1. Viết một chương trình 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. Ví dụ, số cho trước là 8 thì kết quả đầu ra phải là 40320.
2. Định nghĩa một hàm có đầu vào là 2 chuỗi và in chuỗi có độ dài lớn hơn ra màn hình. Nếu 2 chuỗi có chiều dài như nhau thì in tất cả các chuỗi theo từng dòng. Sử dụng hàm len() để lấy chiều dài của một chuỗi.
3. Định nghĩa một hàm có thể tạo và in một danh sách chứa các giá trị bình phương của các số từ 1 đến 20 (tính cả 1 và 20). Yêu cầu: 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 danh sách.
4. Định nghĩa một hàm có thể tạo một danh sách chứa các giá trị bình phương của các số từ 1 đến 20 (bao gồm cả 1 và 20) và in 5 phần tử đầu tiên trong danh sách. Yêu cầu: 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 lấy các phần tử của danh sách.
5. Định nghĩa một hàm có thể tạo ra một danh sách chứa các giá trị bình phương của các số từ 1 đến 20 (bao gồm cả 1 và 20), rồi in 5 phần tử cuối cùng trong list.
6. Định nghĩa một hàm có thể tạo một danh sách 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 danh sách, trừ 5 phần tử đầu tiên.
7. Viết một hàm tính giá trị bình phương của một số, sử dụng toán tử \*\*.
8. Viết 02 hàm có 3 tham số: Một hàm trả về tham số trung vị (tham số có giá trị ở giữa) của 3 tham số đó và một hàm trả giá trị trung bình cộng của 3 tham số đó. Sau đó tạo hàm chính là main() gọi 02 hàm này để đọc ba giá trị từ người dùng và hiển thị số trung vị và giá trị trung bình cộng của chúng. Thực thi hàm main().
9. Số nguyên tố là số nguyên lớn hơn 1 và chỉ chia hết cho một và chính nó. Viết hàm có 1 tham số để xác định tham số đó có phải là số nguyên tố hay không, trả về True nếu là số nguyên tố và ngược lại trả về False.

Viết chương trình chính đọc số nguyên từ người dùng và hiển thị thông báo cho biết đó có phải là số nguyên tố hay không.

1. Viết hàm không có tham số để tạo mật khẩu ngẫu nhiên. Mật khẩu phải có độ dài ngẫu nhiên từ 7 đến 10 ký tự. Mỗi ký tự phải được chọn ngẫu nhiên từ các vị trí 33 đến 126 trong bảng ASCII. Hàm sẽ trả về mật khẩu được tạo ngẫu nhiên như là kết quả duy nhất của nó. Hiển thị mật khẩu được tạo ngẫu nhiên trong chương trình chính. Chương trình chính chỉ chạy khi giải pháp của bạn chưa được nhập vào một chương trình khác.
2. Viết một hàm có một tham số password để xác định xem mật khẩu có tốt hay không. Một mật khẩu tốt là một mật khẩu dài ít nhất 8 ký tự và chứa ít nhất một chữ cái viết hoa, ít nhất một chữ cái viết thường và ít nhất một số. Hàm sẽ trả về True nếu mật khẩu là tốt, ngược thì nó sẽ trả về Fales. Chương trình có một chương trình chính đọc mật khẩu từ người dùng và hiển thị xem nó có tốt hay không. Đảm bảo rằng chương trình chính chỉ chạy khi giải pháp của bạn chưa được nhập vào một chương trình khác.
3. Một số nguyên, n, được cho là hoàn hảo khi tổng của tất cả các ước số của n bằng n. Ví dụ, 28 là một số hoàn hảo vì các ước số của nó là 1, 2, 4, 7 và 14. Tổng 1 + 2 + 4 + 7 + 14 = 28. Viết hàm xác định xem số nguyên dương có hoàn hảo hay không. Hàm chứa một tham số, nếu tham số đó là một số hoàn hảo thì hàm sẽ trả về True. Nếu không nó sẽ trả về False. Ngoài ra, hãy viết một chương trình chính sử dụng hàm này để xác định và hiển thị tất cả các số hoàn hảo trong khoảng từ 1 đến 10.000.
4. Khi viết ra một danh sách các mục bằng tiếng Anh, người ta thường phân tách các mục bằng dấu phẩy. Ngoài ra, thêm từ “and” trước mục cuối cùng, trừ khi danh sách chỉ chứa một mục. Hãy xem xét bốn danh sách sau đây:

apples

apples and oranges

apples, oranges and bananas

apples, oranges, bananas and lemons

Viết hàm lấy danh sách các chuỗi làm tham số của nó. Hàm sẽ trả về một chuỗi chứa tất cả các mục trong danh sách, được định dạng theo cách được mô tả như trên. Hàm hoạt động cho danh sách có độ dài bất kỳ. Code bao gồm một chương trình chính đọc một số mục từ người dùng, định dạng chúng bằng cách gọi hàm và sau đó hiển thị kết quả mà hàm trả về.

1. Thư viện chuẩn Python bao gồm một phương thức có tên là xác định số lần giá trị cụ thể xuất hiện trong danh sách. Trong bài tập này, bạn sẽ tạo một hàm mới có tên là CountRange. Nó sẽ xác định và trả về số lượng phần tử trong danh sách lớn hơn hoặc bằng một số giá trị tối thiểu và nhỏ hơn một số giá trị tối đa. Hàm của bạn sẽ có ba tham số: danh sách, giá trị tối thiểu và giá trị tối đa. Nó sẽ trả về một kết quả số nguyên lớn hơn hoặc bằng 0. Chương trình gồm một chương trình chính thể hiện hàm cho một số danh sách khác nhau, giá trị tối thiểu và giá trị tối đa. Đảm bảo rằng chương trình hoạt động chính xác cho cả danh sách các số nguyên và danh sách các số thực.
2. Viết một hàm trả về một danh sách chứa mọi danh sách con có thể có của danh sách. Ví dụ: danh sách con của [1, 2, 3] là [], [1], [2], [3], [1, 2], [2, 3] và [1, 2, 3]. Lưu ý rằng hàm của bạn sẽ luôn trả về một danh sách chứa trong gatleast danh sách trống vì danh sách trống là danh sách con của mọi danh sách. Bao gồm một chương trình chính thể hiện chức năng của bạn bằng cách hiển thị tất cả các danh sách phụ của một số danh sách khác nhau.
3. 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). Yêu cầu: 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.
4. Định nghĩa một class có ít nhất 2 method:

getString: để nhận một chuỗi do người dùng nhập vào.

printString: in chuỗi vừa nhập sang chữ hoa.

Ví dụ: Chuỗi nhập vào là hello python thì đầu ra phải là: HELLO PYTHON