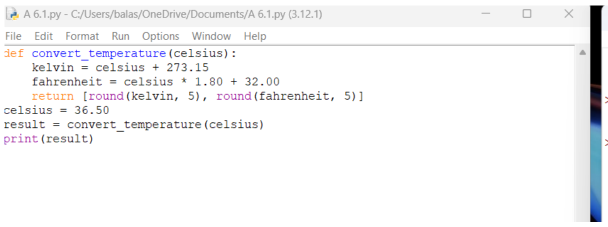
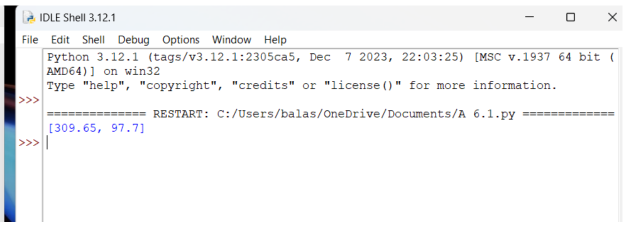
1. Convert the Temperature You are given a non-negative floating point number rounded to two decimal places celsius, that denotes the temperature in Celsius.You should convert Celsius into Kelvin and Fahrenheit and return it as an array ans = [kelvin, fahrenheit]. Return the array ans. Answers within 10-5 of the actual answer will be accepted.

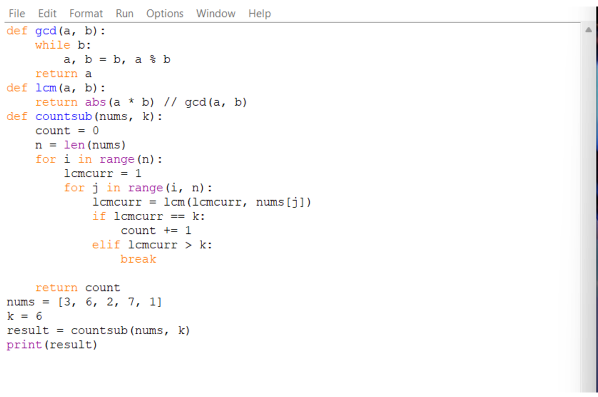

A 6.1.py - C:/Users/balas/OneDrive/Documents/A 6.1.py (3.12.1)
File Edit Format Run Options Window Help

```python
def convert_temperature(celsius):
    kelvin = celsius + 273.15
    fahrenheit = celsius * 1.80 + 32.00
    return [round(kelvin, 5), round(fahrenheit, 5)]
celsius = 36.50
result = convert_temperature(celsius)
print(result)
```

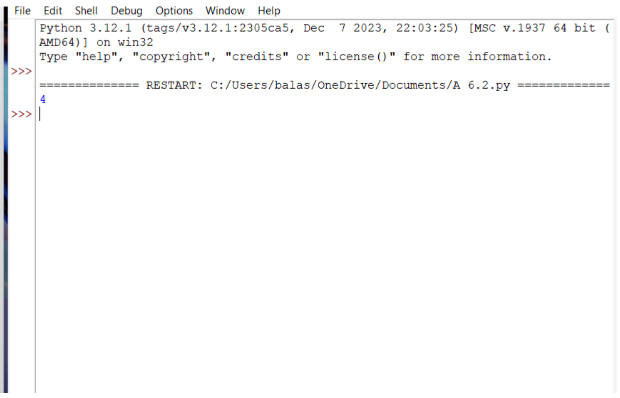


```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
============== RESTART: C:/Users/balas/OneDrive/Documents/A 6.1.py =============
[309.65, 97.7]
>>> 
```

2. Number of Subarrays With LCM Equal to K Given an integer array nums and an integer k, return the number of subarrays of nums where the least common multiple of the subarray's elements is k.A subarray is a contiguous non- empty sequence of elements within an array.The least common multiple of an array is the smallest positive integer that is divisible by all the array elements.

