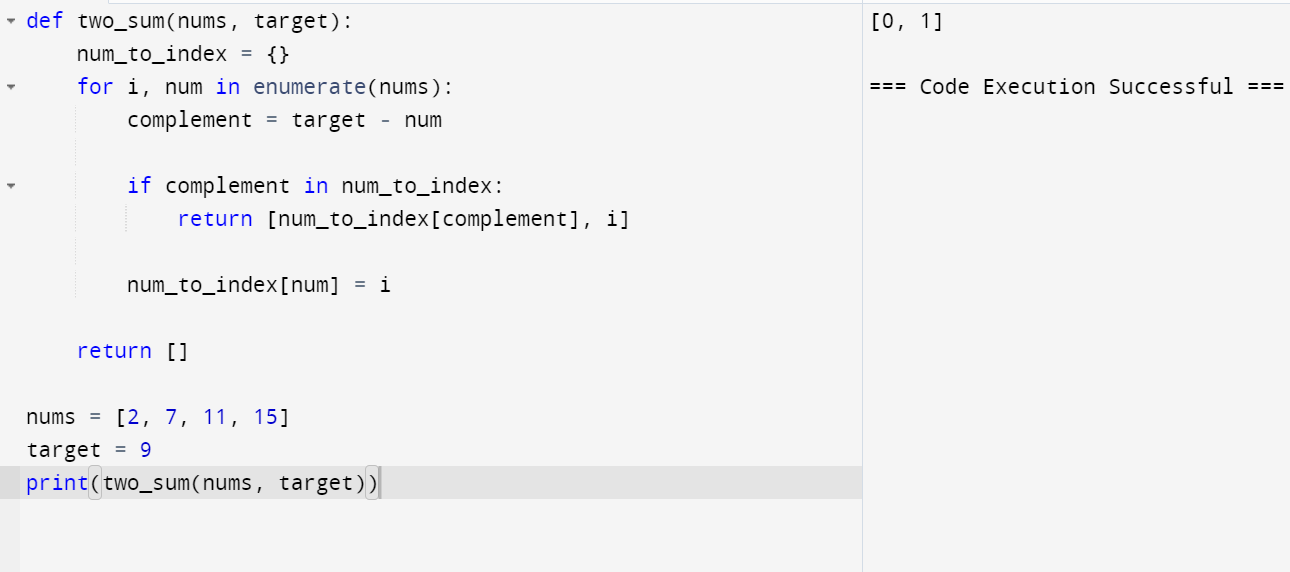
1. Two Sum

Given an array of integers nums and an integer target, return indices of the two numbers such

that they add up to target.

You may assume that each input would have exactly one solution, and you may not use the same

element twice.

