

Programiz  
Python Online Compiler

Careers for those who love  
UNITEDHEALTH GROUP | Optum | UnitedHealthcare

Programiz PRO >

main.py

Save

Run

```
1 # Definition for a binary tree node.
2 class TreeNode:
3     def __init__(self, val=0, left=None, right=None):
4         self.val = val
5         self.left = left
6         self.right = right
7
8 def sortedArrayToBST(nums):
9     if not nums:
10         return None
11
12     # Helper function to convert a subarray to a BST
13     def convertToBST(left, right):
14         if left > right:
15             return None
16
17         # Choose the middle element as the root
18         mid = (left + right) // 2
19         node = TreeNode(nums[mid])
20
21         # Recursively construct the left and right subtrees
22         node.left = convertToBST(left, mid - 1)
23         node.right = convertToBST(mid + 1, right)
24
25     return node
```

Output

Clear

[0, -10, 5, None, -3, None, 9]

=== Code Execution Successful ===

Waiting for apex.go.sonobi.com...

28°

Search

ENG IN


08:51  
18-06-2024

arms.sse.saveetha.com | GitHub | The Latest Login in 2020 - Lect... | ChatGPT | Online Python Compiler (Interp... | +

programiz.com/python-programming/online-compiler/

Programiz  
Python Online Compiler



Triple Robust Bolt  
Locking for  
Enhanced Safety



Godrej Smart Lock | Catus Touch Plus | Digital Lock f...  
★★★★☆ 170  
-32% ₹11,946<sup>00</sup> ₹17,499.00 ✓prime

Programiz PRO >

main.py

 Save Run

```
1 def findSubstrings(words):
2     result = []
3     n = len(words)
4
5     for i in range(n):
6         for j in range(n):
7             if i != j and words[i] in words[j]:
8                 result.append(words[i])
9                 break # No need to check further for this word
10
11     return result
12
13 # Example usage
14 words = ["mass", "as", "hero", "superhero"]
15 print(findSubstrings(words)) # Output: ["as", "hero"]
16
```

Output


Clear

```
['as', 'hero']

=== Code Execution Successful ===
```

28°

Search



ENG IN 08:53 18-06-2024

Programiz  
Python Online Compiler

Godrej Smart Lock I Catus Touch Plus I Digital...  
₹11,946.00 ~~₹17,499.00~~

Programiz PRO >

main.py

```
1 def wiggleSort(nums):  
2     nums.sort()  
3     n = len(nums)  
4     mid = (n + 1) // 2  
5     left = nums[:mid][::-1]  
6     right = nums[mid:][::-1]  
7  
8     nums[::2] = left  
9     nums[1::2] = right  
10  
11 # Example usage  
12 nums1 = [1, 5, 1, 1, 6, 4]  
13 wiggleSort(nums1)  
14 print(nums1) # Output: [1, 6, 1, 5, 1, 4] or any valid wiggle sort arrangement  
15  
16 nums2 = [1, 3, 2, 2, 3, 1]  
17 wiggleSort(nums2)  
18 print(nums2) # Output: [2, 3, 1, 3, 1, 2] or any valid wiggle sort arrangement  
19
```

Output

Clear

```
[1, 6, 1, 5, 1, 4]  
[2, 3, 1, 3, 1, 2]  
  
=== Code Execution Successful ===
```

28°

ENG  
IN

08:54  
18-06-2024



main.py



Save

Run

Output

Clear

```
1 from collections import deque
2
3 def updateMatrix(mat):
4     rows, cols = len(mat), len(mat[0])
5     distances = [[float('inf')] * cols for _ in range(rows)]
6     queue = deque()
7
8     # Initialize the queue with all positions of 0s and set their distance to 0
9     for r in range(rows):
10         for c in range(cols):
11             if mat[r][c] == 0:
12                 distances[r][c] = 0
13                 queue.append((r, c))
14
15     # Directions for moving up, down, left, right
16     directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]
17
18     # Perform BFS
19     while queue:
20         r, c = queue.popleft()
21
22         for dr, dc in directions:
23             new_r, new_c = r + dr, c + dc
24
25             if 0 <= new_r < rows and 0 <= new_c < cols:
26                 if distances[new_r][new_c] > distances[r][c] + 1:
```

```
[[0, 0, 0], [0, 1, 0], [0, 0, 0]]
[[0, 0, 0], [0, 1, 0], [1, 2, 1]]

=== Code Execution Successful ===
```



```
main.py
1 def min_length_after_removals(nums):
2     n = len(nums)
3     left, right = 0, n - 1
4     pairs_removed = 0
5
6     while left < right:
7         # Check if we can form a pair
8         if nums[left] < nums[right]:
9             # Remove the pair
10            pairs_removed += 1
11            left += 1
12            right -= 1
13        else:
14            # If we cannot form a pair, move the right pointer to the left
15            right -= 1
16
17        # The remaining length is the original length minus twice the number of pairs removed
18        return n - 2 * pairs_removed
19
20 # Example usage
21 nums = [1, 2, 3, 4]
22 print(min_length_after_removals(nums)) # Output: 0
23
```

0  
=== Code Execution Successful ===