

RunDebugStopShareSaveBeautify

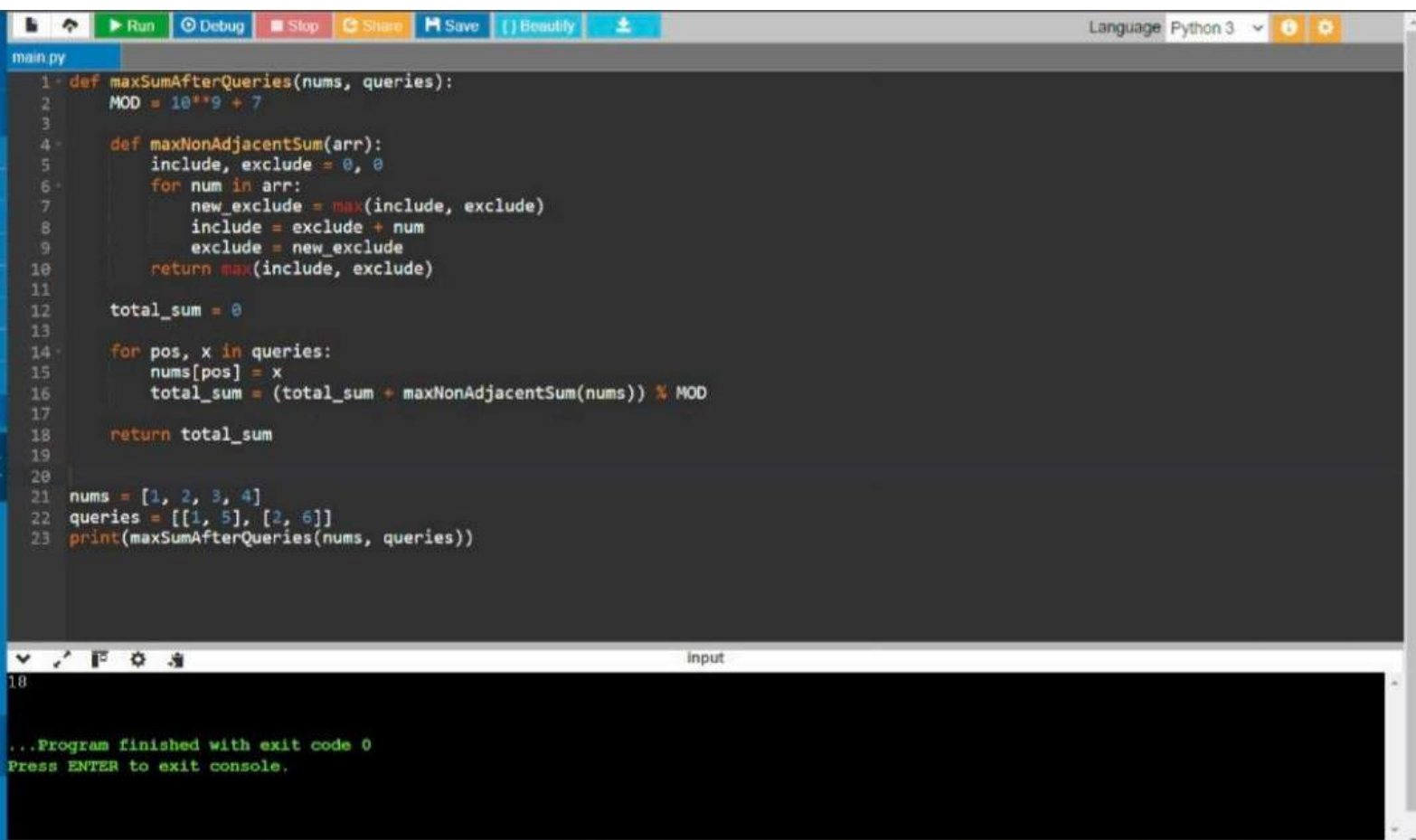
LanguagePython 3

```
main.py
1 import heapq
2
3
4 import math
5
6 def kClosest(points, k):
7     def euclidean_distance(point):
8         return math.sqrt(point[0]**2 + point[1]**2)
9
10    min_heap = [(euclidean_distance(point), point) for point in points]
11    heapq.heapify(min_heap)
12
13    return [heapq.heappop(min_heap)[1] for _ in range(k)]
14
15 # Example usage:
16 points = [[1,3],[-2,2]]
17 k = 1
18 print(kClosest(points, k))
```