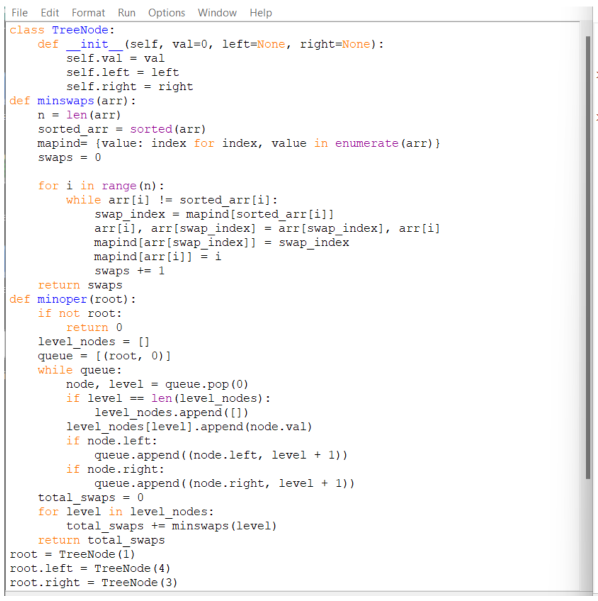


```python
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a
def lcm(a, b):
    return abs(a * b) // gcd(a, b)
def countsub(nums, k):
    count = 0
    n = len(nums)
    for i in range(n):
        lcmcurr = 1
        for j in range(i, n):
            lcmcurr = lcm(lcmcurr, nums[j])
            if lcmcurr == k:
                count += 1
            elif lcmcurr > k:
                break

    return count
nums = [3, 6, 2, 7, 1]
k = 6
result = countsub(nums, k)
print(result)
```

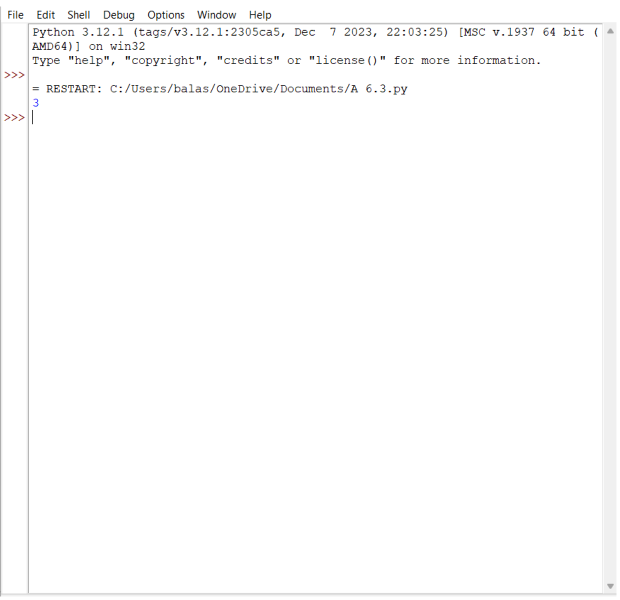


```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
============== RESTART: C:/Users/balas/OneDrive/Documents/A 6.2.py =============
4
>>> 
```

3. . Minimum Number of Operations to Sort a Binary Tree by Level You are given the root of a binary tree with unique values.In one operation, you can choose any two nodes at the same level and swap their values.Return the minimum number of operations needed to make the values at each level sorted in a strictly increasing order. The level of a node is the number of edges along the path between it and the root node.

