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.

You can return the answer in any order.

Example 1:

Input: nums = [2,7,11,15], target = 9

Output: [0,1]

Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].

Example 2:

Input: nums = [3,2,4], target = 6

Output: [1,2]

Example 3:

Input: nums = [3,3], target = 6

Output: [0,1]

Constraints:

2 <= nums.length <= 104

-109 <= nums[i] <= 109

-109 <= target <= 109

Only one valid answer exists.

# program

class Solution:

def twoSum(self, nums: List[int], target: int) -> List[int]:

numToIndex = {}

for i in range(len(nums)):

if target - nums[i] in numToIndex:

return [numToIndex[target - nums[i]], i]

numToIndex[nums[i]] = i

return []

Given an integer x, return true if x is a

palindrome

, and false otherwise.

Example 1:

Input: x = 121

Output: true

Explanation: 121 reads as 121 from left to right and from right to left.

Example 2:

Input: x = -121

Output: false

Explanation: From left to right, it reads -121. From right to left, it becomes 121-. Therefore it is not a palindrome.

Example 3:

Input: x = 10

Output: false

Explanation: Reads 01 from right to left. Therefore it is not a palindrome.

Constraints:

-231 <= x <= 231 - 1

Follow up: Could you solve it without converting the integer to a string?

# program

class Solution:

def isPalindrome(self, x: int) -> bool:

if x < 0:

return False

reversed\_num = 0

num = x

while num != 0:

a = num % 10

reversed\_num = reversed\_num \* 10 + a

num //= 10

return reversed\_num == x

Pallindrome

You are given an array prices where prices[i] is the price of a given stock on the ith day.

You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock.

Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0.

Example 1:

Input: prices = [7,1,5,3,6,4]

Output: 5

Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.

Note that buying on day 2 and selling on day 1 is not allowed because you must buy before you sell.

Example 2:

Input: prices = [7,6,4,3,1]

Output: 0

Explanation: In this case, no transactions are done and the max profit = 0.

Constraints:

1 <= prices.length <= 105

0 <= prices[i] <= 104

class Solution:

def maxProfit(self, prices: List[int]) -> int:

min\_price = prices[0]

max\_profit = 0

for price in prices[1:]:

max\_profit = max(max\_profit, price - min\_price)

min\_price = min(min\_price, price)

return max\_profit

Best time to buy

Two sum

Contains duplicate

Given an integer array nums, return true if any value appears at least twice in the array, and return false if every element is distinct.

Example 1:

Input: nums = [1,2,3,1]

Output: true

Example 2:

Input: nums = [1,2,3,4]

Output: false

Example 3:

Input: nums = [1,1,1,3,3,4,3,2,4,2]

Output: true

Constraints:

1 <= nums.length <= 105

-109 <= nums[i] <= 109

class Solution:

def containsDuplicate(self, nums: List[int]) -> bool:

n = len(nums)

for i in range(n - 1):

for j in range(i + 1, n):

if nums[i] == nums[j]:

return True

return False