Input

```
[[1, 3], [-2, 2]]

...Program finished with exit code 0
Press ENTER to exit console.
```



The image shows a code editor window with a dark theme. The top toolbar includes icons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language is set to Python 3. The editor displays a file named 'main.py' containing a Python script. The script defines a function 'maxSumAfterQueries' which uses a helper function 'maxNonAdjacentSum' to calculate the maximum sum of a subset of numbers, excluding a specific element based on a query. The script initializes 'nums' as [1, 2, 3, 4] and 'queries' as [[1, 5], [2, 6]], then prints the result of 'maxSumAfterQueries(nums, queries)'. Below the editor, a console window shows the output: '...Program finished with exit code 0' and 'Press ENTER to exit console.'

```
1 def maxSumAfterQueries(nums, queries):
2     MOD = 10**9 + 7
3
4     def maxNonAdjacentSum(arr):
5         include, exclude = 0, 0
6         for num in arr:
7             new_exclude = max(include, exclude)
8             include = exclude + num
9             exclude = new_exclude
10        return max(include, exclude)
11
12    total_sum = 0
13
14    for pos, x in queries:
15        nums[pos] = x
16        total_sum = (total_sum + maxNonAdjacentSum(nums)) % MOD
17
18    return total_sum
19
20
21 nums = [1, 2, 3, 4]
22 queries = [[1, 5], [2, 6]]
23 print(maxSumAfterQueries(nums, queries))
```

...Program finished with exit code 0
Press ENTER to exit console.

Run

Debug

Stop

Share

Save

Beautify

Language Python 3

```
main.py
1 def maxSubarraySumCircular(nums):
2     def kadane(arr):
3         current_sum = max_sum = arr[0]
4         for num in arr[1:]:
5             current_sum = max(num, current_sum + num)
6             max_sum = max(max_sum, current_sum)
7         return max_sum
8
9     max_kadane = kadane(nums)
10    total_sum = sum(nums)
11    min_kadane = kadane([-num for num in nums])
12
13    if total_sum + min_kadane == 0:
14        return max_kadane
15    else:
16        return max(max_kadane, total_sum + min_kadane)
17
18
19 nums = [1, -2, 3, -2]
20 print(maxSubarraySumCircular(nums))
21
22
```