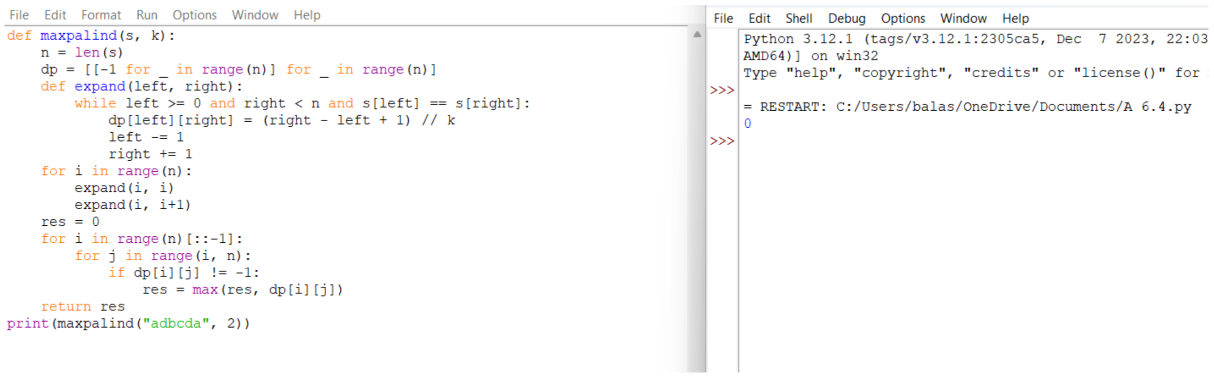


```python
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right
def minswaps(arr):
    n = len(arr)
    sorted_arr = sorted(arr)
    mapind= {value: index for index, value in enumerate(arr)}
    swaps = 0

    for i in range(n):
        while arr[i] != sorted_arr[i]:
            swap_index = mapind[sorted_arr[i]]
            arr[i], arr[swap_index] = arr[swap_index], arr[i]
            mapind[arr[swap_index]] = swap_index
            mapind[arr[i]] = i
            swaps += 1
    return swaps
def minoper(root):
    if not root:
        return 0
    level_nodes = []
    queue = [(root, 0)]
    while queue:
        node, level = queue.pop(0)
        if level == len(level_nodes):
            level_nodes.append([])
        level_nodes[level].append(node.val)
        if node.left:
            queue.append((node.left, level + 1))
        if node.right:
            queue.append((node.right, level + 1))
    total_swaps = 0
    for level in level_nodes:
        total_swaps += minswaps(level)
    return total_swaps
root = TreeNode(1)
root.left = TreeNode(4)
root.right = TreeNode(3)
```



```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:/Users/balas/OneDrive/Documents/A 6.3.py
3
>>> 
```

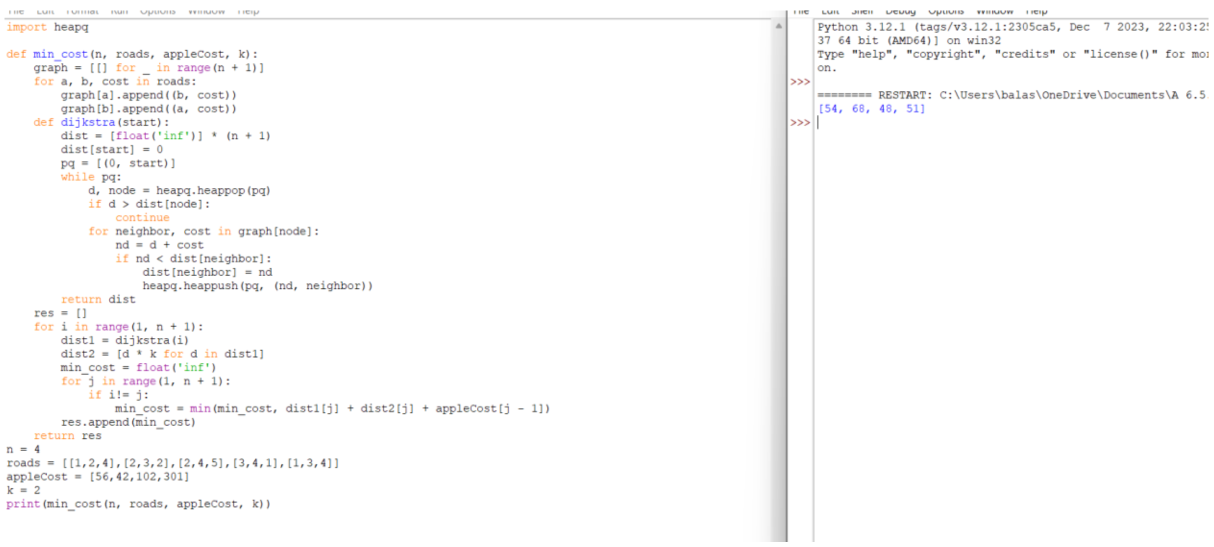
4. . Maximum Number of Non-overlapping Palindrome Substrings You are given a string s and a positive integer k.Select a set of non-overlapping substrings from the string s that satisfy the following conditions: ● The length of each substring is at least k. ● Each substring is a palindrome. Return the maximum number of substrings in an optimal selection.A substring is a contiguous sequence of characters within a string.