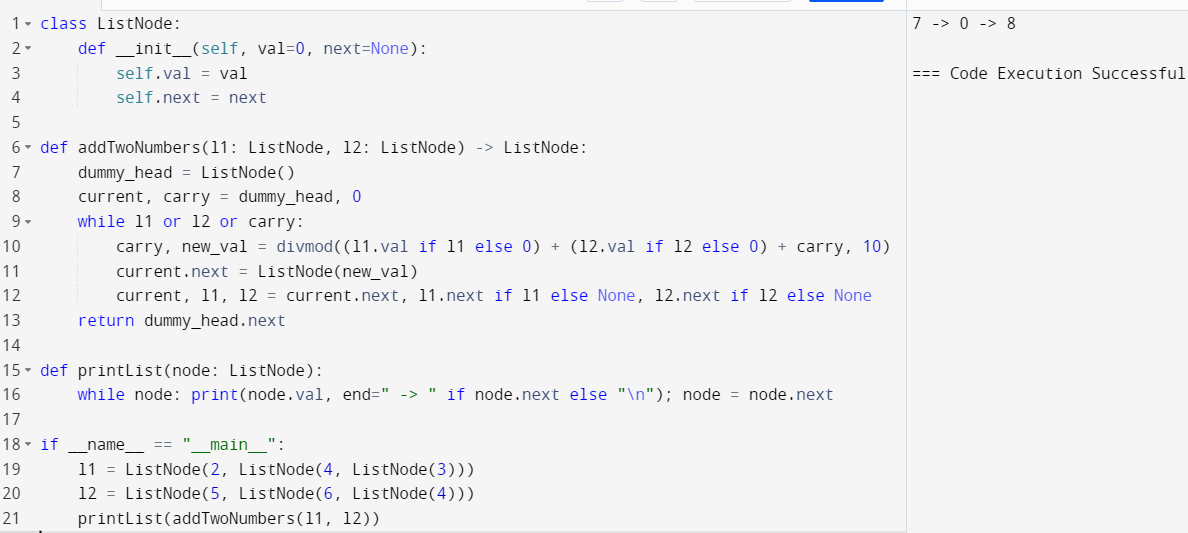


2. Add Two Numbers

You are given two non-empty linked lists representing two non-negative integers. The digits are

stored in reverse order, and each of their nodes contains a single digit. Add the two numbers and

return the sum as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

3. Longest Substring without Repeating Characters

Given a string s, find the length of the longest substring without repeating characters.

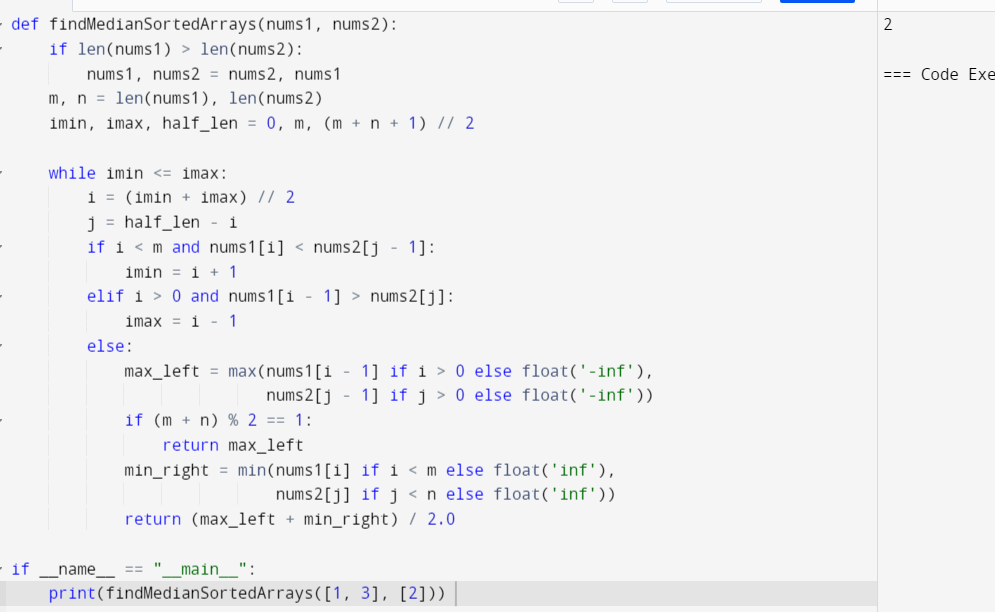


4. Median of Two Sorted Arrays

Given two sorted arrays nums1 and nums2 of size m and n respectively, return the median of the

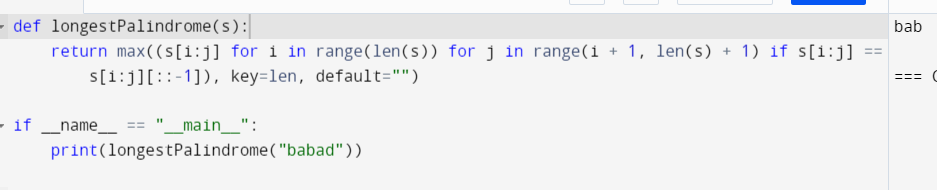
two sorted arrays.

The overall run time complexity should be O(log (m+n)).



