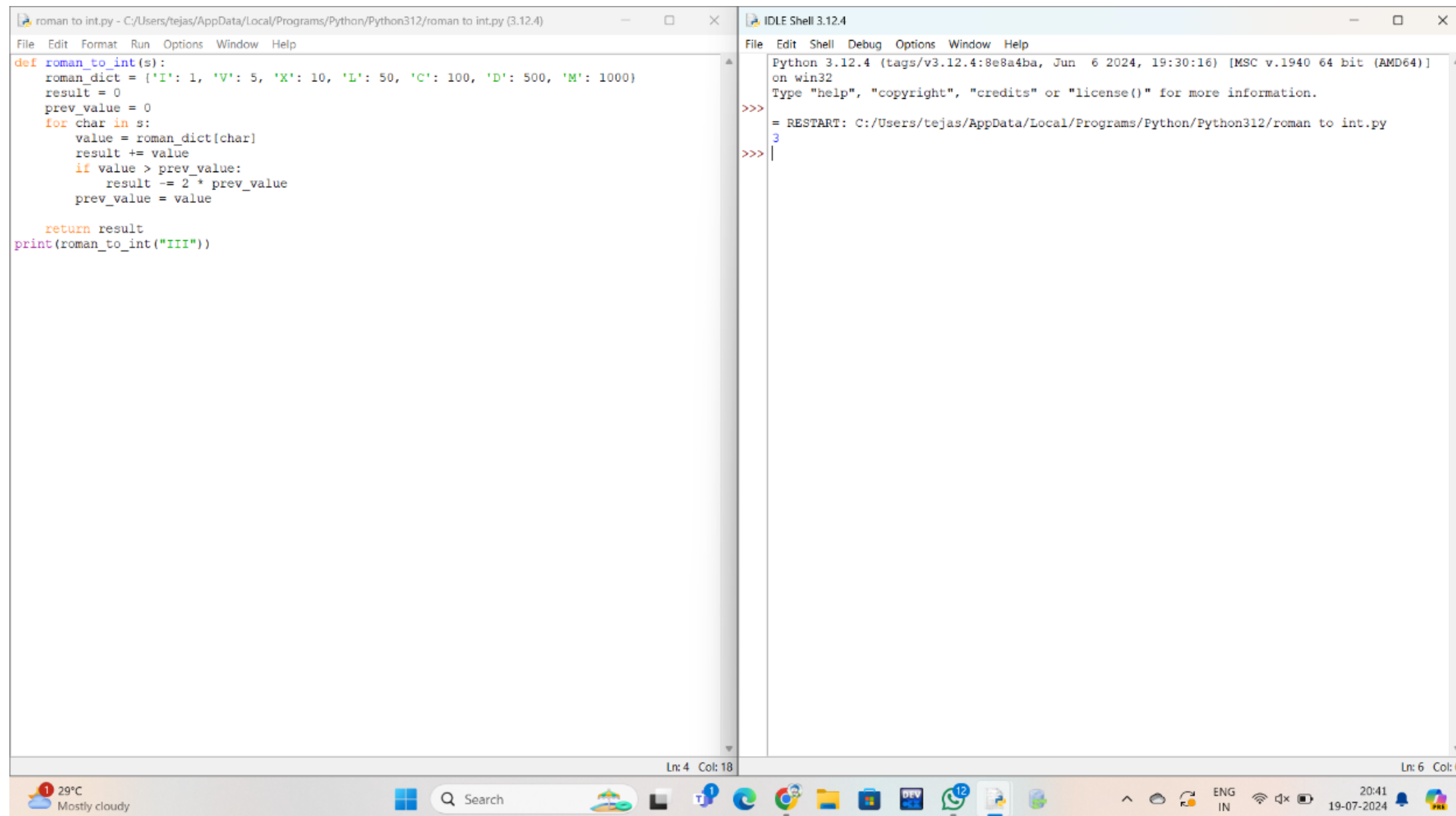


ASSIGNMENT 2

1. Converting roman to integers



```
roman to int.py - C:/Users/tejas/AppData/Local/Programs/Python/Python312/roman to int.py (3.12.4)
File Edit Format Run Options Window Help
def roman_to_int(s):
    roman_dict = {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}
    result = 0
    prev_value = 0
    for char in s:
        value = roman_dict[char]
        result += value
        if value > prev_value:
            result -= 2 * prev_value
            prev_value = value
    return result
print(roman_to_int("III"))

IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/tejas/AppData/Local/Programs/Python/Python312/roman to int.py
3
>>>
```

2.

