☆ Anagram Difference

We define an anagram to be a word whose characters can be rearranged to create another word. Given two strings, we want to know the minimum number of characters in either string that we must modify to make the two strings anagrams . If it's not possible to make the two strings anagrams, we consider this number to be -1.

For example:

- *tea* and *ate* are anagrams, so we would need to modify 0 characters.
- *tea* and *toe* are not anagrams, but we can modify 1 character in either string (o → a or a → o) to make them anagrams.
- *act* and *acts* are not anagrams and cannot be converted to anagrams because they contain different numbers of characters.

Function Description

Complete the function *getMinimumDifference* in the editor below. The function must return an array of integers which denote the minimum number of characters in either string that need to be modified to make the two strings anagrams. If it's not possible, return -1.

getMinimumDifference has the following parameter(s):

a: the first string
b: the second string

Constraints

- Each string consists of lowercase characters [a-z].
- $1 \leq n \leq 100$

▶ Input Format for Custom Testing

▼ Sample Case 0

Sample Input For Custom Testing

```
a = {'a','jk','abb','mn','abc'}
b = {'bb','kj','bbc','op','def'
```

Sample Output

```
-1
0
1
2
3
```

Explanation

Given a = [a, jk, abb, mn, abc] and b = [bb, kj, bbc, op, def], we perform the following n = 5 calculations:

- Index 0: a and bb cannot be anagrams because they contain different numbers of characters.
- Index 1: jk and kj are already anagrams because they both contain the same characters at the same frequencies.
- Index 2: abb and bbc differ by one character.
- Index 3: mn and op differ by two characters.
- Index 4: abc and def differ by three characters.

After checking each pair of strings, we return the array [-1, 0, 1, 2, 3] as our answer.

APPDYNAMICS | AppDynamics Java Developer Hiring Test April 2... | 03s to test end | 7/8 Attempted

tour. Start tour

Saving draft.. | Original code | Java 8

```
1  import ↔;
6
7  public class Solution {
8
9
10     /*
11      * Complete the getMinimumDifference function below.
12      */
13     private  static int[] getMinimumDifference(String[] a, String[] b) {
14                 int n=a.length;
15                 int ans[]=new int[n];
16                 for(int i=0;i<n;i++) {
17                     String s1=a[i];
18                     String s2=b[i];
19                     int result=getAnagramDiff(s1,s2);
20                     ans[i]=result;
21 //                 System.out.println(result);
22             }
23                 return ans;
24             }
25
26   static int getAnagramDiff(String s1, String s2)
```

```
            int s2len=s2.length();
            if(s1len!=s2len)return-1;
            int char_count[] = new int[26];

            for (int i = 0; i < s1.length(); i++)
                char_count[s1.charAt(i) - 'a']++;
            for (int i = 0; i < s2.length(); i++)
                if (char_count[s2.charAt(i) - 'a']-- <= 0)
                    count++;

            return count;
        }

    private static final Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) throws IOException {
        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));

        int aCount = scanner.nextInt();
        scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])*");

        String[] a = new String[aCount];

        for (int aItr = 0; aItr < aCount; aItr++) {
            String aItem = scanner.nextLine();
            scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])*");
            a[aItr] = aItem;
        }
```

APPDYNAMICS

AppDynamics Java Developer Hiring Test April 2...

03s
to test end

7/8 Attempted

```
        String[] b = new String[bCount];

        for (int bItr = 0; bItr < bCount; bItr++) {
            String bItem = scanner.nextLine();
            scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])*");
            b[bItr] = bItem;
        }

        int[] res = getMinimumDifference(a, b);

        for (int resItr = 0; resItr < res.length; resItr++) {
            bufferedWriter.write(String.valueOf(res[resItr]));

            if (resItr != res.length - 1) {
                bufferedWriter.write("\n");
            }
        }

        bufferedWriter.newLine();

        bufferedWriter.close();

        scanner.close();
    }
}

```

Line: 10 Col: 1

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*