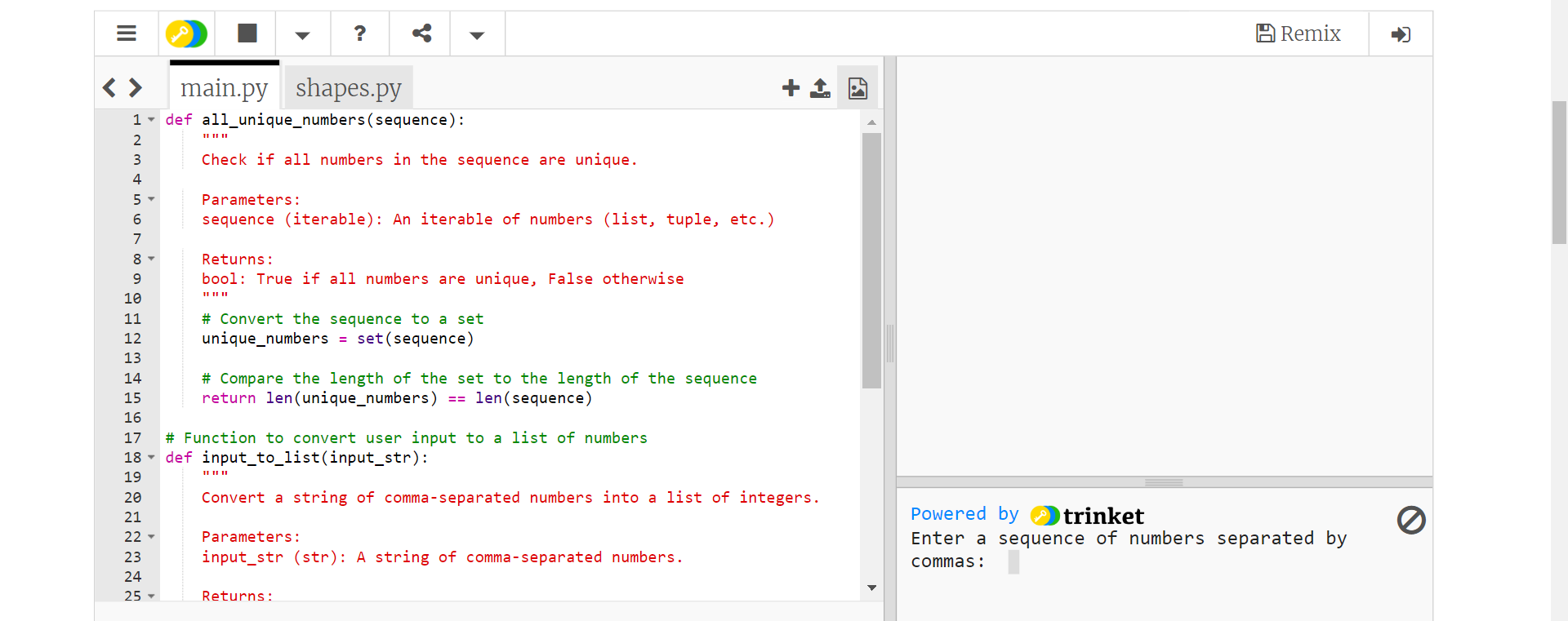
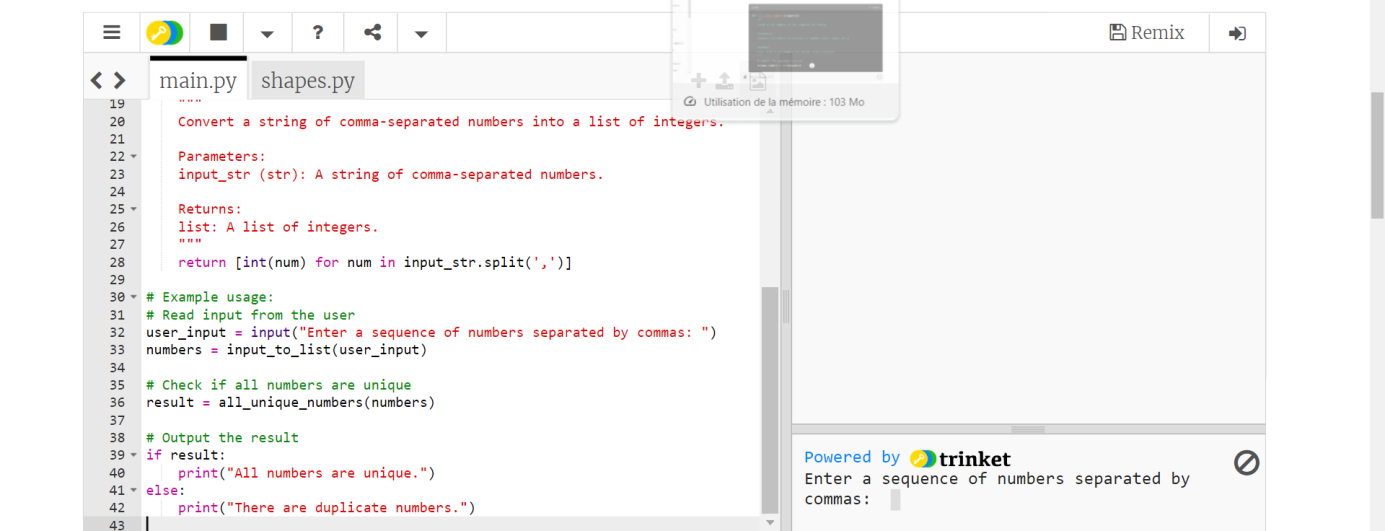
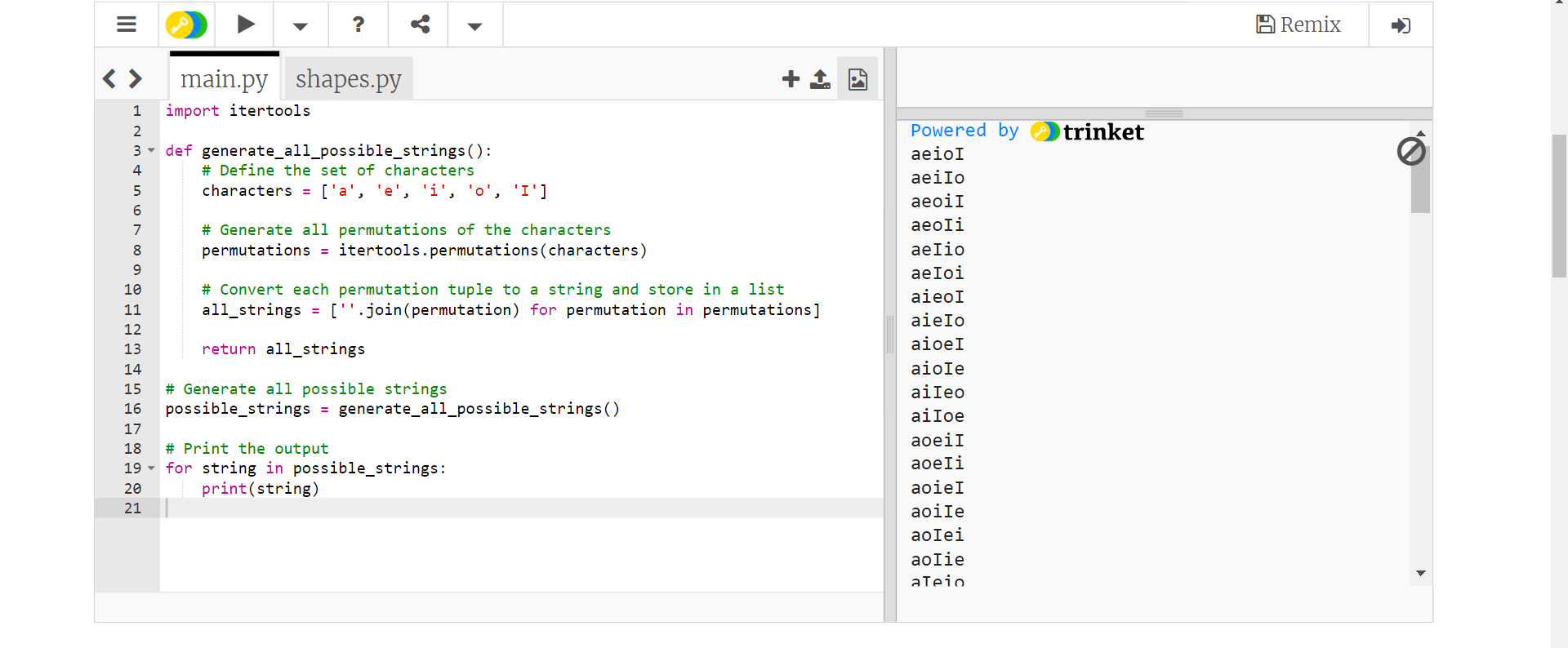
1/- Write a Python function that takes a sequence of numbers and determines whether all the numbers are different from each other.