File Edit Format Run Options Window Help
File Edit Shell Debug Options Window Help

```python
def maxpalind(s, k):
    n = len(s)
    dp = [[-1 for _ in range(n)] for _ in range(n)]
    def expand(left, right):
        while left >= 0 and right < n and s[left] == s[right]:
            dp[left][right] = (right - left + 1) // k
            left -= 1
            right += 1
    for i in range(n):
        expand(i, i)
        expand(i, i+1)
    res = 0
    for i in range(n)[::-1]:
        for j in range(i, n):
            if dp[i][j] != -1:
                res = max(res, dp[i][j])
    return res
print(maxpalind("adbcda", 2))
```

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
>>>
= RESTART: C:/Users/balas/OneDrive/Documents/A 6.4.py
0
>>>
```

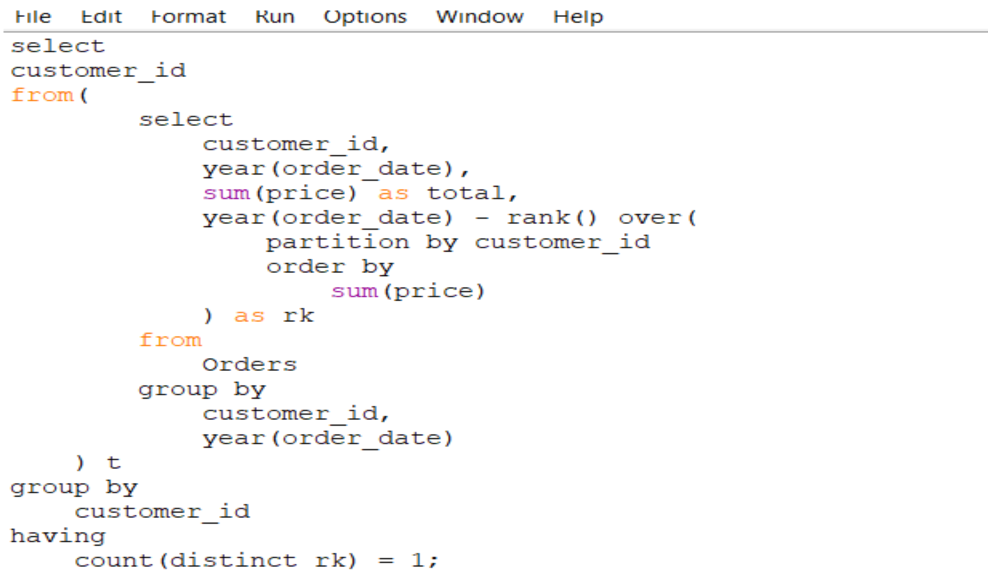
5. Minimum Cost to Buy Apples You are given a positive integer n representing n cities numbered from 1 to n. You are also given a 2D array roads, where roads[i] = [ai, bi, costi] indicates that there is a bidirectional road between cities ai and bi with a cost of traveling equal to costi. You can buy apples in any city you want, but some cities have different costs to buy apples. You are given the array appleCost where appleCost[i] is the cost of buying one apple from city i. You start at some city, traverse through various roads, and eventually buy exactly one apple from any city. After you buy that apple, you have to return back to the city you started at, but now the cost of all the roads will be multiplied by a given factor k. Given the integer k, return an array answer of size n where answer[i] is the minimum total cost to buy an apple if you start at city i.



