STT	Mã bài tập	Đề bài (Hàm – đặc tả - kiểm thử)	Test case	
			Input	Output của hàm
1	W5A1	Viết hàm truyền vào hai số nguyên a, b. Trả về số lớn hơn trong hai số. def Max_Of_Two(a, b) -> int:	34 56 23 -90 234 345 1 90 13 78	56 23 345 90 78
2	W5A2	Viết hàm hoán đổi hai số nguyên. def swap(a, b) -> Tuple[int, int]:	12 -30 13 10 45 12 10 90 100 45	-30 12 10 13 12 45 90 10 45 100
3	W5A3	Viết hàm kiểm tra một số có phải là số nguyên tố hay không?	2 3 10 9 13	True True False False True
4	W5A4	Vlết hàm kiểm tra một số có phải số hoàn hảo hay không, xử lý trên dữ liệu kiểu số nguyên.	23 6 28 123 24	False True True False False
5	W5A5	Viết hàm đưa vào một list số nguyên và một số nguyên dương k. Hãy tìm và trả về vị trí của phần tử đầu tiên có giá trị k trong list số nguyên, nếu không có thì trả về -1	1 2 3 4 5 -6 7 8 19 1 2 3 4 5 5 34 25 67 89 100 1000 34 25 67 89 89 2 45 67 12 90 23 23 34 8 90 67 24 34 -45 -78 -29 -78 -199 -199 -29	5 4 6 1 3
6	W5A6	Viết hàm tính giai thừa. Nhập vào số nguyên n, trả về giai thừa của n (n!)	4 10 14 0 20	24 3628800 87178291200 1 2432902008176640000
7	W5A7	Viết hàm giả lập một máy tính bỏ túi. Hàm thực hiện tính toán kết quả của phép toán giữa số \(num1 \) và \(num2 \) với toán tử \(operat \) Biết các toán tử đơn giản thực hiện phép toán lần lượt là: Phép trừ \(- \) Phép cộng \(+ \) Phép chia \(/ \) Phép nhân \(* \) Kết quả làm tròn đến chữ số thập phân thứ hai. Đầu vào: Hai số thực và phép toán cần thực hiện (trong số các phép toán \(+, -, *, /\)). Đầu ra: Kết quả phép toán.	2 + 3 12 / 4 23.89 * 23.12 82 - 12 12 * -23.004	5.00 3.00 552.34 70.00 -276.05

8	W5A8	Khoảng cách Hamming giữa hai số nguyên là số lượng vị trí khác nhau giữa hai dãy bits tương ứng của chúng. Ví dụ: Khoảng cách Hamming giữa 1 (1) và 4 (100) là 2. Cho hai số nguyên x và y, hãy viết hàm trả về khoảng cách Hamming giữa hai số này. Khoảng cách Hamming là cái tên được đặt theo tên của Richard Hamming, người giới thiệu lý thuyết này trong tài liệu có tính cơ sở của ông về mã phát hiện lỗi và sửa lỗi (error-detecting and error-correcting codes). Nó được sử dụng trong kỹ thuật viễn thông để tính số lượng các bit trong một từ nhị phân (binary word) bị đổi ngược, như một hình thức để ước tính số lỗi xảy ra trong quá trình truyền thông, và vì thế, đôi khi, nó còn được gọi là khoảng cách tín hiệu (signal distance). Việc phân tích trọng số Hamming của các bit còn được sử dụng trong một số ngành, bao gồm lý thuyết tin học, lý thuyết mã hóa, và mật mã học. Tuy vậy, khi so sánh các dãy ký tự có chiều dài khác nhau, hay các dãy ký tự có xu hướng không chỉ bị thay thế không thôi, mà còn bị ảnh hưởng bởi dữ liệu bị lồng thêm vào, hoặc bị xóa đi, phương pháp đo lường phức tạp hơn, như khoảng cách Levenshtein (Levenshtein distance) là một phương pháp có tác dụng và thích hợp hơn.	1 4 123456 32321 11111111 11111111 87654321 21321835 75849321 45687214	2 6 0 15 18
9	W5A9	Viết hàm trả về tổng các chữ số của số nguyên dương n	1234 1000000 98765 00001 10045995	10 1 35 1 33
10	W5A10	Hai từ được gọi là đẳng cấu nếu các kí tự trong một từ có thể được ánh xạ để được từ còn lại. Ánh xạ một kí tự nghĩa là thay đổi toàn bộ các kí tự đó trong từ bởi một kí tự khác. Thứ tự của các kí tự được giữ không đổi. Hai kí tự khác nhau không thể ánh xạ đến cùng một kí tự, nhưng một kí tự có thể ánh xạ thành chính nó. Ví dụ, từ "abca" và "zbxz" là hai từ đẳng cấu vì ta có thể ánh xạ 'a' thành 'z', 'b' thành 'b' và 'c' thành 'x'. Viết hàm trả về \(true\) nếu hai từ \(a\) và \(b\) là đẳng cấu, ngược lại \(false\). Bạn có thể giả sử hai từ đầu vào có cùng độ dài.	egg add paper title foo bar show same show seem	true true false true false