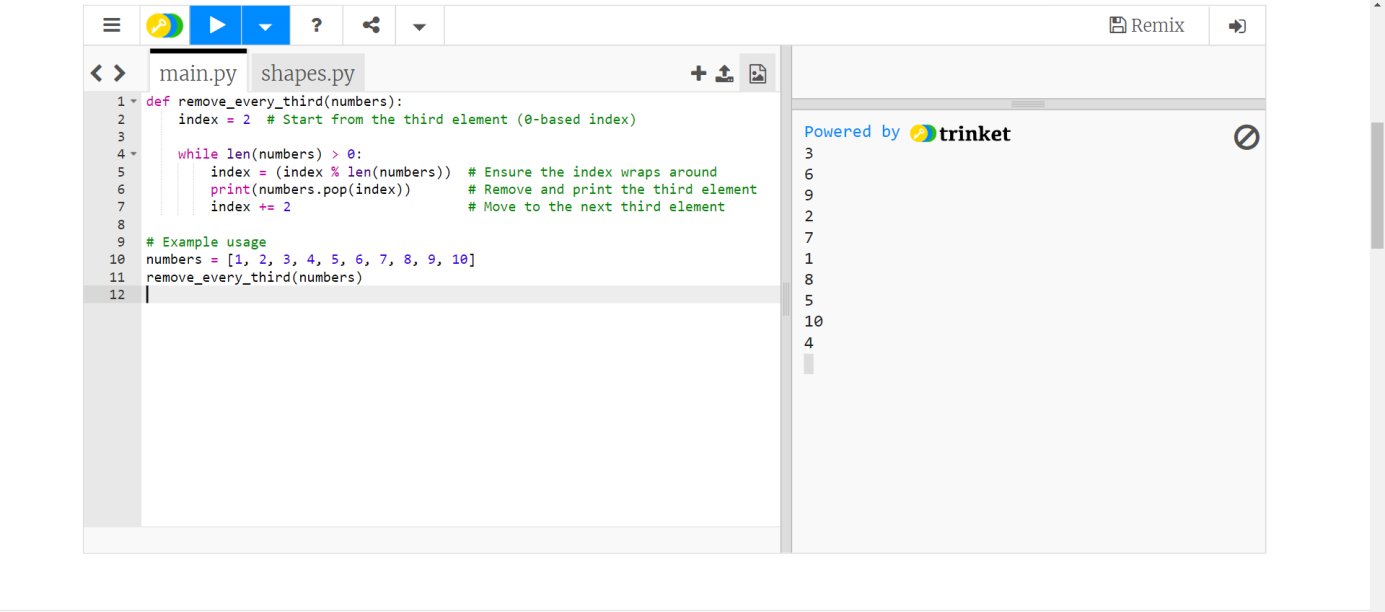




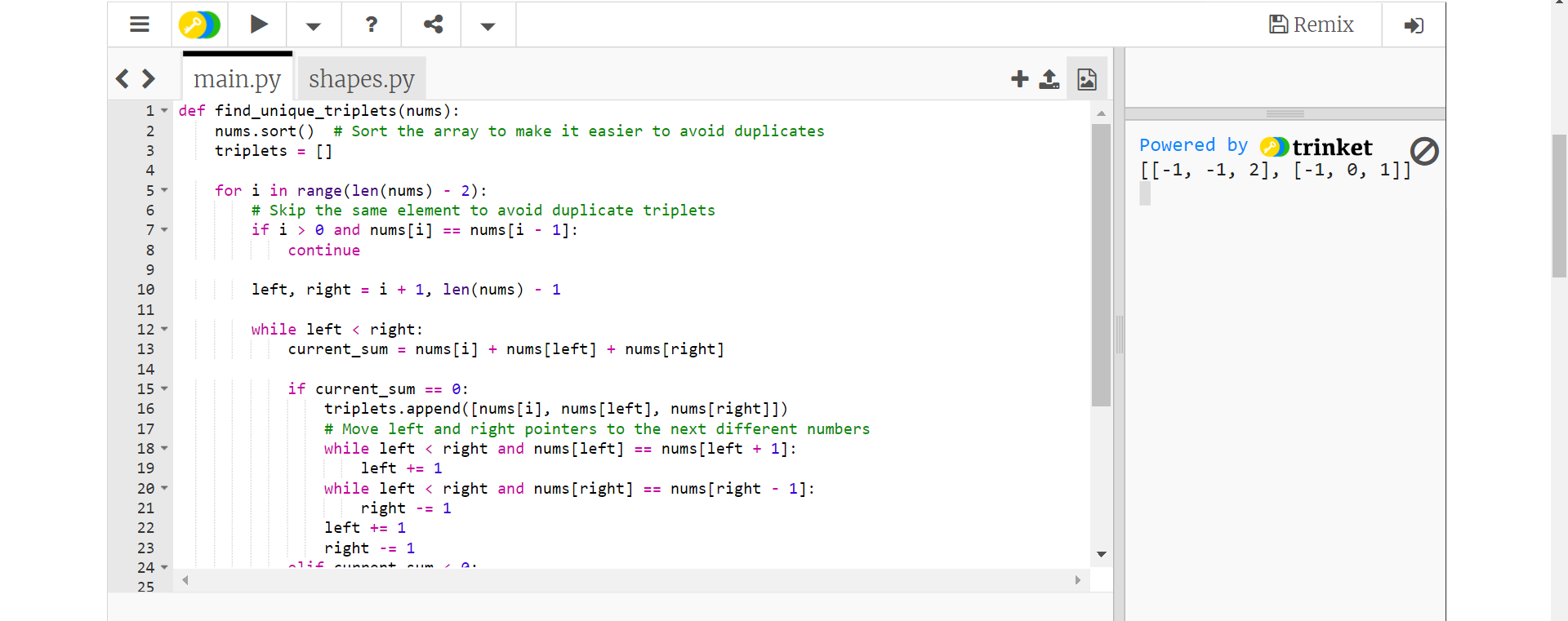
2/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program that creates all possible strings using the letters 'a', 'e', 'i', 'o', and 'I'. Ensure that each character is used only once.



3/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program that removes and prints every third number from a list of numbers until the list is empty.

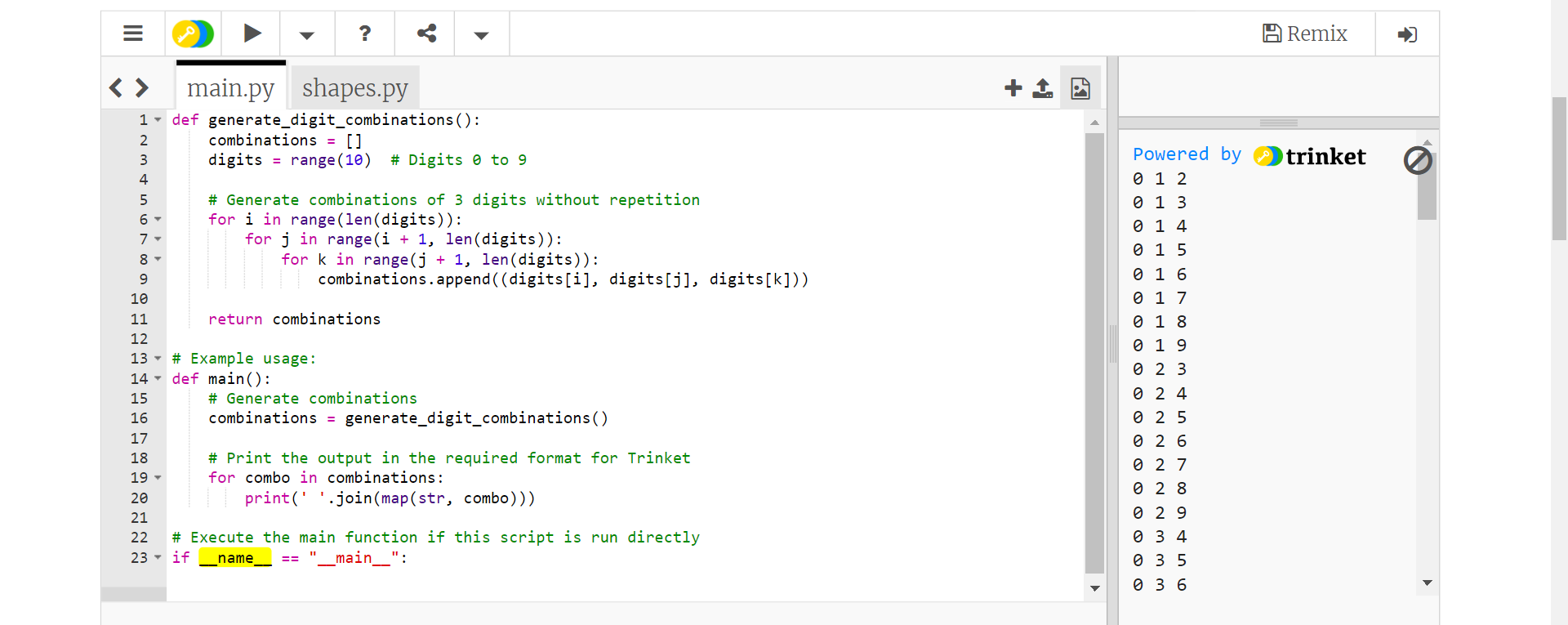


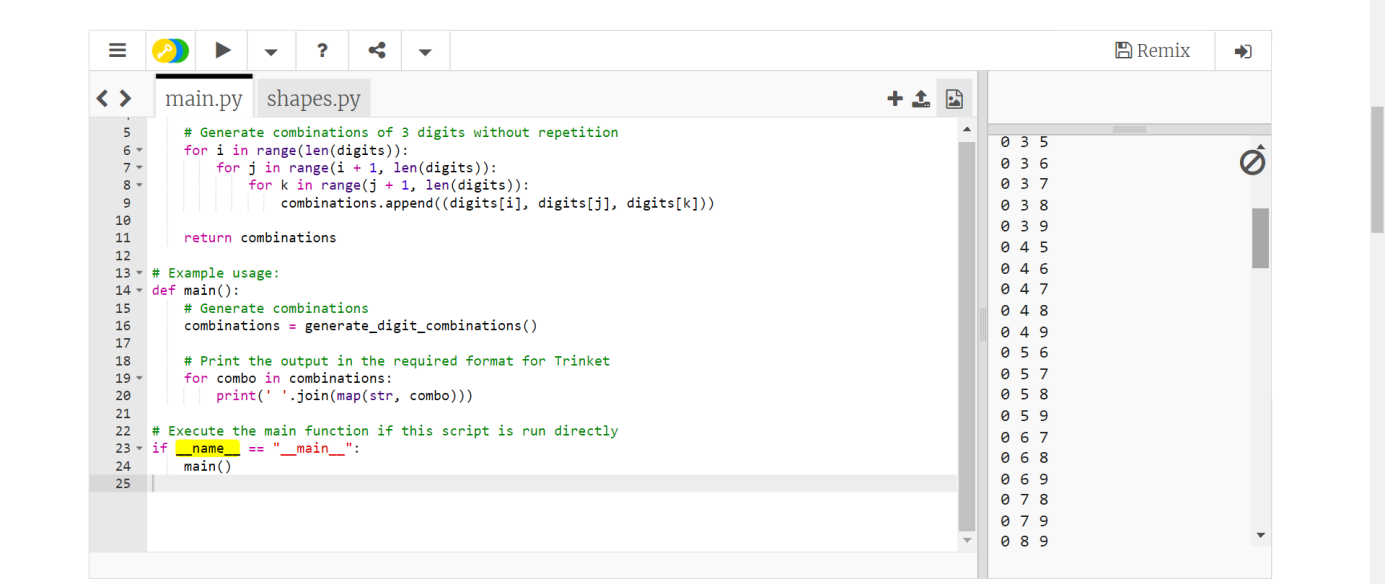
4/- Write a Python program to identify unique triplets whose three elements sum to zero from an array of n integers.

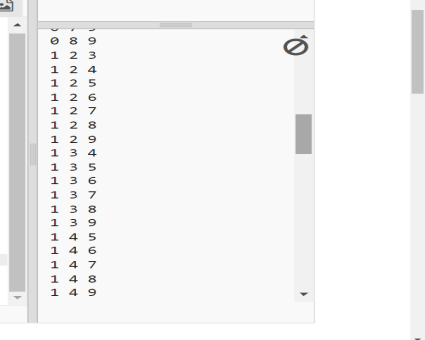


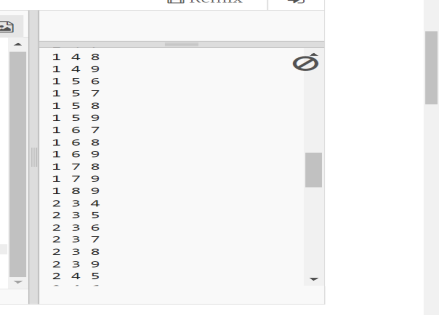


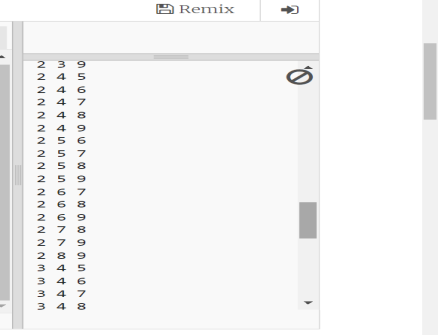
5/- Write a Python program to make combinations of 3 digits.