Input

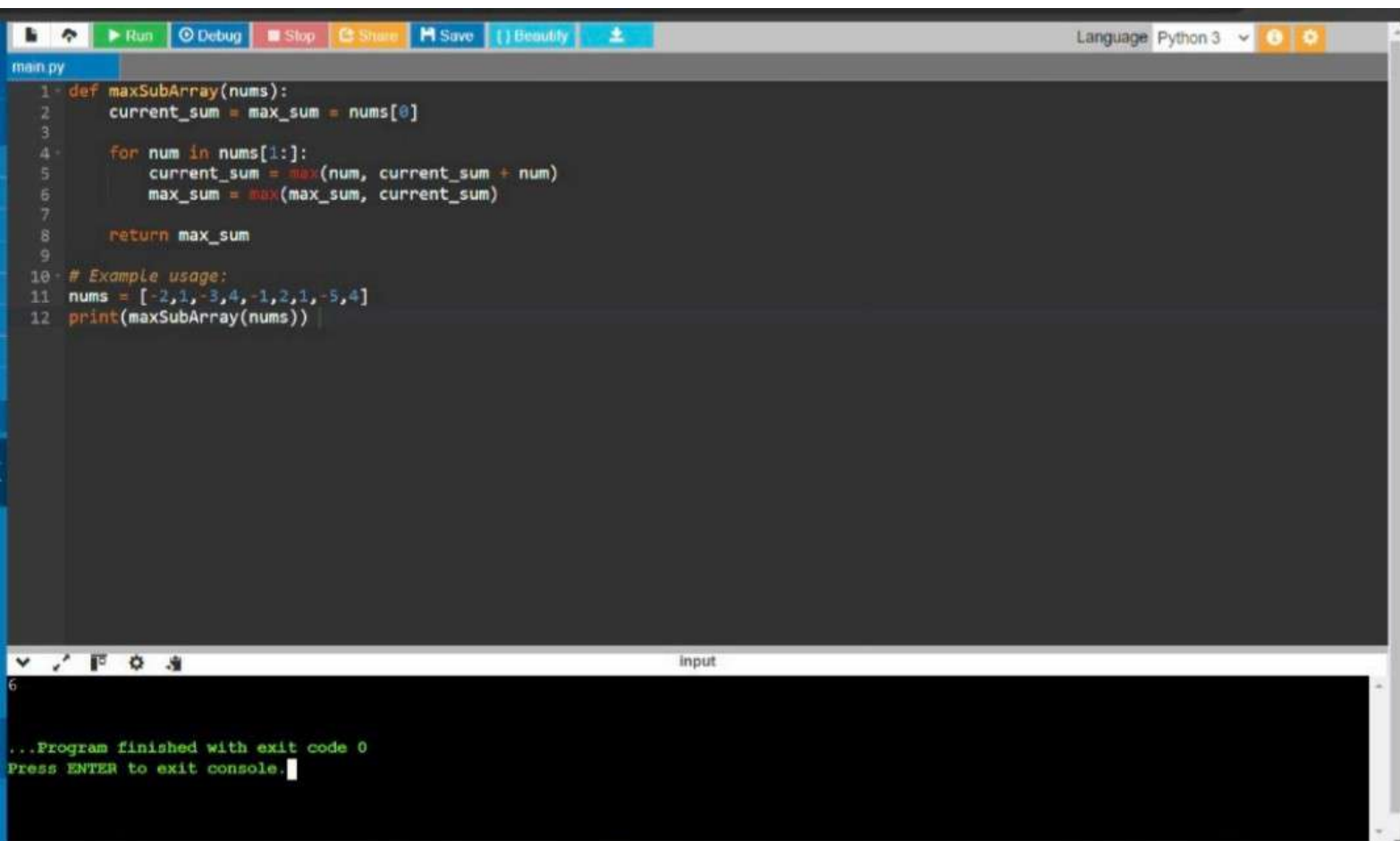
3

...Program finished with exit code 0
Press ENTER to exit console.

The image shows a code editor window with a toolbar at the top containing icons for Run, Debug, Stop, Share, Save, and a Beautify button. The language is set to Python 3. The editor displays a file named 'main.py' with the following Python code:

```
1 class TreeNode:
2     def __init__(self, val=0, left=None, right=None):
3         self.val = val
4         self.left = left
5         self.right = right
6
7 def constructMaximumBinaryTree(nums):
8     if not nums:
9         return None
10
11     max_val = max(nums)
12     max_index = nums.index(max_val)
13
14     root = TreeNode(max_val)
15     root.left = constructMaximumBinaryTree(nums[:max_index])
16     root.right = constructMaximumBinaryTree(nums[max_index + 1:])
17
18     return root
19
20 # Example usage:
21 nums = [3,2,1,6,0,5]
22 root = constructMaximumBinaryTree(nums)
23
```

Below the code editor is an 'Input' section, which is currently empty. At the very bottom, a console window shows the message: '...Program finished with exit code 0' and 'Press ENTER to exit console.'



The image shows a Python IDE interface. At the top, there is a toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language is set to Python 3. The main editor window displays a file named 'main.py' containing the following Python code:

```
1 def maxSubArray(nums):
2     current_sum = max_sum = nums[0]
3
4     for num in nums[1:]:
5         current_sum = max(num, current_sum + num)
6         max_sum = max(max_sum, current_sum)
7
8     return max_sum
9
10 # Example usage:
11 nums = [-2,1,-3,4,-1,2,1,-5,4]
12 print(maxSubArray(nums))
```

Below the editor, there is an 'Input' section and a console window. The console shows the output of the program:

```
6
...Program finished with exit code 0
Press ENTER to exit console.
```