Longest Common Prefix

Write a function to find the longest common prefix string amongst an array of strings.

If there is no common prefix, return an empty string "".

Example 1:

Input: strs = ["flower","flow","flight"]

Output: "fl"

Example 2:

Input: strs = ["dog","racecar","car"]

Output: ""

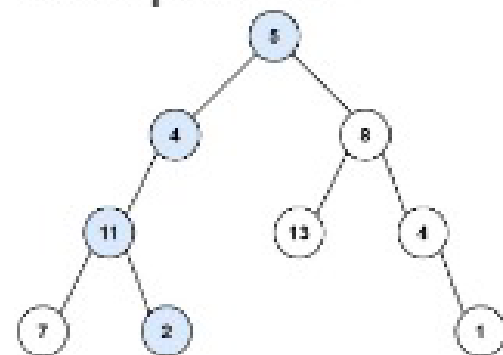
Explanation: There is no common prefix among the input strings.

```
longest common prefix.py - C:/Users/tejas/AppData/Local/Programs/Python/Python312/longest common prefix...
File Edit Format Run Options Window Help
def longest_common_prefix(strs):
    if not strs:
        return ""
    strs.sort()
    prefix = ""
    for i in range(len(strs[0])):
        if all(s[i] == strs[0][i] for s in strs):
            prefix += strs[0][i]
        else:
            break
    return prefix
str_list = ["flower", "flow", "flight"]
print(longest_common_prefix(str_list))

IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/tejas/AppData/Local/Programs/Python/Python312/longest common prefix.
py
fl
>>>
```

3. Given the root of a binary tree and an integer of targetsum return true if the tree has a root to leaf such that adding up all the values

Example 1:

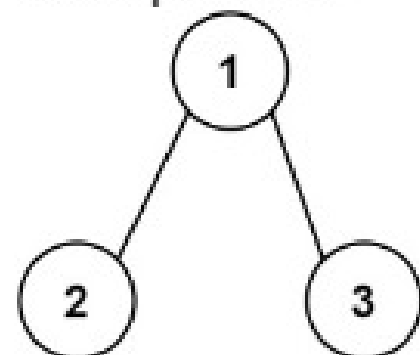


Input: root = [5,4,8,11,null,13,4,7,2,null,null,null,1], targetSum = 22

Output: true

Explanation: The root-to-leaf path with the target sum is shown.

Example 2:



Input: root = [1,2,3], targetSum = 5

Output: false

Explanation: There two root-to-leaf paths in the tree:

(1 --> 2): The sum is 3.

(1 --> 3): The sum is 4.

There is no root-to-leaf path with sum = 5.

```
FILE TARGETSUM.py - C:/Users/tejas/AppData/Local/Programs/Python/Python312/TREE TARGETSUM.py (3.12.4)
File Edit Format Run Options Window Help
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right
class Solution:
    def hasPathSum(self, root: TreeNode, targetSum: int) -> bool:
        if not root:
            return False
        def dfs(node, current_sum):
            if not node:
                return False
            current_sum += node.val
            if not node.left and not node.right:
                return current_sum == targetSum
            return dfs(node.left, current_sum) or dfs(node.right, current_sum)
        return dfs(root, 0)
root = TreeNode(5)
root.left = TreeNode(4)
root.right = TreeNode(8)
root.left.left = TreeNode(11)
root.right.left = TreeNode(13)
root.right.right = TreeNode(4)
root.left.left.left = TreeNode(7)
root.left.left.right = TreeNode(2)
solution = Solution()
print(solution.hasPathSum(root, 22))

Ln: 5 Col: 26

IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/tejas/AppData/Local/Programs/Python/Python312/TREE TARGETSUM.py
True
>>>

Ln: 6 Col: 0
```