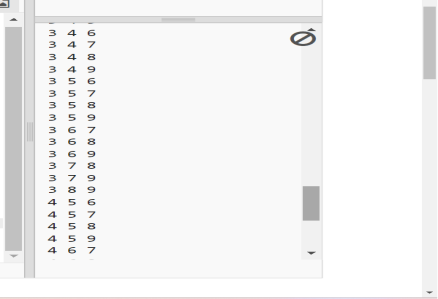








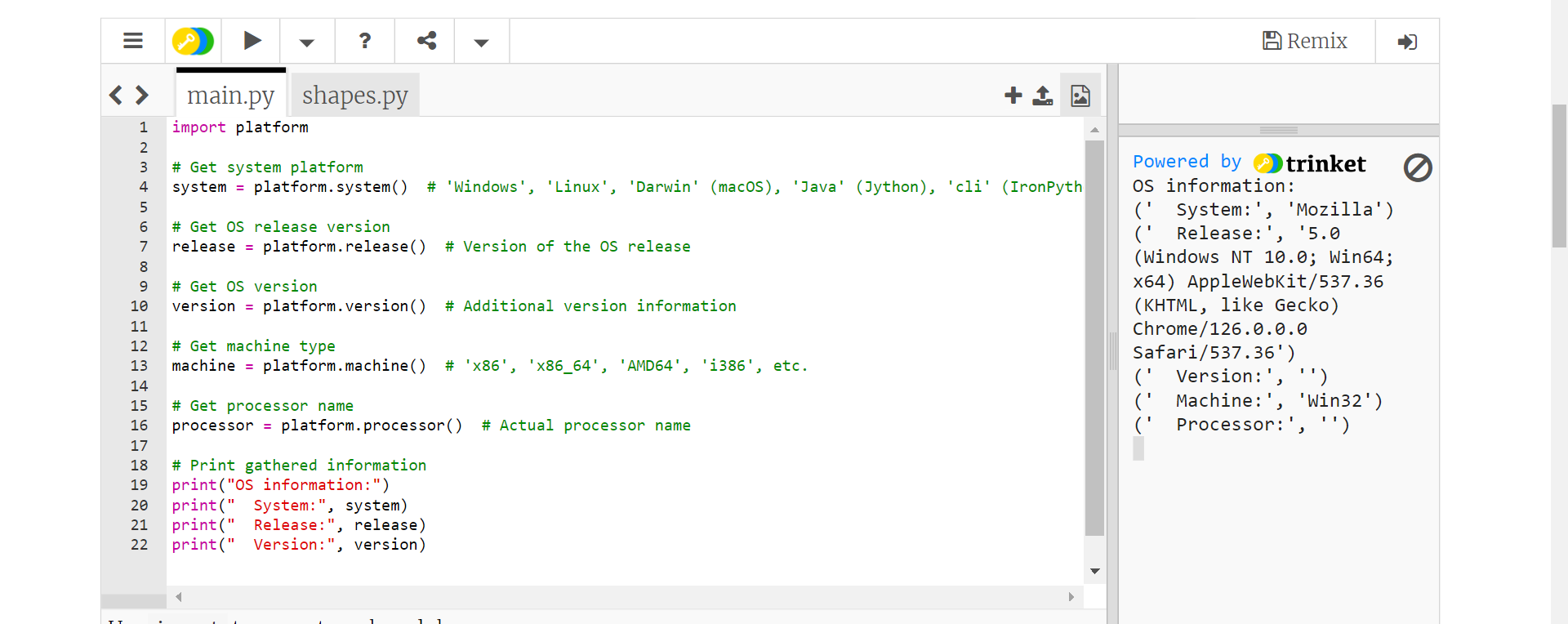


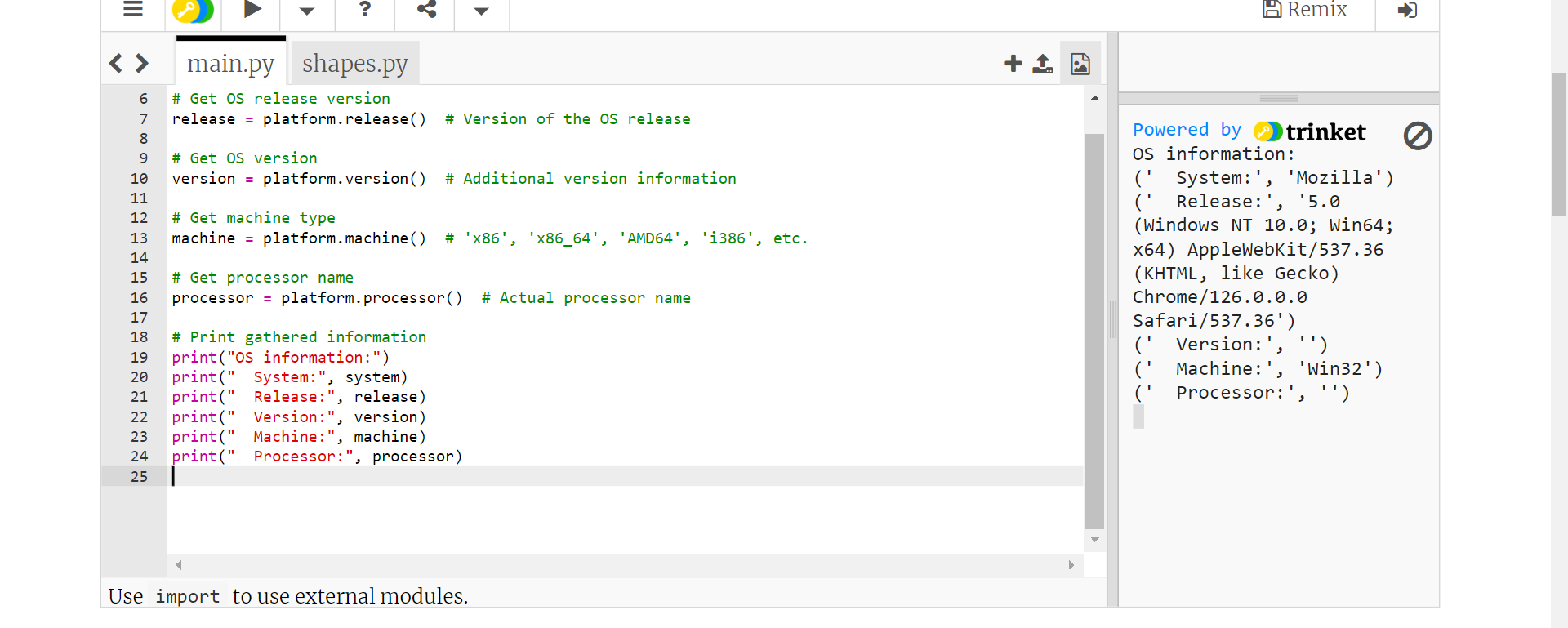


9/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to get a list of locally installed Python modules.

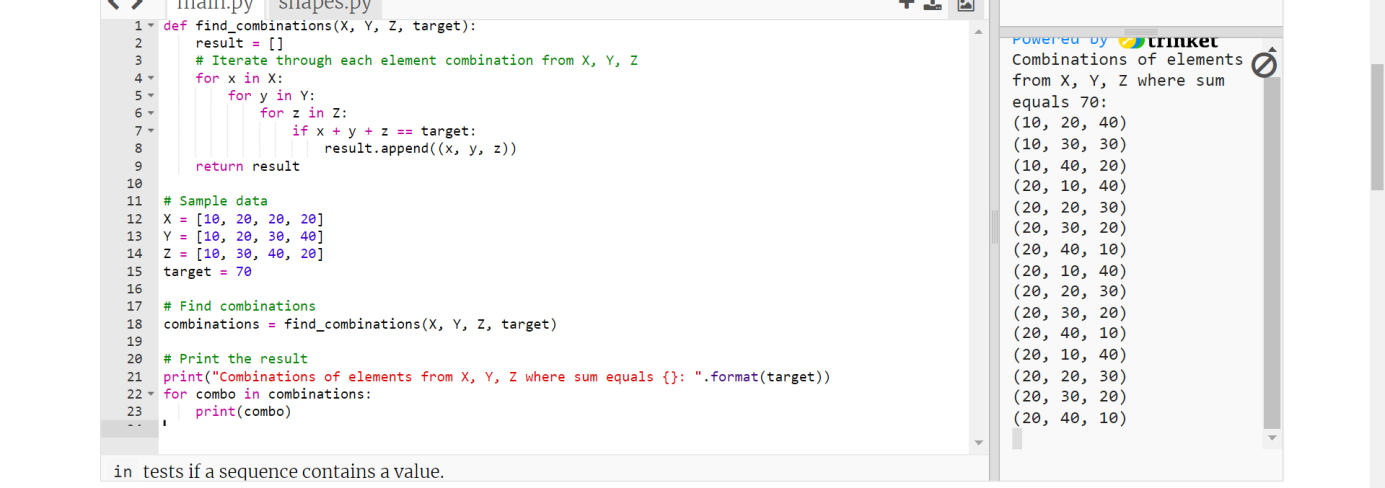


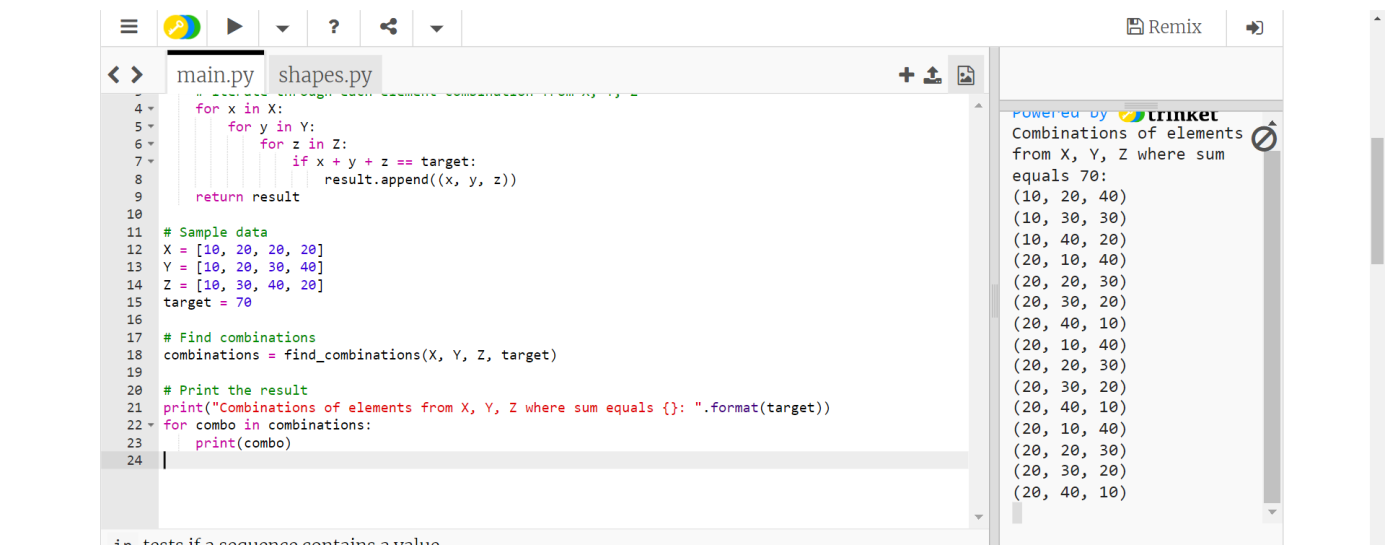
10/- Write a Python program to display some information about the OS where the [script](https://www.w3resource.com/python-exercises/basic/) is running.



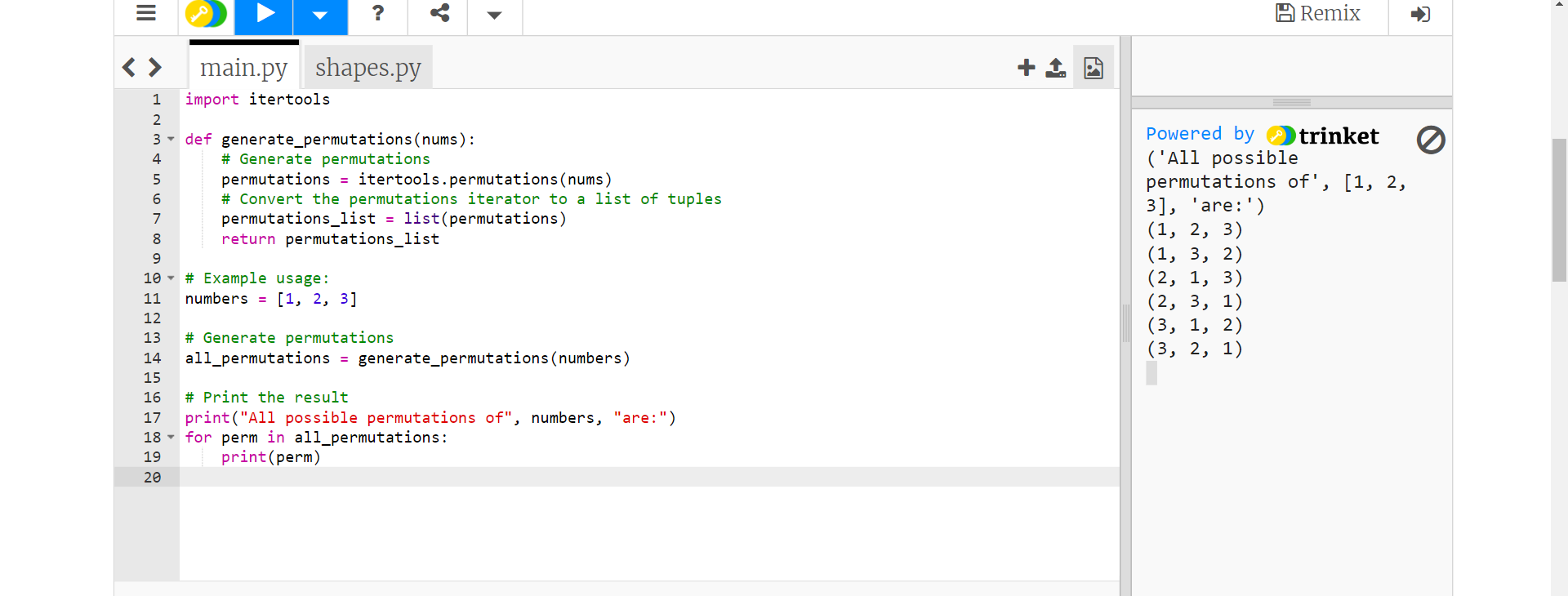


11/- Write a Python program to check the sum of three elements (each from an array) from three arrays is equal to a target value. Print all those three-element combinations.  
Sample data:  
/\*  
X = [10, 20, 20, 20]  
Y = [10, 20, 30, 40]  
Z = [10, 30, 40, 20]  
target = 70  
\*/