```python
import heapq

def min_cost(n, roads, appleCost, k):
    graph = [[] for _ in range(n + 1)]
    for a, b, cost in roads:
        graph[a].append((b, cost))
        graph[b].append((a, cost))
    def dijkstra(start):
        dist = [float('inf')] * (n + 1)
        dist[start] = 0
        pq = [(0, start)]
        while pq:
            d, node = heapq.heappop(pq)
            if d > dist[node]:
                continue
            for neighbor, cost in graph[node]:
                nd = d + cost
                if nd < dist[neighbor]:
                    dist[neighbor] = nd
                    heapq.heappush(pq, (nd, neighbor))
        return dist
    res = []
    for i in range(1, n + 1):
        dist1 = dijkstra(i)
        dist2 = [d * k for d in dist1]
        min_cost = float('inf')
        for j in range(1, n + 1):
            if i!= j:
                min_cost = min(min_cost, dist1[j] + dist2[j] + appleCost[j - 1])
        res.append(min_cost)
    return res
n = 4
roads = [[1,2,4],[2,3,2],[2,4,5],[3,4,1],[1,3,4]]
appleCost = [56,42,102,301]
k = 2
print(min_cost(n, roads, appleCost, k))
```

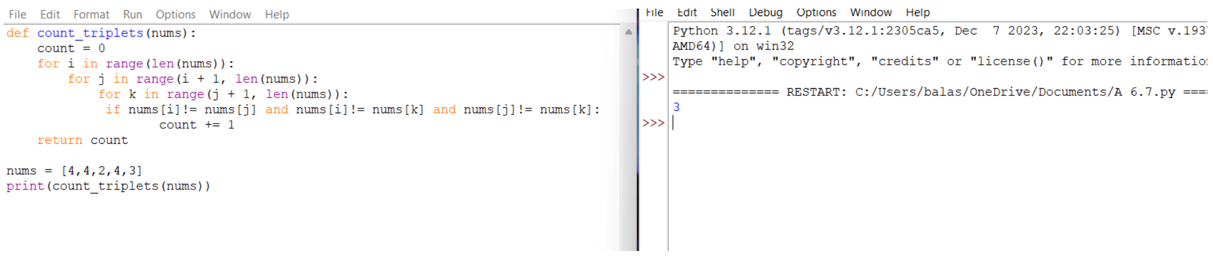
```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:2
37 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for mo
on.
>>>
======== RESTART: C:\Users\balas\OneDrive\Documents\A 6.5
[54, 68, 48, 51]
>>>
```

6. Customers With Strictly Increasing Purchases SQL Schema Table: Orders +--------------+------+ | Column Name | Type | +--------------+------+ | order_id | int | | customer_id | int | | order_date | date | | price | int | +--------------+------+ order_id is the primary key for this table. Each row contains the id of an order, the id of customer that ordered it, the date of the order, and its price. Write an SQL query to report the IDs of the customers with the total purchases strictly increasing yearly



```
select
customer_id
from(
          select
               customer_id,
               year(order_date),
               sum(price) as total,
               year(order_date) - rank() over(
                    partition by customer_id
                    order by
                         sum(price)
               ) as rk
          from
               Orders
          group by
               customer_id,
               year(order_date)
     ) t
group by
     customer_id
having
     count(distinct rk) = 1;
```

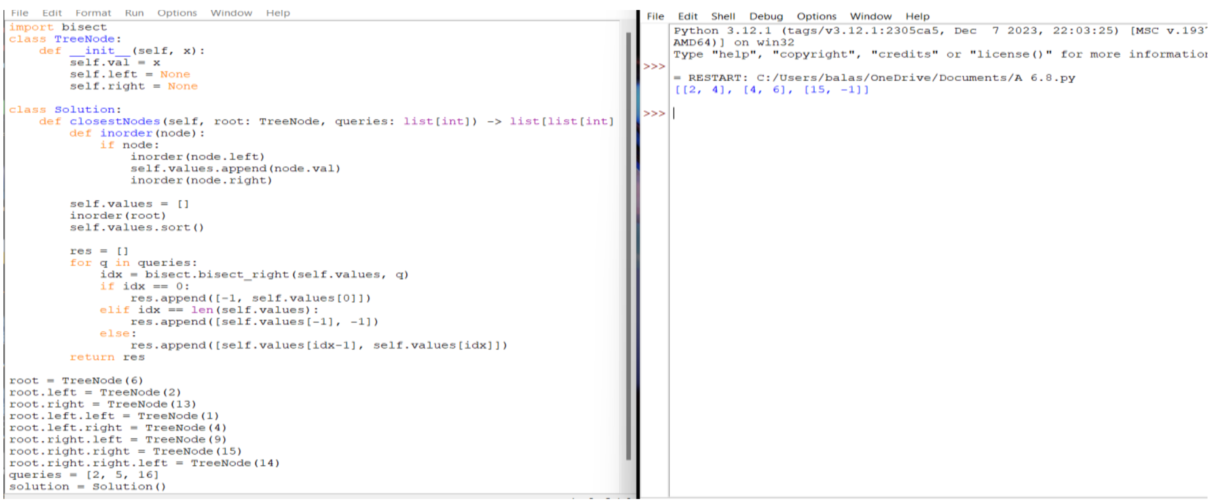
7. Number of Unequal Triplets in Array You are given a 0-indexed array of positive integers nums. Find the number of triplets (i, j, k) that meet the following conditions: ● 0 <= i < j < k < nums.length ● nums[i], nums[j], and nums[k] are pairwise distinct. ○ In other words, nums[i] != nums[j], nums[i] != nums[k], and nums[j] != nums[k]. Return the number of triplets that meet the condi


File Edit Format Run Options Window Help
File Edit Shell Debug Options Window Help

