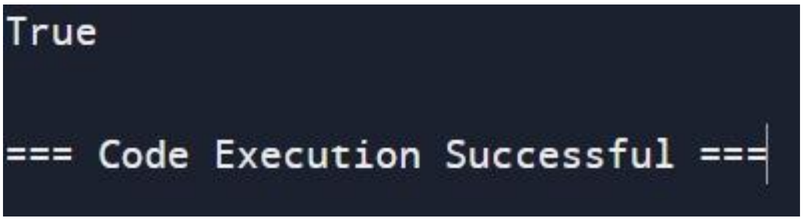
22. Given two strings: s1 and s2 with the same size, check if some permutation of string s1 can break some permutation of string s2 or vice-versa. In other words s2 can break s1 or vice-versa. A string x can break string y (both of size n) if x[i] >= y[i] (in alphabetical order) for all i between 0 and n-1.

Aim: aim of the program is two strings: s1 and s2 with the same size, check if some permutation of string s1 can break some permutation of string s2 or vice-versa. In other words s2 can break s1 or vice-versa. A string x can break string y (both of size n) if x[i] >= y[i] (in alphabetical order) for all i between 0 and n-1.

Program:

```
def check_if_can_break(s1, s2):

    s1_sorted = sorted(s1)

    s2_sorted = sorted(s2)

    if all(s1_char >= s2_char for s1_char, s2_char in zip(s1_sorted, s2_sorted)) or all(s2_char >= s1_char for s1_char, s2_char in zip(s1_sorted, s2_sorted)):

        return True

    else:

        return False

s1 = "abc"

s2 = "xya"

result = check_if_can_break(s1, s2)

print(result)
```

Output:



```
True

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

time complexity:O(nlogn)2