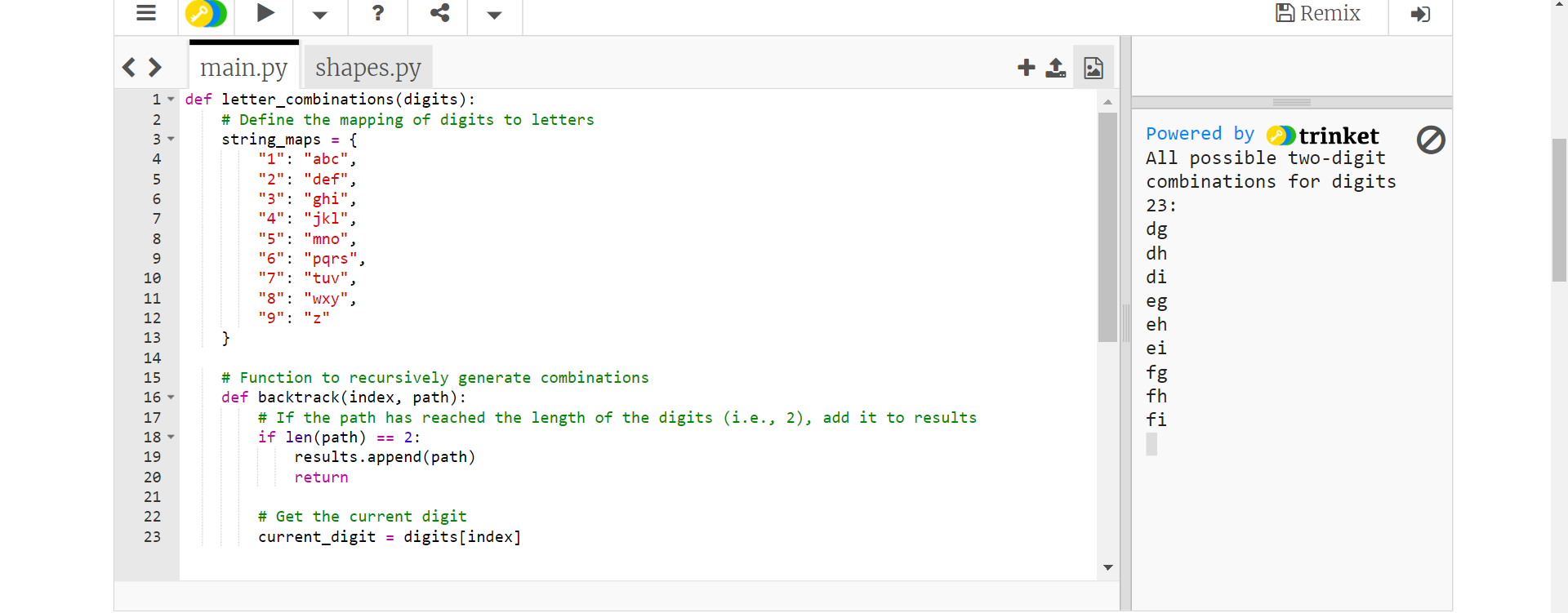


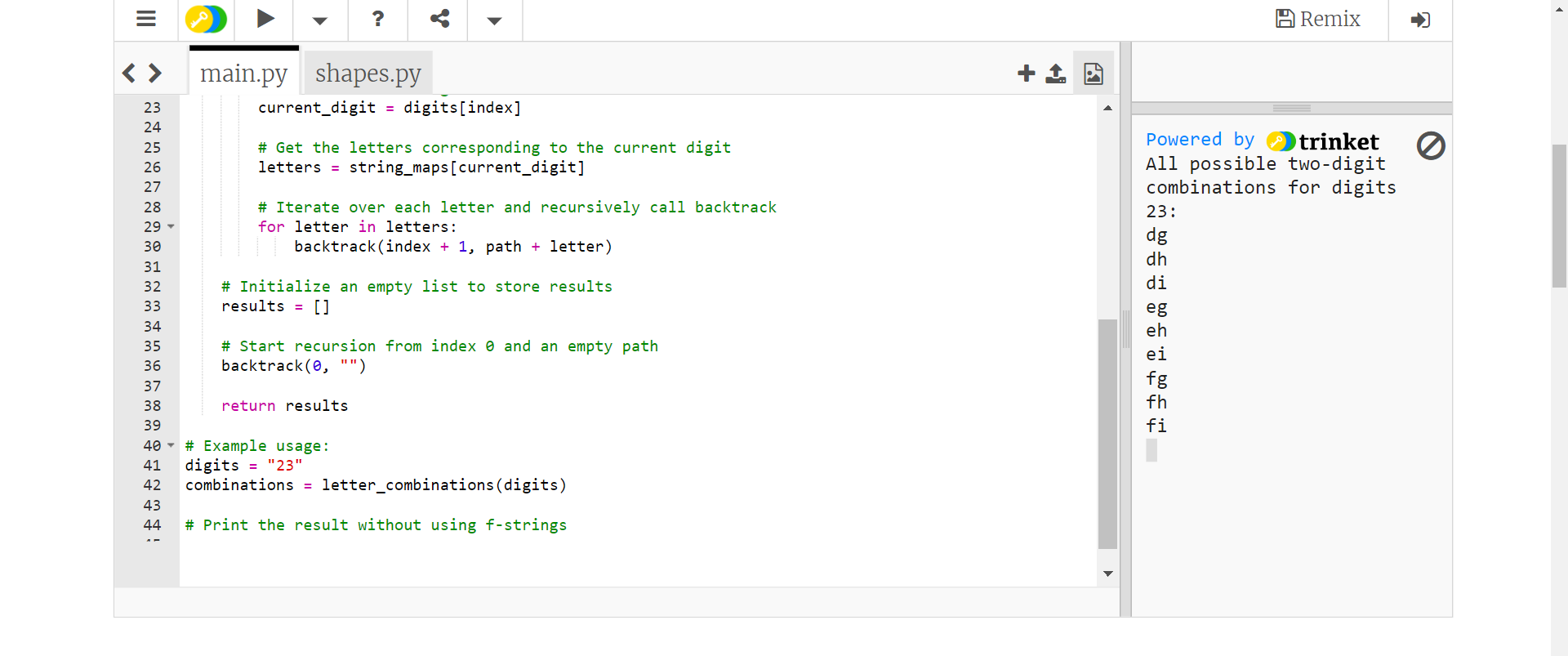


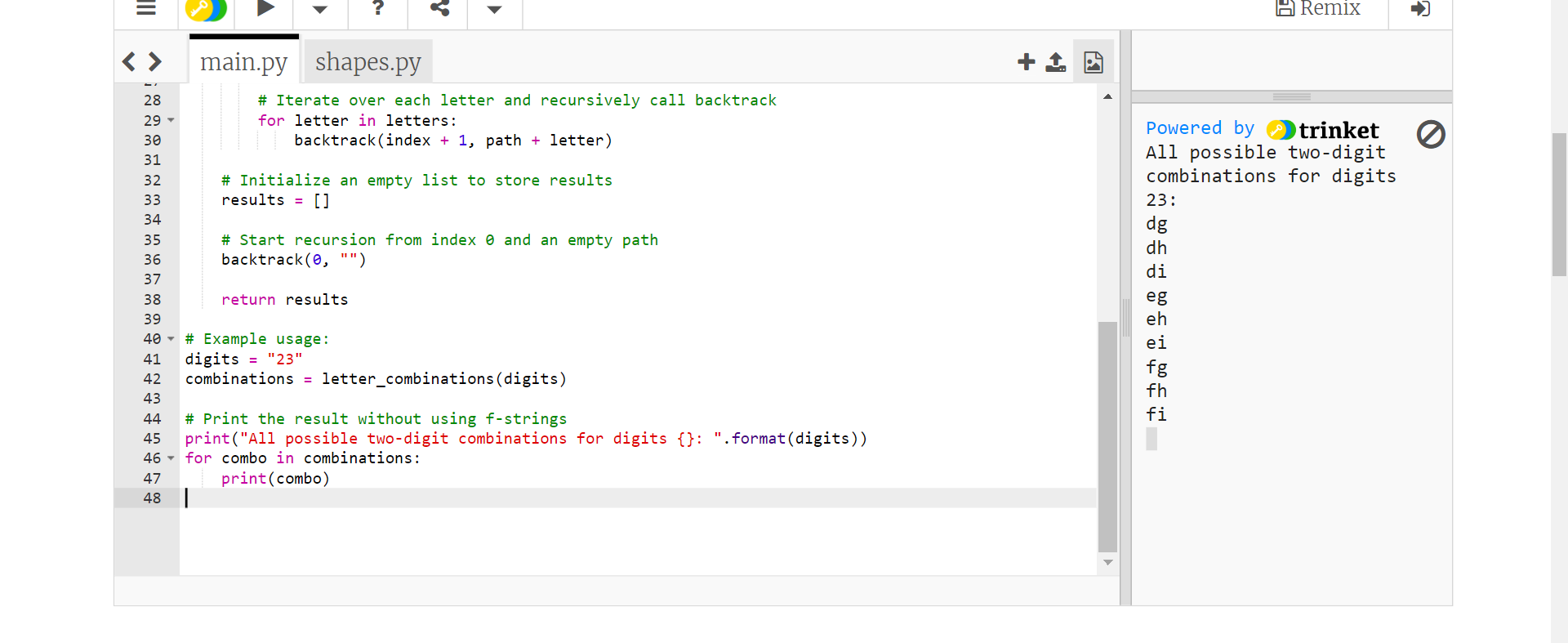
12/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program that generates a list of all possible permutations from a given collection of distinct numbers.



13/-  Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to get all possible two-digit letter combinations from a 1-9 digit string.  
string\_maps = {  
"1": "abc",  
"2": "def",  
"3": "ghi",  
"4": "jkl",  
"5": "mno",  
"6": "pqrs",  
"7": "tuv",  
"8": "wxy",  
"9": "z"  
}