4. Binary tree traversal

```
binary tree traversal.py - C:/Users/tejas/AppData/Local/Programs/Python/Python312/binary tree traversal.py (3.12.4)
File Edit Format Run Options Window Help
class Node:
    def __init__(self, key):
        self.left = None
        self.right = None
        self.val = key
def inorder_traversal(node):
    if node:
        inorder_traversal(node.left)
        print(node.val)
        inorder_traversal(node.right)
def preorder_traversal(node):
    if node:
        print(node.val)
        preorder_traversal(node.left)
        preorder_traversal(node.right)
def postorder_traversal(node):
    if node:
        postorder_traversal(node.left)
        postorder_traversal(node.right)
        print(node.val)
root = Node(1)
root.left = Node(2)
root.right = Node(3)
root.left.left = Node(4)
root.left.right = Node(5)
print("\nInorder Traversal:")
inorder_traversal(root)
print("\nPreorder Traversal:")
preorder_traversal(root)
print("\nPostorder Traversal:")
postorder_traversal(root)

Ln: 1 Col: 0

IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/tejas/AppData/Local/Programs/Python/Python312/binary tree traversal.py
Inorder Traversal:
4
2
5
1
3
Preorder Traversal:
1
2
4
5
3
Postorder Traversal:
4
5
2
3
1
>>>

Ln: 25 Col: 0
```

5. Bit reserving

```
bit_reservingg.py - C:/Users/tejas/AppData/Local/Programs/Python/Python312/bit_reservingg.py (3.12.4)
File Edit Format Run Options Window Help
def reverse_bits(num, num_bits):
    result = 0
    for _ in range(num_bits):
        result = (result << 1) | (num & 1)
        num >>= 1
    return result
num = 10
num_bits = 4
reversed_num = reverse_bits(num, num_bits)
print(f"Original Number: {num}, Reversed Number: {reversed_num}")

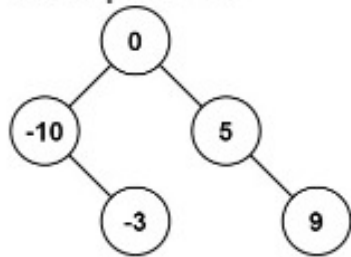
IDLE Shell 3.12.4
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/tejas/AppData/Local/Programs/Python/Python312/bit_reservingg.py
Original Number: 10, Reversed Number: 5
>>>
```

6.

Convert Sorted Array to Binary Search Tree

Given an integer array `nums` where the elements are sorted in ascending order, convert it to a height-balanced binary search tree.

Example 1:

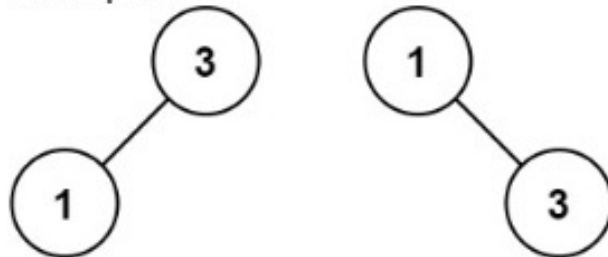


Input: `nums = [-10,-3,0,5,9]`

Output: `[0,-3,9,-10,null,5]`

Explanation: `[0,-10,5,null,-3,null,9]` is also accepted:

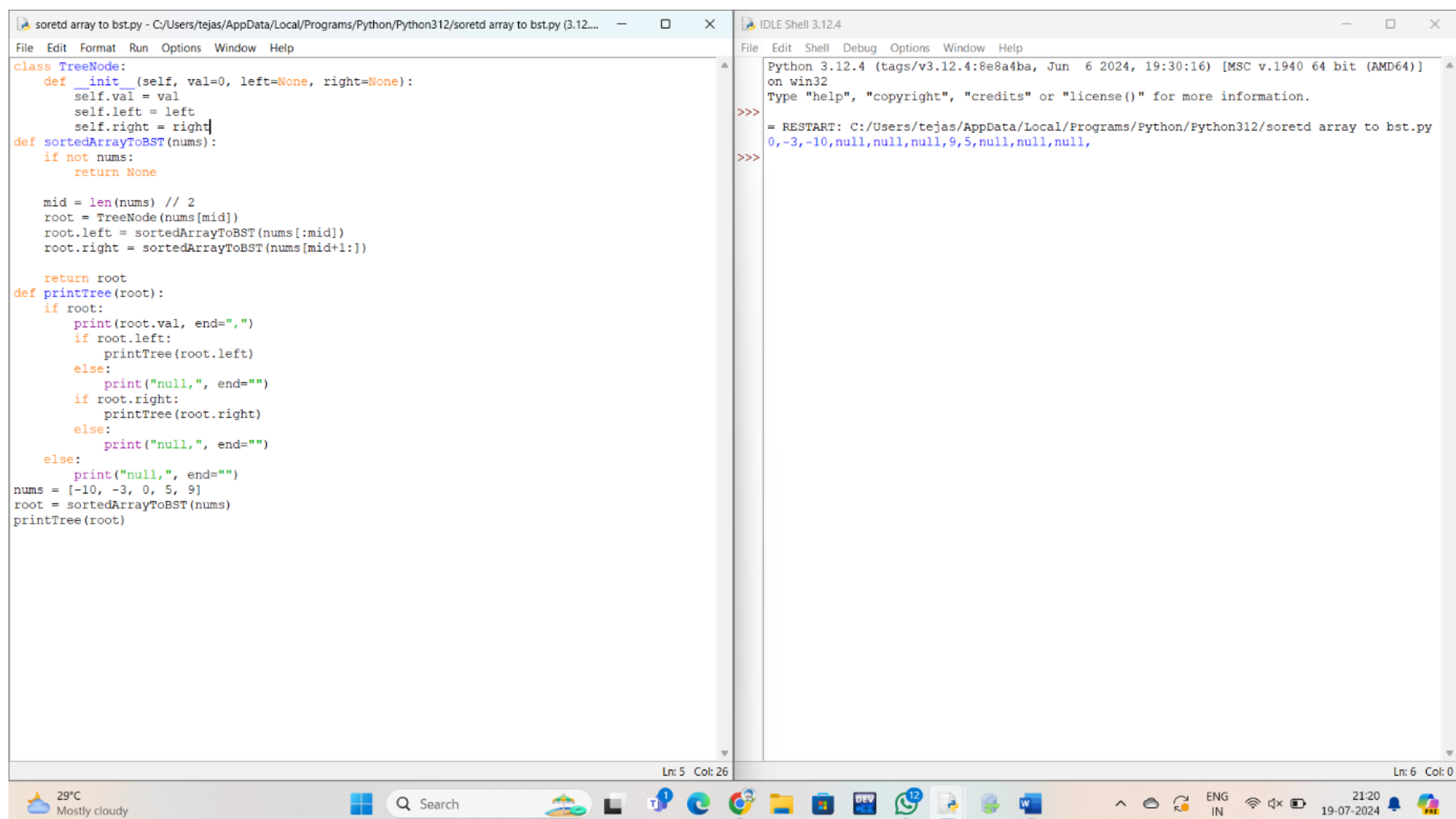
Example 2:



Input: `nums = [1,3]`

Output: `[3,1]`

Explanation: `[1,null,3]` and `[3,1]` are both height-balanced BSTs.



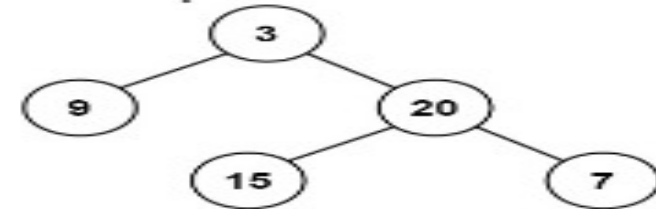
7.

Balanced Binary Tree

Given a binary tree, determine if it is height-balanced

.

