiên list gồm 5 số chẵn nằm trong đoạn [100;200].

Gợi ý:

Giống bài 77.

Bài 79:

Yêu cầu:

Viết chương trình để tạo ngẫu nhiên một list gồm 5 số, chia hết cho 5 và 7, nằm trong đoạn [1;1000].

Gợi ý:

Giống bài 77, 78.

Bài 80:

Yêu cầu:

Viết chương trình để in một số nguyên ngẫu nhiên từ 7 đến 15.

Gợi ý:

Sử dụng random.randrange() để lấy số nguyên ngẫu nhiên trong một phạm vi nhất định.

Bài 81:

Yêu cầu:

Viết chương trình để nén và giải nén string ""hello world!hello world!hello world!hello world!".

Gợi ý:

Sử dụng zlib.compress() và zlib.decompress() để nén và giải nén string.

Bài 82:

Yêu cầu:

Bạn hãy viết một chương trình để in thời gian thực thi (running time of execution) phép tính "1+1" 100 lần.

Gợi ý:

Sử dụng timeit() để đo thời gian chạy

Bài 83:

Yêu cầu:

Viết chương trình để trộn và in list [3,6,7,8].

Gợi ý:

Sử dụng shuffle() để trộn list.

Bài 84:

Yêu cầu:

Viết một chương trình để tạo tất cả các câu có chủ ngữ nằm trong ["Anh","Em"], động từ nằm trong ["Chơi","Yêu"] và tân ngữ là ["Bóng đá","Xếp hình"].

Gợi ý:

Sử dụng list[index] để lấy phần tử từ list.

Bài 85:

Yêu cầu:

Viết chương trình in list sau khi xóa các số chẵn trong [5,6,77,45,22,12,24].

Gợi ý:

Sử dụng [list comprehension](https://quantrimang.com/cach-su-dung-list-comprehension-trong-python-154301) để xóa một loạt phần tử của list.

Yêu cầu:

Sử dụng list comprehension để viết chương trình in list sau khi đã loại bỏ các số chia hết cho 5 và 7 trong [12,24,35,70,88,120,155].

Gợi ý:

Giống bài 85.

Bài 87:

Yêu cầu:

Viết chương trình in list sau khi đã xóa số thứ 0, thứ 2, thứ 4, thứ 6 trong [12,24,35,70,88,120,155].

Gợi ý:

Sử dụng list comprehension để xóa một loạt phần tử trong list.

Sử dụng hàm enumerate() để lấy index, value của tuple.

Bài 88:

Yêu cầu:

Viết chương trình tạo mảng 3D 3\*5\*8 có mỗi phần tử là 0.

Gợi ý:

Sử dụng list comprehension để tạo mảng.

Bài 89:

Yêu cầu:

Viết chương trình in list sau khi đã xóa số ở vị trí thứ 0, thứ 5, thứ 5 trong [12,24,35,70,88,120,155].

Gợi ý:

Giống bài 87.

Bài 90:

Yêu cầu:

Viết chương trình in list sau khi đã xóa giá trị 24 trong [12,24,35,24,88,120,155].

Gợi ý:

Sử dụng phương thức xóa của list để xóa giá trị.

Bài 91:

Yêu cầu:

Với 2 list cho trước: [1,3,6,78,35,55] và [12,24,35,24,88,120,155], viết chương trình để tạo list có phần tử là giao của 2 list đã cho.

Gợi ý:

Sử dụng set() và "&=" để thiết lập điểm giao.

Bài 92:

Yêu cầu:

Viết chương trình in list từ list [12,24,35,24,88,120,155,88,120,155], sau khi đã xóa hết các giá trị trùng nhau.

Gợi ý:

Sử dụng set() để lưu trữ các giá trị không bị trùng lặp.

Bài 93:

Yêu cầu:

Định nghĩa class Nguoi và 2 class con của nó: Nam, Nu. Tất cả các class có method "getGender" có thể in "Nam" cho class Nam và "Nữ" cho class Nu.

Gợi ý:

Sử dụng Subclass(Parentclass) để định nghĩa 1 class con.

Bài 94:

Yêu cầu:

Viết chương trình đếm và in số ký tự của chuỗi do người dùng nhập vào.

Ví dụ:

Nếu chuỗi nhập vào là quantrimang.com thì đầu ra sẽ là:

q,1  
u,1  
a,2  
n,2  
t,1  
r,1  
i,1  
m,2  
g,1  
.,1  
c,1  
o,1

Gợi ý:

Sử dụng dict để lưu trữ các cặp key/value.

Sử dụng dict.get() để tra cứu key với giá trị mặc định.

Bài 95:

Yêu cầu:

Viết chương trình nhận chuỗi đầu vào từ giao diện điều khiển và in nó theo thứ tự ngược lại.

Ví dụ nếu chuỗi nhập vào là:

i love you

Thì kết quả đầu ra là:

uoy evol i

Gợi ý:

Sử dụng list[::-1] để lặp list theo thứ tự ngược lại.

Bài 96:

Yêu cầu:

Viết chương trình nhận chuỗi do người dùng nhập vào và in các ký tự có chỉ số chẵn.

Ví dụ: Nếu chuỗi sau được nhập vào: q1u2a3n4t5r6i7m8a9n4g5.6c7o8m, thì đầu ra sẽ là: quantrimang.com.

Gợi ý:

Sử dụng list[::2] để lặp list cách 2 vị trí.

Bài 97:

Yêu cầu:

Viết chương trình in tất cả các hoán vị của [1,2,3].

Gợi ý:

Sử dụng itertools.permutations() để lấy hết các hoán vị của list.

Bài 98:

Yêu cầu:

Viết chương trình để giải 1 câu đố cổ của Trung Quốc: Một trang trại thỏ và gà có 35 đầu, 94 chân, hỏi số thỏ và gà là bao nhiêu?

Gợi ý:

Sử dụng [vòng lặp for](https://quantrimang.com/vong-lap-for-trong-python-142234) để lặp qua tất cả các giả thuyết có thể.