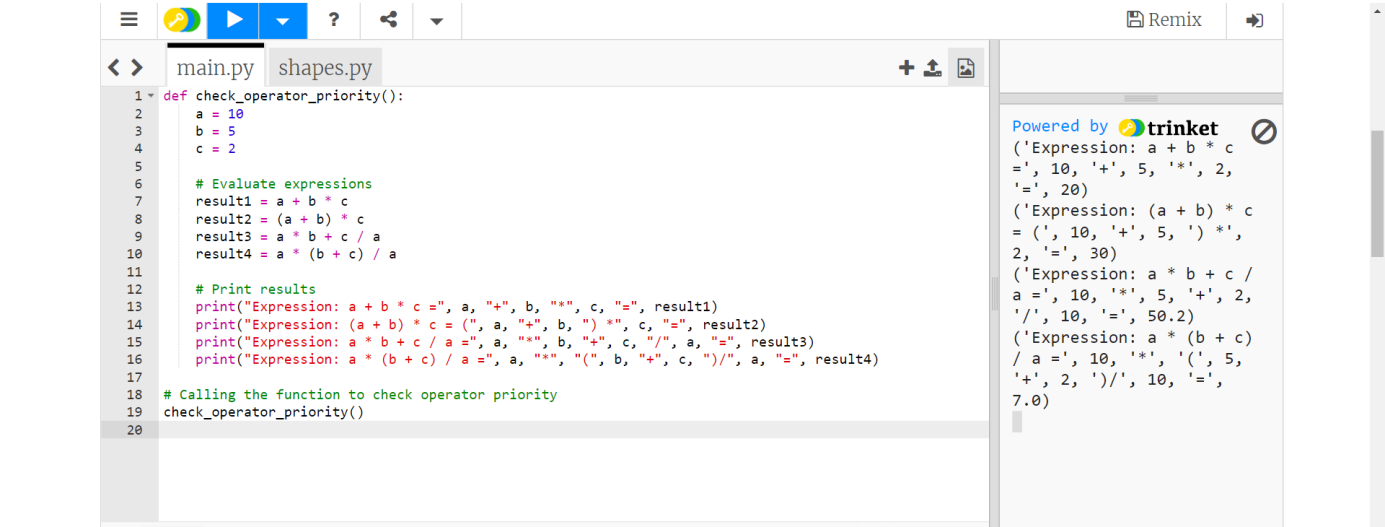


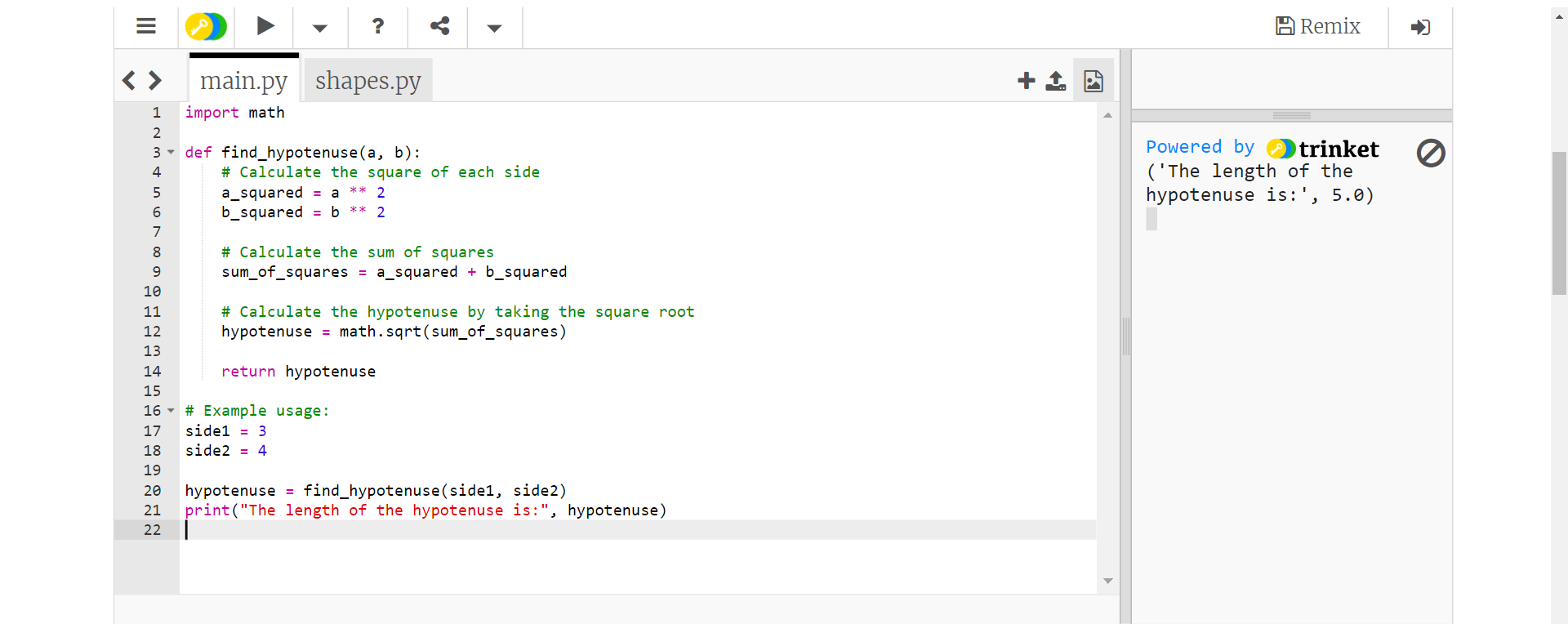


14/- Write a Python program to add two positive integers without using the '+' operator.  
Note: Use bit wise operations to add two numbers.

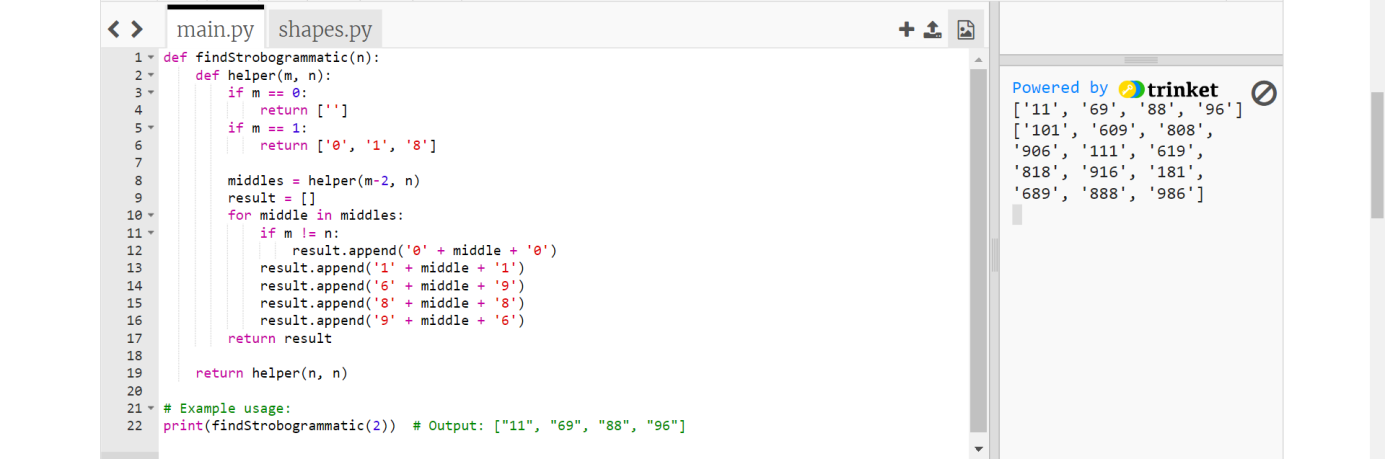


15/- Write a Python program to check the priority of the four operators (+, -, \*, /).



16/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to get the third side of a right-angled triangle from two given sides.

17/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to get all strobogrammatic numbers that are of length n.  
A strobogrammatic number is a number whose numeral is rotationally symmetric, so that it appears the same when rotated 180 degrees. In other words, the numeral looks the same right-side up and upside down (e.g., 69, 96, 1001).  
For example,  
Given n = 2, return ["11", "69", "88", "96"].  
Given n = 3, return ['818', '111', '916', '619', '808', '101', '906', '609', '888', '181', '986', '689']



20/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to find the number of zeros at the end of a factorial of a given positive number.  
Range of the number(n): (1 <= n <= 2\*109)



22/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to create a sequence where the first four members of the sequence are equal to one. Each successive term of the sequence is equal to the sum of the four previous ones. Find the Nth member of the sequence.



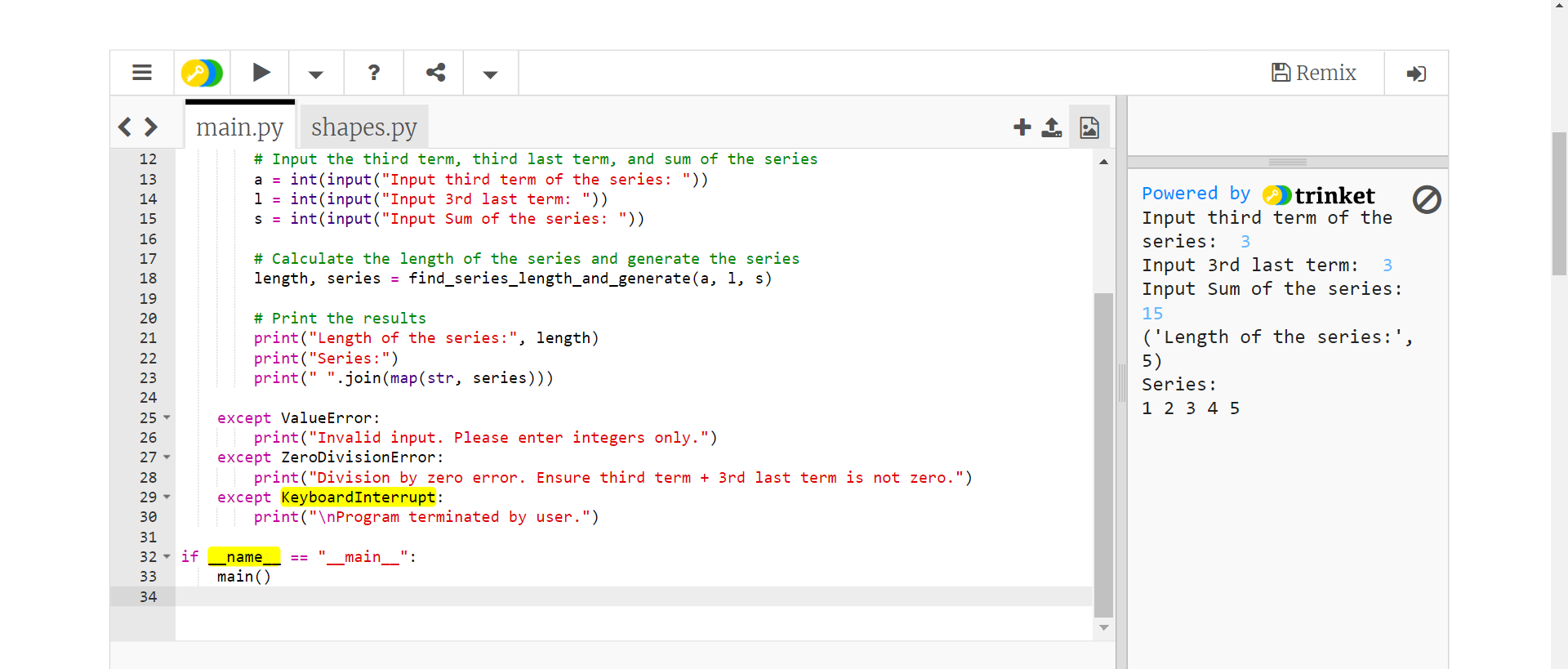
23/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program that accepts a positive number and subtracts from it the sum of its digits, and so on. Continue this operation until the number is positive.





28/-  Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to print the length of the series and the series from the given 3rd term, 3rd last term and the sum of a series.  
Sample Data:  
Input third term of the series: 3  
Input 3rd last term: 3  
Input Sum of the series: 15  
Length of the series: 5  
Series:  
1 2 3 4 5