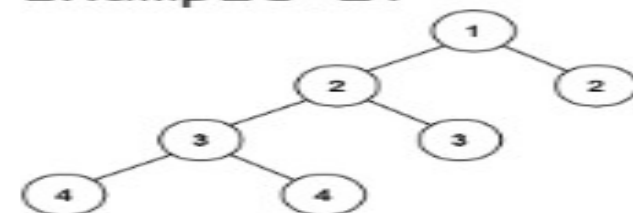
Example 1:



Input: root = [3,9,20,null,null,15,7]

Output: true

Example 2:



Input: root = [1,2,2,3,3,null,null,4,4]

Output: false

Example 3:

Input: root = []

Output: true

The screenshot shows a Python IDE with two windows. The left window, titled 'balanced binary tree.py', contains the following code:

```
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right

def isBalanced(root):
    def checkBalance(node):
        if node is None:
            return 0, True

        leftHeight, leftBalanced = checkBalance(node.left)
        rightHeight, rightBalanced = checkBalance(node.right)

        return max(leftHeight, rightHeight) + 1, leftBalanced and rightBalanced and abs(leftHeight - rightHeight) < 2

    return checkBalance(root)[1]

root = TreeNode(3)
root.left = TreeNode(9)
root.right = TreeNode(20)
root.right.left = TreeNode(15)
root.right.right = TreeNode(7)

print(isBalanced(root))
```

The right window, titled 'IDLE Shell 3.12.4', shows the output of the program:

```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/tejas/AppData/Local/Programs/Python/Python312/balanced binary tree.
py
True
>>>
```

The taskbar at the bottom shows the system clock as 21:25 on 19-07-2024, with a temperature of 29°C and 'Mostly cloudy' weather.

8.

Climbing Stairs

You are climbing a staircase. It takes n steps to reach the top.

Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

Example 1:

Input: $n = 2$

Output: 2

Explanation: There are two ways to climb to the top.

1. 1 step + 1 step

2. 2 steps

Example 2:

Input: $n = 3$

Output: 3

Explanation: There are three ways to climb to the top.

1. 1 step + 1 step + 1 step

2. 1 step + 2 steps

3. 2 steps + 1 step

The screenshot shows a Python IDE with two windows. The left window, titled 'climbing stairs.py', contains the following code:

```
def climbStairs(n: int) -> int:
    if n <= 2:
        return n
    dp = [0] * (n + 1)
    dp[1] = 1
    dp[2] = 2
    for i in range(3, n + 1):
        dp[i] = dp[i - 1] + dp[i - 2]
    return dp[n]

print(climbStairs(2))
```

The right window, titled 'IDLE Shell 3.12.4', shows the output of the program:

```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/tejas/AppData/Local/Programs/Python/Python312/climbing stairs.py
2
>>>
```

The taskbar at the bottom shows the system clock as 21:28 on 19-07-2024.

9.

Best Time to Buy and Sell Stock

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.

The screenshot shows a Python IDE with two windows. The left window, titled 'best time to buy.py', contains the following code:

```
def maxProfit(prices):
    if not prices:
        return 0
    min_price = prices[0]
    max_profit = 0
    for price in prices:
        if price < min_price:
            min_price = price
        else:
            max_profit = max(max_profit, price - min_price)
    return max_profit
prices = [7,1,5,3,6,4]
print(maxProfit(prices))
```

The right window, titled 'IDLE Shell 3.12.4', shows the output of the program:

```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/tejas/AppData/Local/Programs/Python/Python312/best time to buy.py
5
>>>
```

The taskbar at the bottom shows the system clock as 21:31 on 19-07-2024.

10.

Add Binary

Given two binary strings *a* and *b*, return their sum as a binary string.

Example 1:

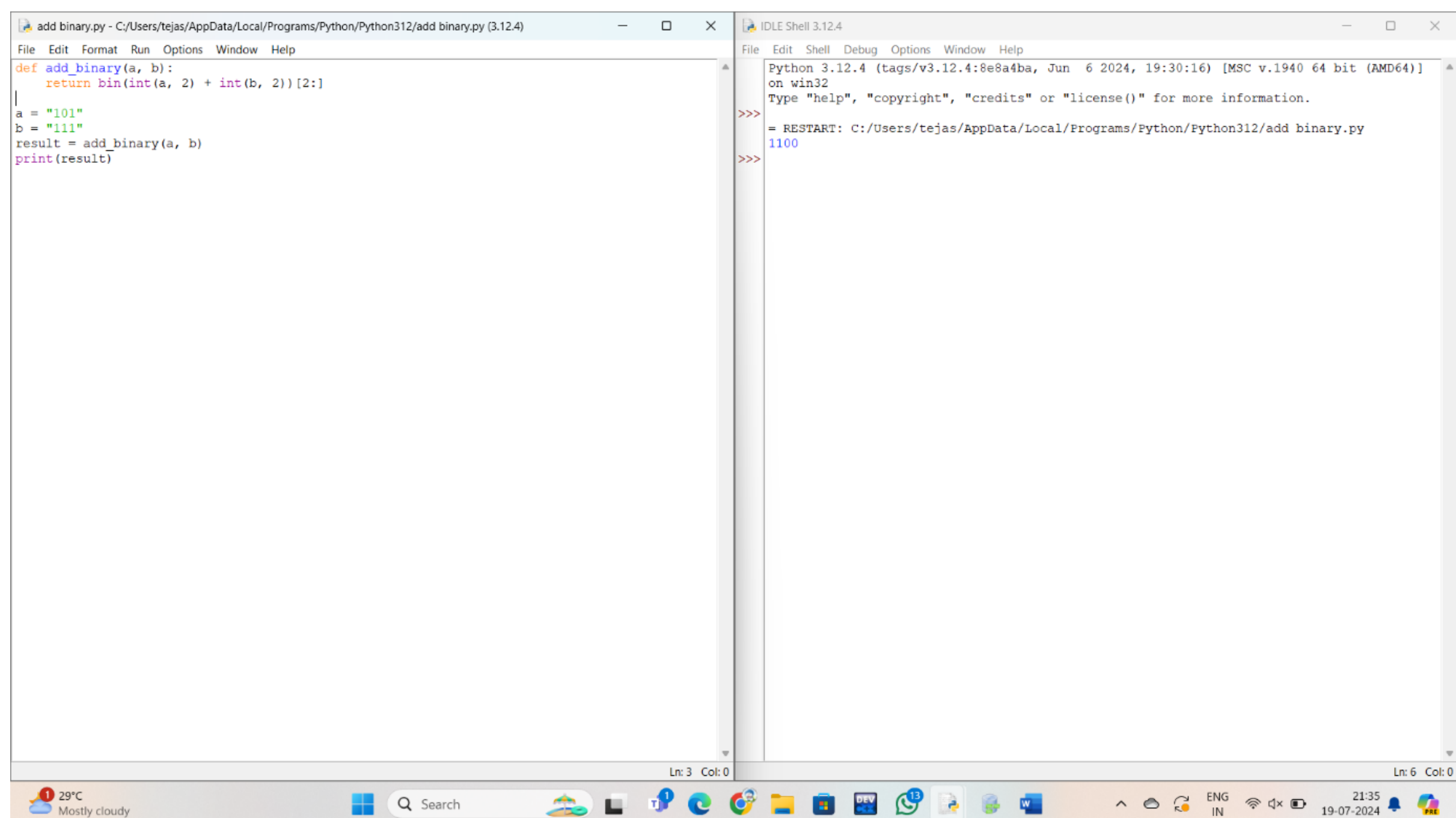
Input: *a* = "11", *b* = "1"

Output: "100"

Example 2:

Input: *a* = "1010", *b* = "1011"

Output: "10101"



The screenshot displays a Python IDE with two windows. The left window, titled 'add binary.py', contains the following code:

```
def add_binary(a, b):  
    return bin(int(a, 2) + int(b, 2))[2:]  
|  
a = "101"  
b = "111"  
result = add_binary(a, b)  
print(result)
```

The right window, titled 'IDLE Shell 3.12.4', shows the execution output:

```
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]  
on win32  
>>> Type "help", "copyright", "credits" or "license()" for more information.  
>>> = RESTART: C:/Users/tejas/AppData/Local/Programs/Python/Python312/add binary.py  
1100  
>>>
```

The taskbar at the bottom shows the system clock as 21:35 on 19-07-2024, with a temperature of 29°C and 'Mostly cloudy' weather.