5. Longest Palindromic Substring

Given a string s, return the longest palindromic substring in s.



7. Reverse Integer

Given a signed 32-bit integer x, return x with its digits reversed. If reversing x causes the value

to go outside the signed 32-bit integer range [-231, 231 - 1], then return 0.

Assume the environment does not allow you to store 64-bit integers (signed or unsigned).



8. String to Integer (atoi)

Implement the myAtoi(string s) function, which converts a string to a 32-bit signed integer

(similar to C/C++'s atoi function).

The algorithm for myAtoi(string s) is as follows:

1. Read in and ignore any leading whitespace.

2. Check if the next character (if not already at the end of the string) is '-' or '+'. Read this

character in if it is either. This determines if the final result is negative or positive

respectively. Assume the result is positive if neither is present.

3. Read in next the characters until the next non-digit character or the end of the input is

reached. The rest of the string is ignored.

4. Convert these digits into an integer (i.e. "123" -> 123, "0032" -> 32). If no digits were

read, then the integer is 0. Change the sign as necessary (from step 2).

5. If the integer is out of the 32-bit signed integer range [-231, 231 - 1], then clamp the

integer so that it remains in the range. Specifically, integers less than -231 should be

clamped to -231, and integers greater than 231 - 1 should be clamped to 231 - 1.

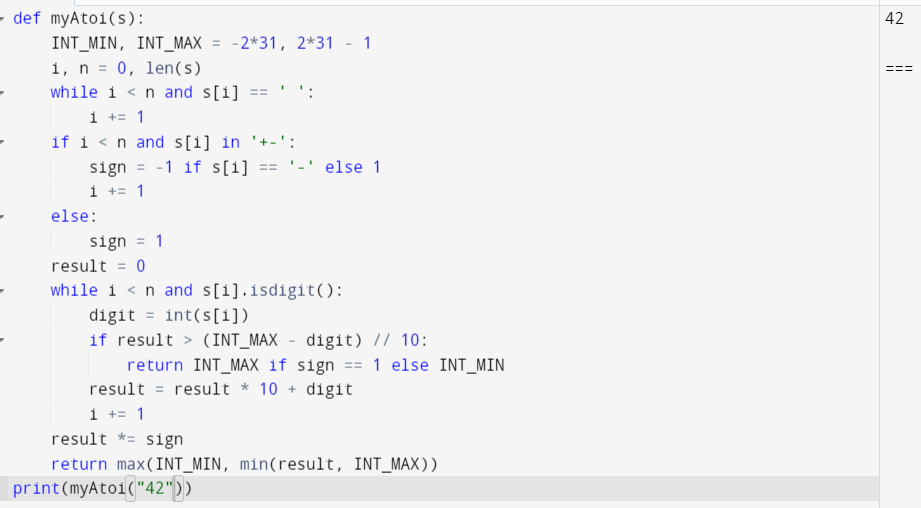
6. Return the integer as the final result.

Note:

● Only the space character ' ' is considered a whitespace character.

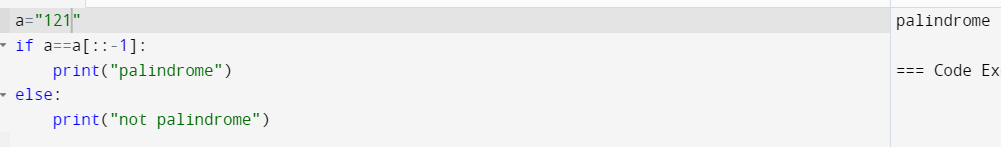
● Do not ignore any characters other than the leading whitespace or the rest of the string

after the digits.



9. Palindrome Number

Given an integer x, return true if x is a palindrome, and false otherwise.



10. Regular Expression Matching

Given an input string s and a pattern p, implement regular expression matching with support for

'.' and '\*' where:

● '.' Matches any single character.

● '\*' Matches zero or more of the preceding element.

The matching should cover the entire input string (not partial).