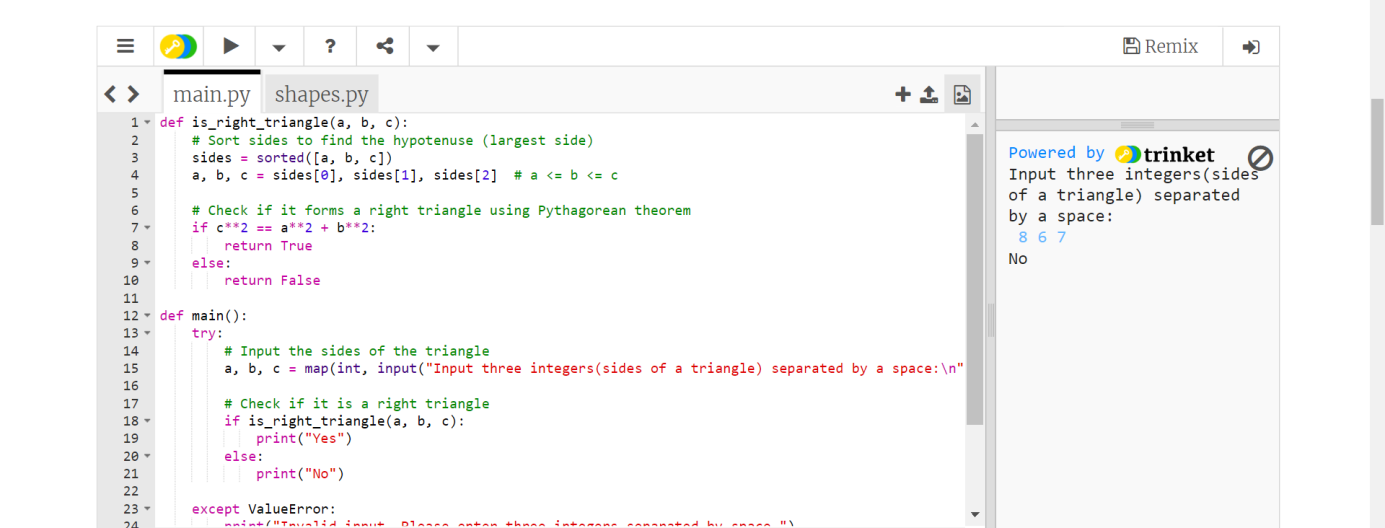


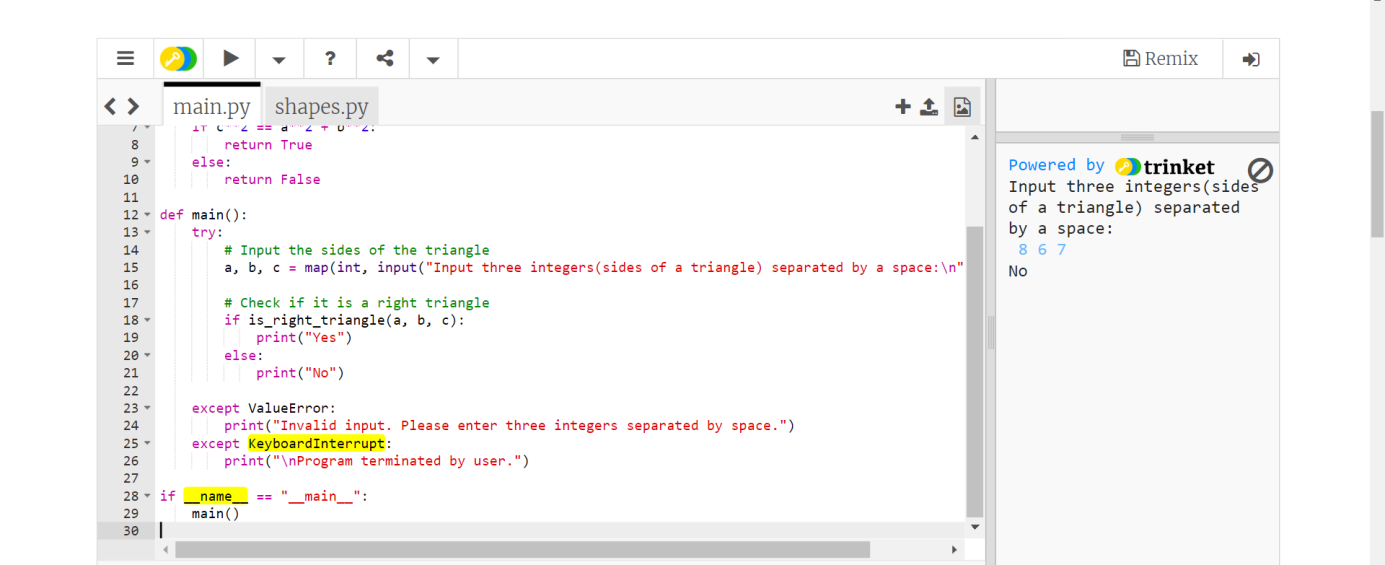
33/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to compute the digit number of the sum of two given integers.  
**Input:**  
Each test case consists of two non-negative integers x and y which are separated by a space in a line.  
0 <= x, y <= 1,000,000  
Input two integers(a b):  
5 7  
Sum of two integers a and b.:  
2



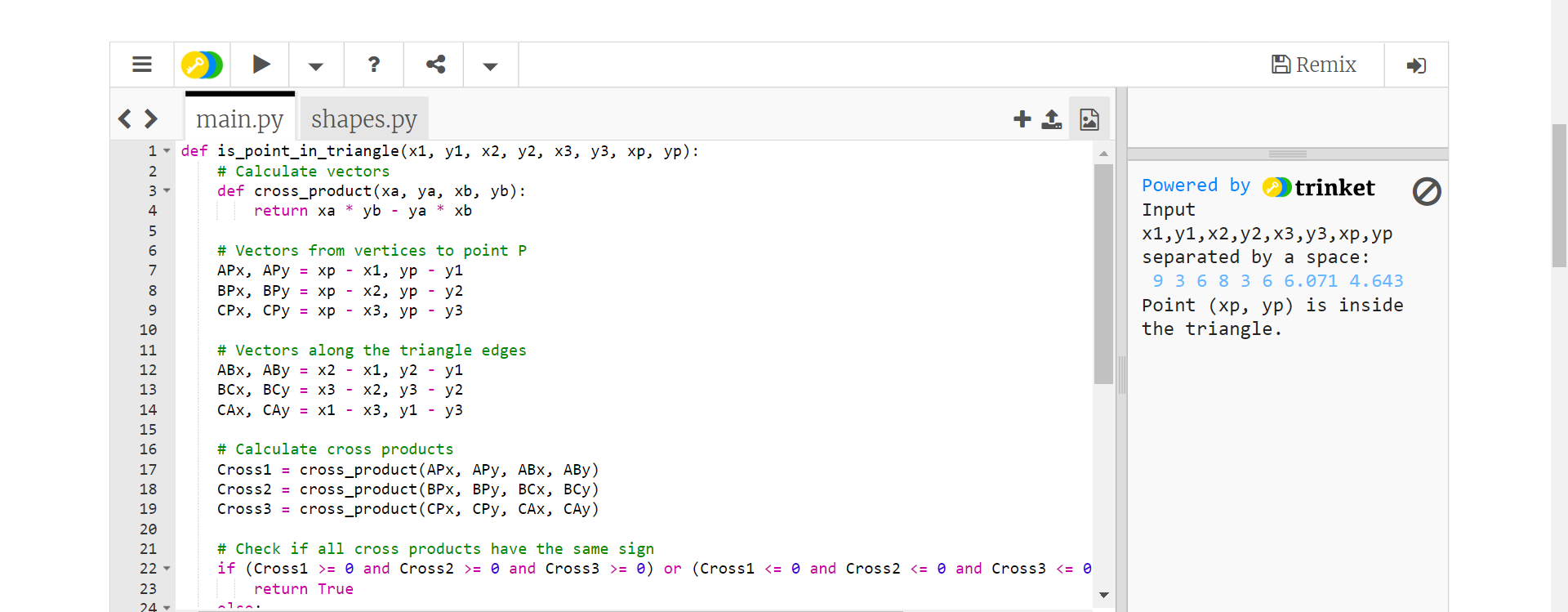


34/-  Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to check whether three given lengths (integers) of three sides form a right triangle. Print "Yes" if the given sides form a right triangle otherwise print "No".  
**Input:**  
Integers separated by a single space.  
1 <= length of the side <= 1,000  
Input three integers(sides of a triangle)  
8 6 7  
No



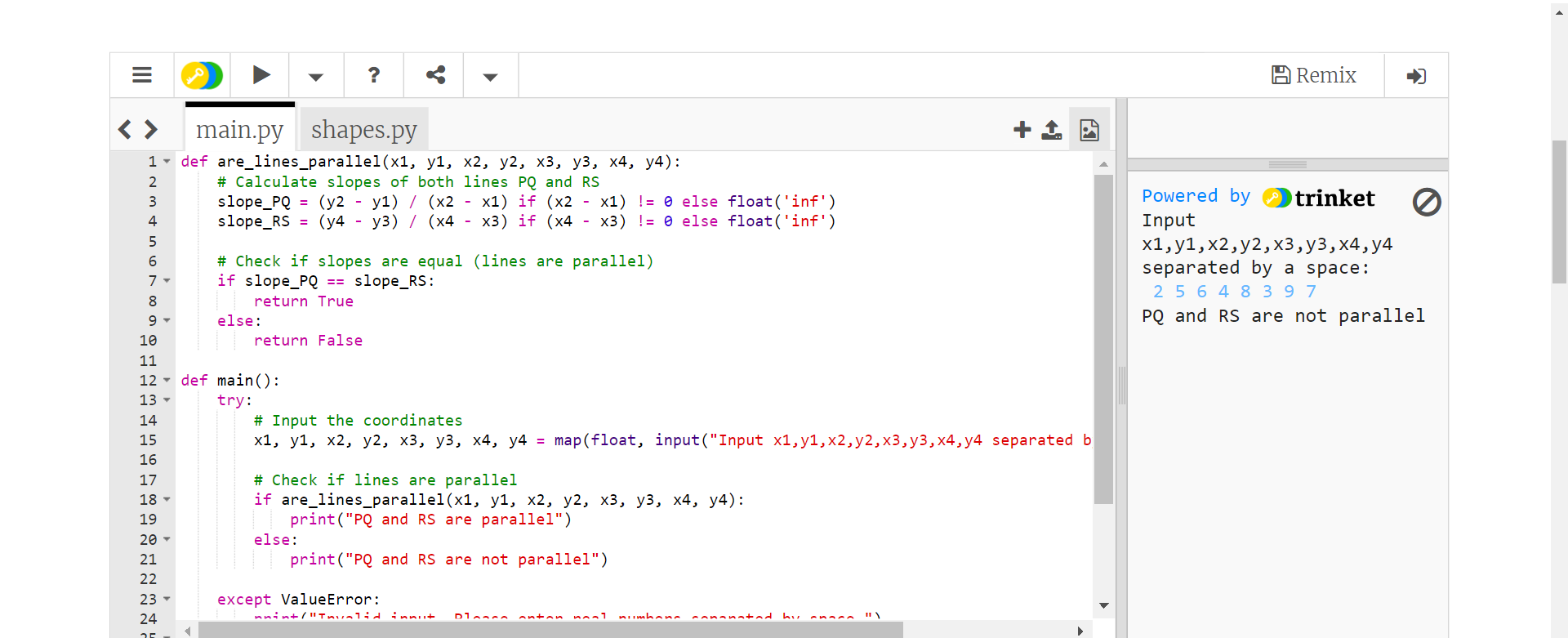


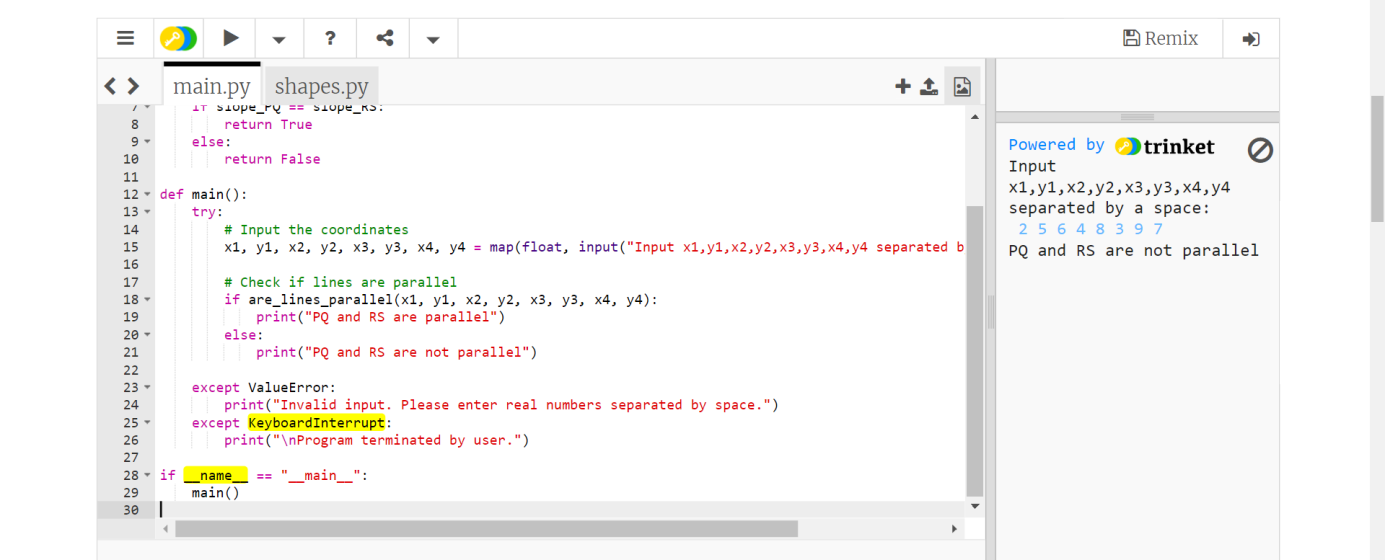
40/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to check if a point (x,y) is in a triangle or not. A triangle is formed by three points.  
**Input:**  
x1,y1,x2,y2,x3,y3,xp,yp separated by a single space.  
Input three coordinate of the circle:  
9 3 6 8 3 6  
Radius of the said circle:  
3.358  
Central coordinate (x, y) of the circle:  
6.071 4.643





43/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to test whether two lines PQ and RS are parallel. The four points are P(x1, y1), Q(x2, y2), R(x3, y3), S(x4, y4).  
**Input:**  
x1,y1,x2,y2,x3,y3,xp,yp separated by a single space  
Input x1,y1,x2,y2,x3,y3,xp,yp:  
2 5 6 4 8 3 9 7  
PQ and RS are not parallel