The image shows a Python IDE window with a toolbar at the top containing icons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to Python 3. The editor displays a file named `main.py` with the following code:

```
1 def lastStringBeforeEmpty(s):
2     last_non_empty = s
3
4     while s:
5         removed = set()
6         new_s = []
7         for char in s:
8             if char not in removed:
9                 removed.add(char)
10            else:
11                new_s.append(char)
12        last_non_empty = s
13        s = "".join(new_s)
14
15    return last_non_empty
16
17 # Example usage:
18 s = "aabcbbca"
19 print(lastStringBeforeEmpty(s))
20
```

Below the editor is an input field containing the text "ba". At the bottom is a console window showing the output:

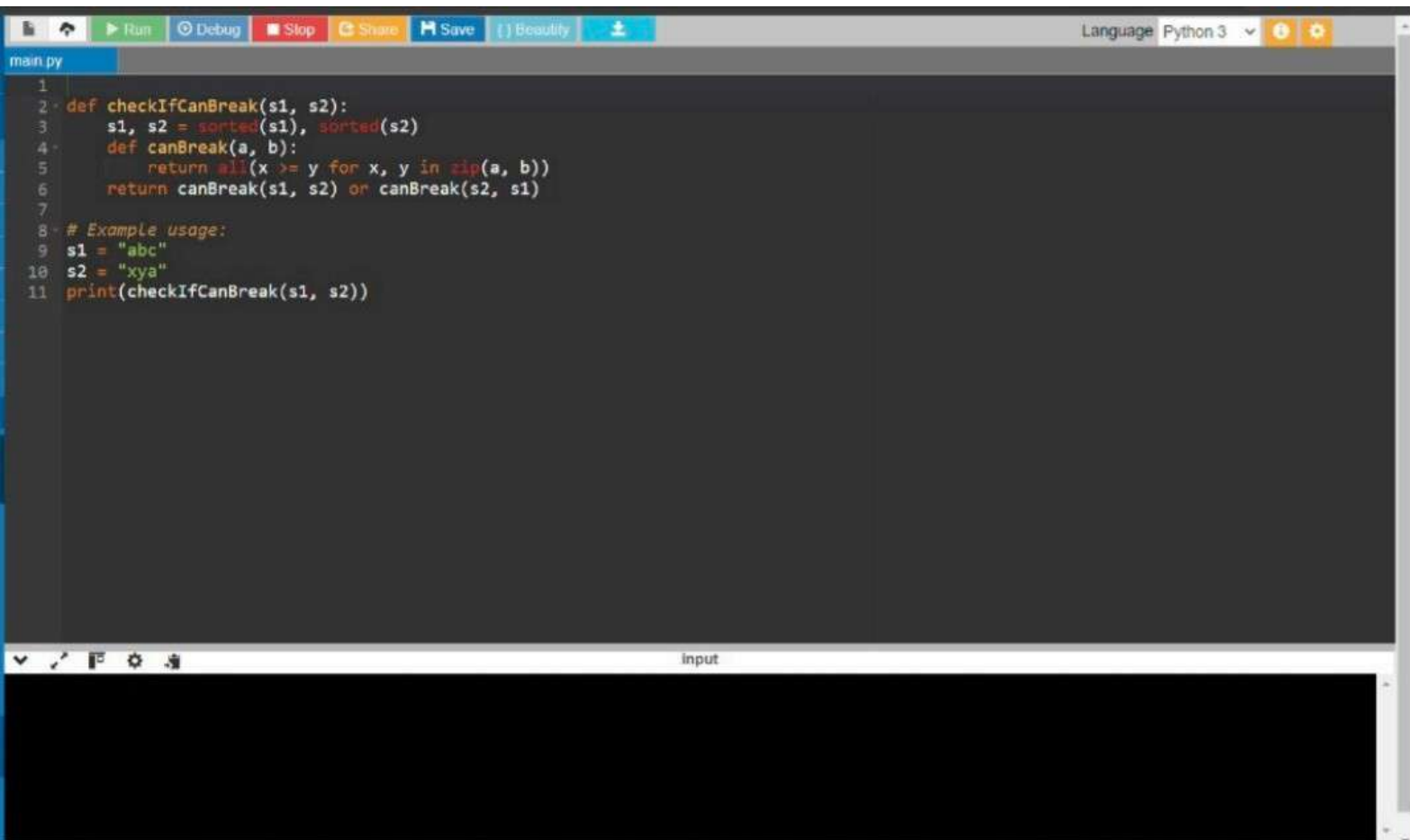
```
...Program finished with exit code 0
Press ENTER to exit console.
```

