30th 2023

FIRST:-

Check if strings are rotations of each other or not : : Basic

Given two strings s1 and s2. The task is to check if s2 is a rotated version of the string s1. The characters in the strings are in lowercase.

Example 1:

```
Input:
geeksforgeeks
forgeeksgeeks
Output:
1
Explanation: s1 is geeksforgeeks, s2 is
forgeeksgeeks. Clearly, s2 is a rotated
version of s1 as s2 can be obtained by
left-rotating s1 by 5 units.
```

Example 2:

```
Input:
mightandmagic
andmagicmigth
Output:
0
Explanation: Here with any amount of
rotation s2 can't be obtained by s1.
```

Your Task:

The task is to complete the function **areRotations()** which checks if the two

strings are rotations of each other. The function returns true if string 1 can be obtained by rotating string 2, else it returns false.

Expected Time Complexity: O(N).
Expected Space Complexity: O(N).
Note: N = |s1|.

Constraints:
1 <= |s1|, |s2| <= 10^7
CODE SECTION:-

```
bool areRotations(string s1, string s2)
   {
       // Your code here

       if (s1.length() != s2.length())
       {

           return false;
       }

       string ans = s1 + s1;

       string ans1 = "";

       int j = 0;

       for (int i = 0; i < ans.length() && j < s1.length(); i++)
       {

           if (ans[i] == s2[j])
           {

               ans1 = ans1 + s2[j];

               j++;
           }
       }
       return ans1 == s2 ? 1 : 0;
   }
```

-: DONE FOR THE DAY :-