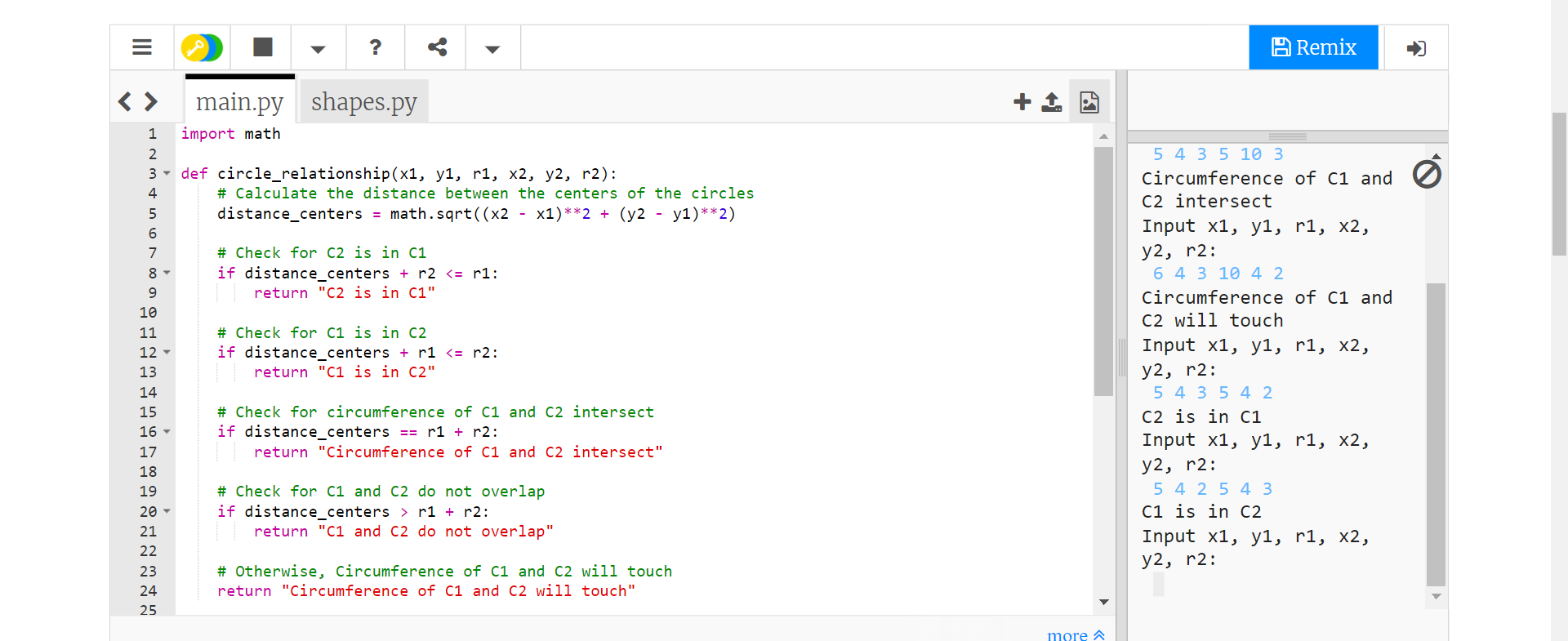
45/-There are two circles C1 with radius r1, central coordinate (x1, y1) and C2 with radius r2 and central coordinate (x2, y2).

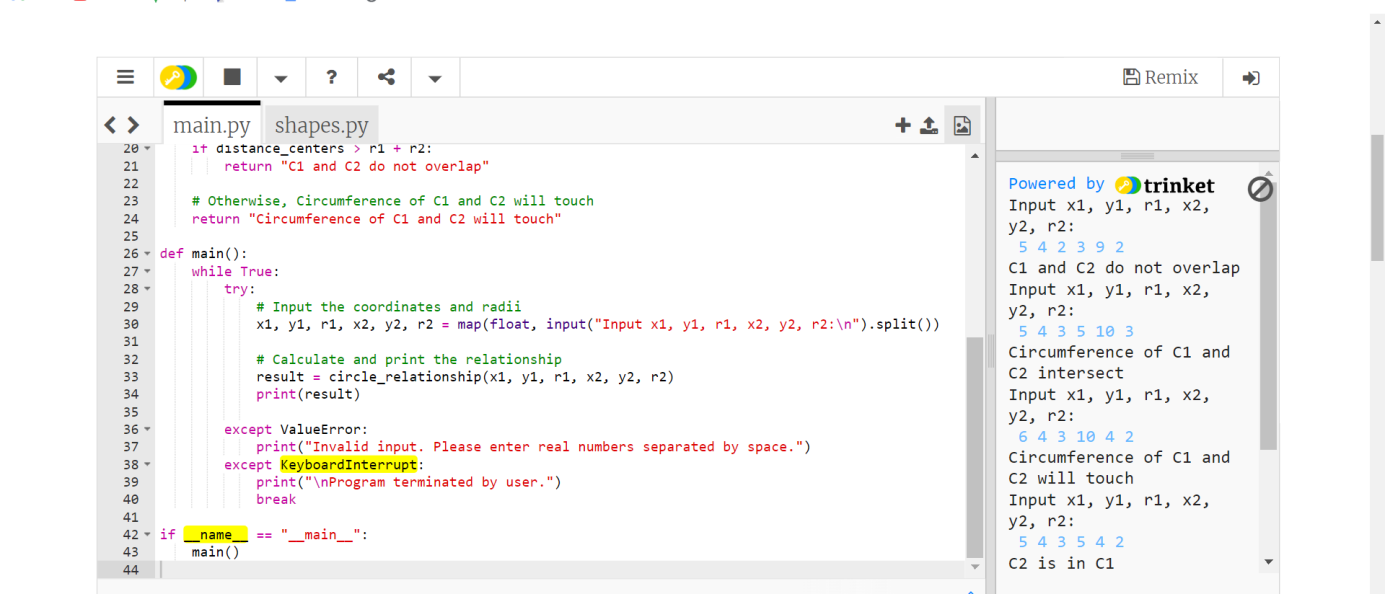
Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to test the followings -

* "C2 is in C1" if C2 is in C1
* "C1 is in C2" if C1 is in C2
* "Circumference of C1 and C2 intersect" if circumference of C1 and C2 intersect
* "C1 and C2 do not overlap" if C1 and C2 do not overlap and
* "Circumference of C1 and C2 will touch" if C1 and C2 touch

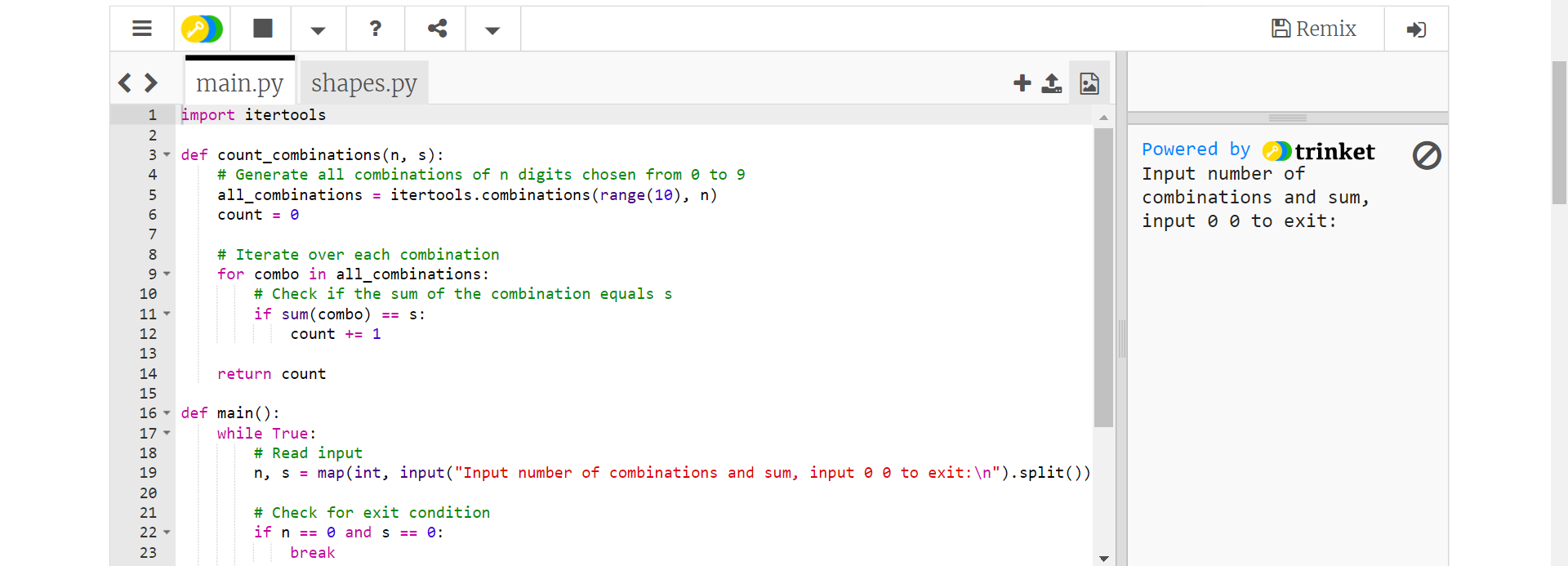
**Input:**  
Input numbers (real numbers) are separated by a space.  
Input x1, y1, r1, x2, y2, r2:  
5 4 2 3 9 2  
C1 and C2 do not overlap  
Input x1, y1, r1, x2, y2, r2:  
5 4 3 5 10 3  
Circumference of C1 and C2 will touch  
Input x1, y1, r1, x2, y2, r2:  
6 4 3 10 4 2  
Circumference of C1 and C2 intersect  
Input x1, y1, r1, x2, y2, r2:  
5 4 3 5 4 2

C2 is in C1  
Input x1, y1, r1, x2, y2, r2:  
5 4 2 5 4 3  
C1 is in C2



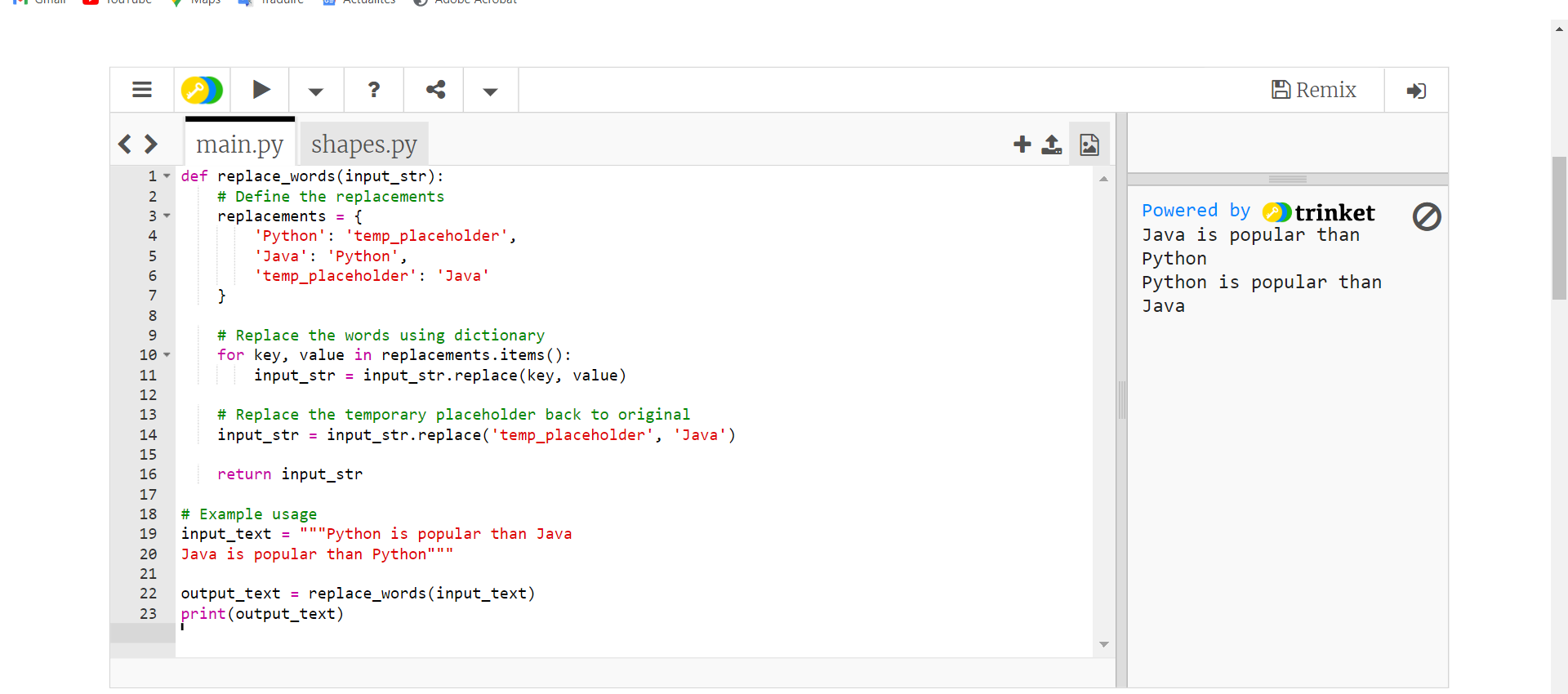


48/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program that reads n digits (given) chosen from 0 to 9 and prints the number of combinations where the sum of the digits equals another given number (s). Do not use the same digits in a combination.  
**Input:**  
Two integers as number of combinations and their sum by a single space in a line. Input 0 0 to exit.  
Input number of combinations and sum, input 0 0 to exit:  
5 6  
2 4  
0 0  
2