```
def count_triplets(nums):
    count = 0
    for i in range(len(nums)):
        for j in range(i + 1, len(nums)):
            for k in range(j + 1, len(nums)):
             if nums[i]!= nums[j] and nums[i]!= nums[k] and nums[j]!= nums[k]:
                    count += 1
    return count

nums = [4,4,2,4,3]
print(count_triplets(nums))
```

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.193
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informatio
>>>
============== RESTART: C:/Users/balas/OneDrive/Documents/A 6.7.py ===
3
>>> |
```

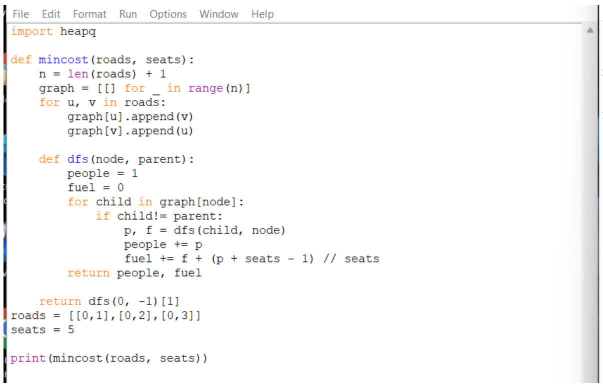
8. Closest Nodes Queries in a Binary Search Tree You are given the root of a binary search tree and an array queries of size n consisting of positive integers. Find a 2D array answer of size n where answer[i] = [mini, maxi]: ● mini is the largest value in the tree that is smaller than or equal to queries[i]. If a such value does not exist, add -1 instead. ● maxi is the smallest value in the tree that is greater than or equal to queries[i]. If a such value does not exist, add -1 instead


File Edit Format Run Options Window Help
File Edit Shell Debug Options Window Help

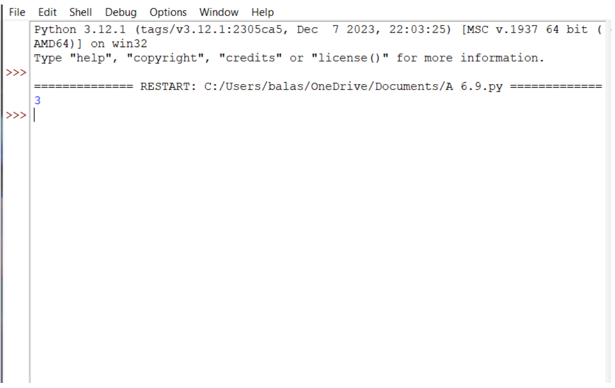
```
import bisect
class TreeNode:
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None

class Solution:
    def closestNodes(self, root: TreeNode, queries: list[int]) -> list[list[int]
        def inorder(node):
            if node:
                inorder(node.left)
                self.values.append(node.val)
                inorder(node.right)

        self.values = []
        inorder(root)
        self.values.sort()

        res = []
        for q in queries:
            idx = bisect.bisect_right(self.values, q)
            if idx == 0:
                res.append([-1, self.values[0]])
            elif idx == len(self.values):
                res.append([self.values[-1], -1])
            else:
                res.append([self.values[idx-1], self.values[idx]])
        return res

root = TreeNode(6)
root.left = TreeNode(2)
root.right = TreeNode(13)
root.left.left = TreeNode(1)
root.left.right = TreeNode(4)
root.right.left = TreeNode(9)
root.right.right = TreeNode(15)
root.right.right.left = TreeNode(14)
queries = [2, 5, 16]
solution = Solution()
```

