PROGRAM21: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.

Progarm:

```
def check_if_can_break(s1, s2):
    # Sort both strings
    s1_sorted = sorted(s1)
    s2_sorted = sorted(s2)

    # Check if s1 can break s2
    can_s1_break_s2 = all(c1 >= c2 for c1, c2 in zip(s1_sorted, s2_sorted))

    # Check if s2 can break s1
    can_s2_break_s1 = all(c2 >= c1 for c1, c2 in zip(s1_sorted, s2_sorted))

    # Return True if either s1 can break s2 or s2 can break s1
    return can_s1_break_s2 or can_s2_break_s1

# Example usage:
s1 = "abc"
s2 = "xya"
print(check_if_can_break(s1, s2))  # Output: True
```

OUTPUT:

```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA COADS.PYTHON\program 22.py"
True

Process finished with exit code 0
```

TIME COMPLEXITY:
O(n log n)