50/-Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to replace a string "[Python](https://www.w3resource.com/python-exercises/basic/)" with "Java" and "Java" with "Python" in a given string.  
**Input:**  
English letters (including single byte alphanumeric characters, blanks, symbols) are given on one line. The length of the input character string is 1000 or less.  
Input a text with two words 'Python' and 'Java'  
Python is popular than Java  
Java is popular than Python

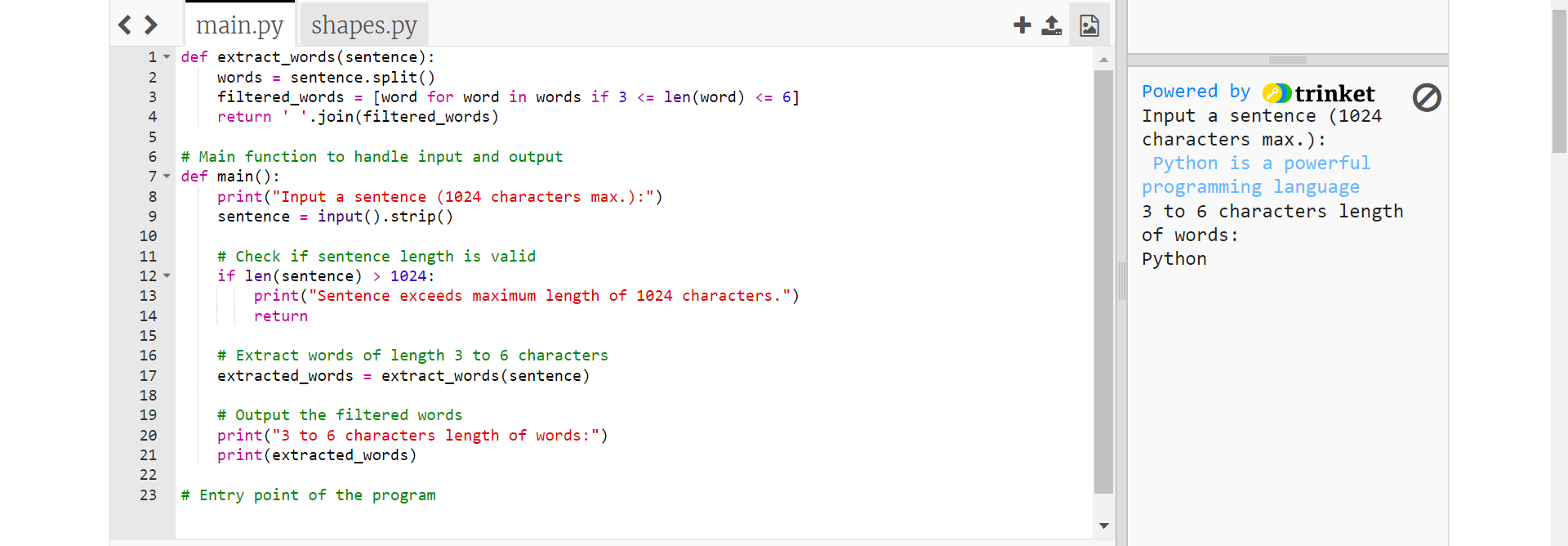


58/-When character are consecutive in a string , it is possible to shorten the character string by replacing the character with a certain rule. For example, in the case of the character string YYYYY, if it is expressed as # 5 Y, it is compressed by one character.  
Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to restore the original string by entering the compressed string with this rule. However, the # character does not appear in the restored character string.  
**Input:**  
The restored character string for each character on one line.  
Original text: XY#6Z1#4023  
XYZZZZZZ1000023  
Original text: #39+1=1#30  
999+1=1000



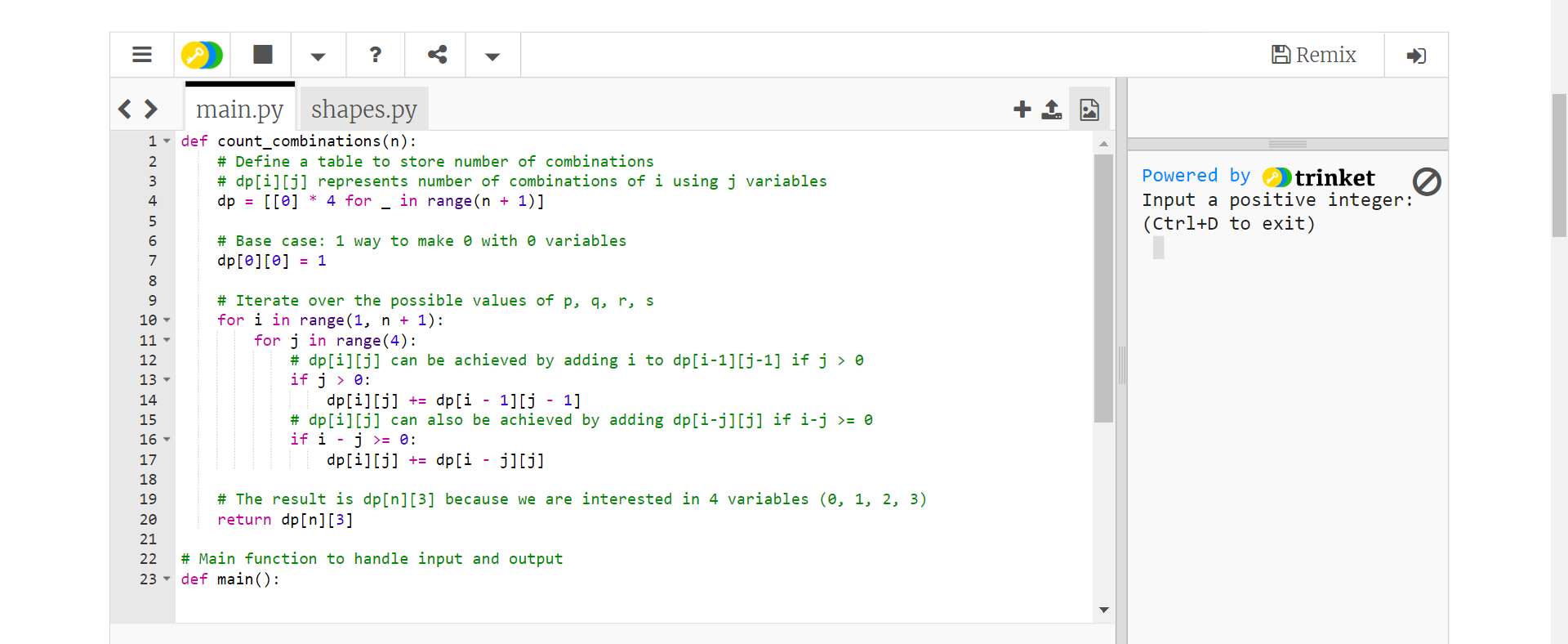


60/-Internet search engine giant, such as Google accepts web pages around the world and classify them, creating a huge database. The search engines also analyze the search keywords entered by the user and create inquiries for database search. In both cases, complicated processing is carried out in order to realize efficient retrieval, but basics are all cutting out words from sentences.  
Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to cut out words of 3 to 6 characters length from a given sentence not more than 1024 characters.  
**Input:**  
English sentences consisting of delimiters and alphanumeric characters are given on one line.  
Input a sentence (1024 characters. max.)  
The quick brown fox  
3 to 6 characters length of words:  
The quick brown fox



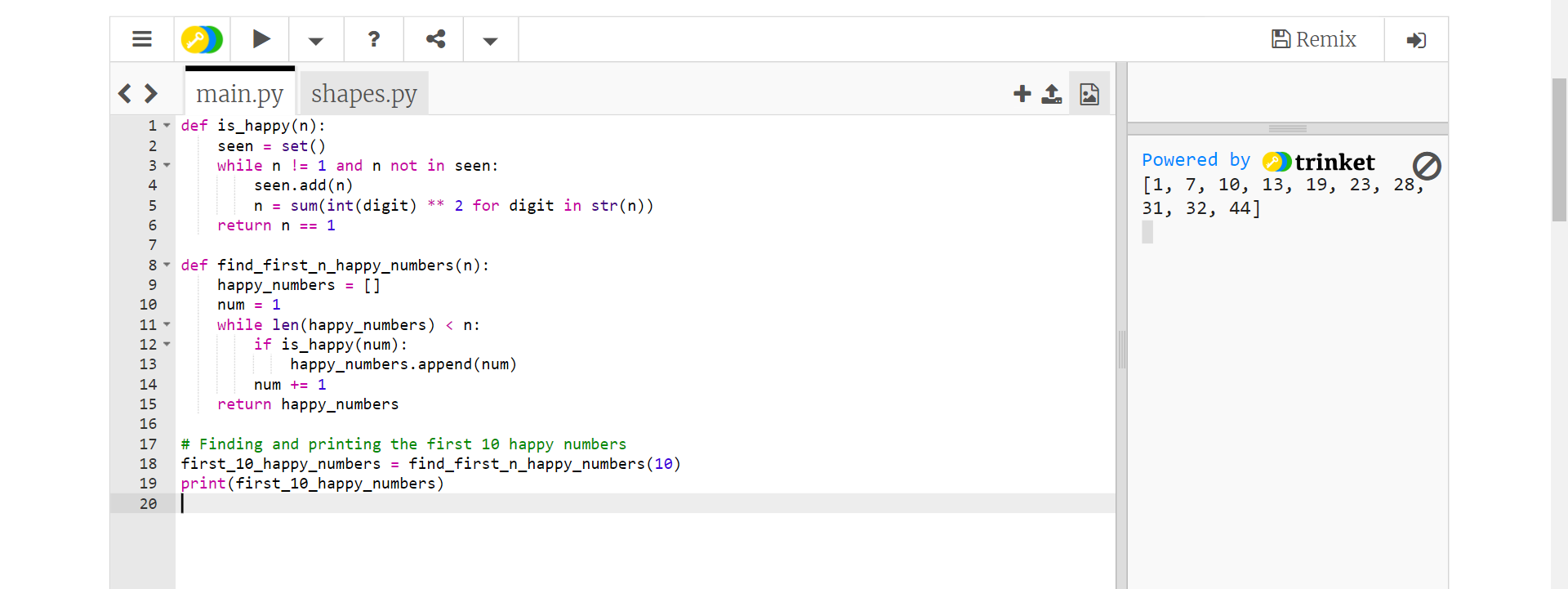


62/- Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to find the number of combinations that satisfy p + q + r + s = n where n is a given number <= 4000 and p, q, r, s are between 0 to 1000.  
Input a positive integer: (ctrl+d to exit)  
252  
Number of combinations of a,b,c,d: 2731135





67/- From Wikipedia,  
A happy number is defined by the following process:  
Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers, while those that do not end in 1 are unhappy numbers.  
Write a [Python](https://www.w3resource.com/python-exercises/basic/) program to find and print the first 10 happy numbers.  
Sample Input:  
[:10]  
Sample Output:  
[1, 7, 10, 13, 19, 23, 28, 31, 32, 44]



68/-  Write a [Python](https://www.w3resource.com/python-exercises/basic/) program that counts the number of prime numbers that are less than a given non-negative number.  
Sample Input:  
(10)  
(100)  
Sample Output:  
4  
25