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.193
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informatio
>>>
= RESTART: C:/Users/balas/OneDrive/Documents/A 6.8.py
[[2, 4], [4, 6], [15, -1]]
>>> |
```

9. Minimum Fuel Cost to Report to the Capital There is a tree (i.e., a connected, undirected graph with no cycles) structure country network consisting of n cities numbered from 0 to n - 1 and exactly n - 1 roads. The capital city is city 0. You are given a 2D integer array roads where roads[i] = [ai, bi] denotes that there exists a bidirectional road connecting cities ai and bi. There is a meeting for the representatives of each city. The meeting is in the capital city.There is a car in each city. You are given an integer seats that indicates the number of seats in each car.A representative can use the car in their city to travel or change the car and ride with another representative. The cost of traveling between two cities is one liter

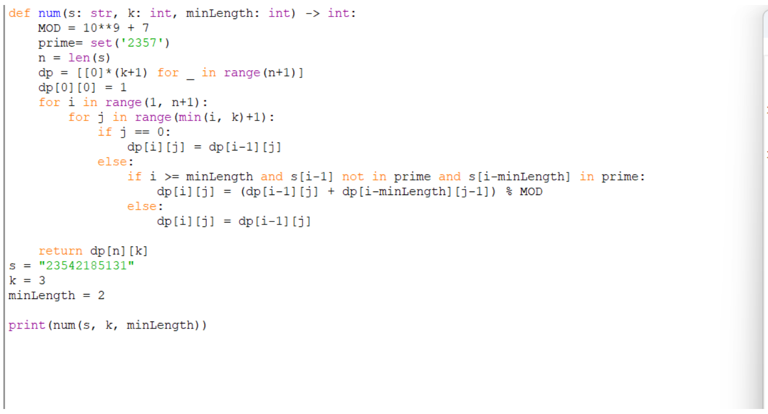


```
import heapq

def mincost(roads, seats):
    n = len(roads) + 1
    graph = [[] for _ in range(n)]
    for u, v in roads:
        graph[u].append(v)
        graph[v].append(u)

    def dfs(node, parent):
        people = 1
        fuel = 0
        for child in graph[node]:
            if child!= parent:
                p, f = dfs(child, node)
                people += p
                fuel += f + (p + seats - 1) // seats
        return people, fuel

    return dfs(0, -1)[1]
roads = [[0,1],[0,2],[0,3]]
seats = 5

print(mincost(roads, seats))
```

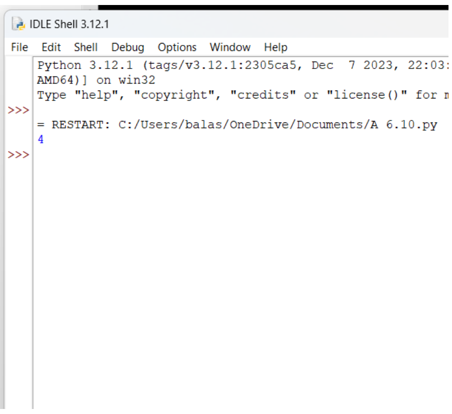


```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:/Users/balas/OneDrive/Documents/A 6.9.py =============
3
>>>
```

10. Number of Beautiful Partitions You are given a string s that consists of the digits '1' to '9' and two integers k and minLength. A partition of s is called beautiful if: ● s is partitioned into k non-intersecting substrings. ● Each substring has a length of at least minLength. ● Each substring starts with a prime digit and ends with a non-prime digit. Prime digits are '2', '3', '5', and '7', and the rest of the digits are non-prime. Return the number of beautiful partitions of s. Since the answer may be very large, return it modulo 109 + 7.A substring is a contiguous sequence of characters within a string



```
def num(s: str, k: int, minLength: int) -> int:
    MOD = 10**9 + 7
    prime= set('2357')
    n = len(s)
    dp = [[0]*(k+1) for _ in range(n+1)]
    dp[0][0] = 1
    for i in range(1, n+1):
        for j in range(min(i, k)+1):
            if j == 0:
                dp[i][j] = dp[i-1][j]
            else:
                if i >= minLength and s[i-1] not in prime and s[i-minLength] in prime:
                    dp[i][j] = (dp[i-1][j] + dp[i-minLength][j-1]) % MOD
                else:
                    dp[i][j] = dp[i-1][j]

    return dp[n][k]
s = "23542185131"
k = 3
minLength = 2

print(num(s, k, minLength))
```



```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for m
>>>
= RESTART: C:/Users/balas/OneDrive/Documents/A 6.10.py
4
>>>
```