The image shows a Python IDE window with a toolbar at the top containing icons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to Python 3. The editor displays a file named `main.py` with the following code:

```
1 def minimizeStringValue(s):
2     from collections import defaultdict
3     char_count = defaultdict(int)
4     result = []
5
6     for char in s:
7         if char == '?':
8             for c in 'abcdefghijklmnopqrstuvwxyz':
9                 if char_count[c] == 0:
10                    result.append(c)
11                    char_count[c] += 1
12                    break
13             else:
14                 result.append(char)
15                 char_count[char] += 1
16
17     return "".join(result)
18
19 # Example usage:
20 s = "ab?ac?"
21 print(minimizeStringValue(s))
```

Below the editor is an input field with the text `abcaacd`. At the bottom, a console window shows the output:

```
...Program finished with exit code 0
Press ENTER to exit console.
```



The image shows a screenshot of a Python IDE. The top toolbar contains buttons for Run, Debug, Stop, Share, Save, and a Beautify button. The language is set to Python 3. The code editor displays the following Python code:

```
1
2 def checkIfCanBreak(s1, s2):
3     s1, s2 = sorted(s1), sorted(s2)
4     def canBreak(a, b):
5         return all(x >= y for x, y in zip(a, b))
6     return canBreak(s1, s2) or canBreak(s2, s1)
7
8 # Example usage:
9 s1 = "abc"
10 s2 = "xya"
11 print(checkIfCanBreak(s1, s2))
```

Below the code editor is an input/output window with the label "input". The window is currently empty.

RunDebugStopShareSaveBeautify

Language Python 3

main.py

```
1 class UnionFind:
2     def __init__(self, size):
3         self.parent = list(range(size))
4
5     def find(self, x):
6         if self.parent[x] != x:
7             self.parent[x] = self.find(self.parent[x])
8         return self.parent[x]
9
10    def union(self, x, y):
11        rootX = self.find(x)
12        rootY = self.find(y)
13        if rootX != rootY:
14            self.parent[rootY] = rootX
15
16    def smallestStringWithSwaps(s, pairs):
17        n = len(s)
18        uf = UnionFind(n)
19
20        for x, y in pairs:
21            uf.union(x, y)
22
23        from collections import defaultdict
24        groups = defaultdict(list)
25
26        for i in range(n):
27            groups[uf.find(i)].append(i)
```

input

bad

...Program finished with exit code 0
Press ENTER to exit console.

The image shows a Python IDE with a toolbar at the top containing icons for Run, Debug, Stop, Share, Save, and a Beautify button. The language is set to Python 3. The main editor window displays a file named 'main.py' with the following code:

```
1 def findMedianSortedArrays(nums1, nums2):
2     if len(nums1) > len(nums2): nums1, nums2 = nums2, nums1
3     m, n = len(nums1), len(nums2)
4     imin, imax, half = 0, m, (m + n + 1) // 2
5
6     while imin <= imax:
7         i = (imin + imax) // 2
8         j = half - i
9         if i < m and nums1[i] < nums2[j - 1]:
10             imin = i + 1
11         elif i > 0 and nums1[i - 1] > nums2[j]:
12             imax = i - 1
13         else:
14             max_left = max(nums1[i - 1] if i > 0 else float('-inf'), nums2[j - 1] if j > 0 else float('-inf'))
15             if (m + n) % 2:
16                 return max_left
17             min_right = min(nums1[i] if i < m else float('inf'), nums2[j] if j < n else float('inf'))
18             return (max_left + min_right) / 2
19
20 # Example usage:
21 nums1 = [1, 3]
22 nums2 = [2]
23 print(findMedianSortedArrays(nums1, nums2))
```

Below the editor is an 'Input' section with a text area containing the number '2'. At the bottom, a console window shows the output: '...Program finished with exit code 0' and 'Press ENTER to exit console.'