

Our Solution(s)

Run Code

Solution 1

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 # O(n) time | O(n) space
4 def balancedBrackets(string):
5     openingBrackets = "([{"
6     closingBrackets = ")]}"
7     matchingBrackets = {"(": ")", "[": "]", "{": "{"}
8     stack = []
9     for char in string:
10         if char in openingBrackets:
11             stack.append(char)
12         elif char in closingBrackets:
13             if len(stack) == 0:
14                 return False
15             if stack[-1] == matchingBrackets[char]:
16                 stack.pop()
17             else:
18                 return False
19     return len(stack) == 0
20
```

Our Tests

Your Solutions

Run Code

Solution 1

Solution 2

Solution 3

```
1 def balancedBrackets(string):
2     # Write your code here.
3     pass
4
```

Custom Output

Submit Code

```

1  #!/usr/bin/env python
2
3  # Import the necessary modules
4  import sys
5  import os
6  import random
7  import time
8
9  # Define the main function
10 def main():
11     # Get the number of arguments
12     num_args = len(sys.argv)
13
14     # Check if the number of arguments is correct
15     if num_args != 2:
16         print("Usage: %s <number>" % sys.argv[0])
17         sys.exit(1)
18
19     # Get the number from the command line
20     number = sys.argv[1]
21
22     # Check if the number is a valid integer
23     try:
24         int(number)
25     except ValueError:
26         print("Error: %s is not a valid integer." % number)
27         sys.exit(1)
28
29     # Generate a random number
30     random_number = random.randint(1, 100)
31
32     # Print the random number
33     print("Random number: %d" % random_number)
34
35 # Call the main function
36 if __name__ == "__main__":
37     main()

```

Run or submit code when you're ready.