Problem | Submissions | Leaderboard | Discussions | Editorial

Two strings, a and b, are called anagrams if they contain all the same characters in the same frequencies. For this challenge, the test is not case-sensitive. For example, the anagrams of `CAT` are `CAT`, `ACT`, `tac`, `TCA`, `aTC`, and `CtA`.

Function Description

Complete the isAnagram function in the editor.

isAnagram has the following parameters:

- string a: the first string
- string b: the second string

Returns

- boolean: If a and b are case-insensitive anagrams, return true. Otherwise, return false.

Input Format

The first line contains a string a.

The second line contains a string b.

Constraints

- $1 \leq |a|, |b| \leq 50$
- Strings a and b consist of English alphabetic characters.
- The comparison should NOT be case sensitive.

Sample Input 0

```
anagram
margana
```

Sample Output 0

```
Anagrams
```

Explanation 0

Character	Frequency: `anagram`	Frequency: `margana`
`A` or `a`	3	3
`G` or `g`	1	1
`N` or `n`	1	1
`M` or `m`	1	1
`R` or `r`	1	1

The two strings contain all the same letters in the same frequencies, so we print "Anagrams".

Sample Input 1

```
anagramm
marganaa
```

Sample Output 1

```
Not Anagrams
```

Explanation 1

Character	Frequency: `anagramm`	Frequency: `marganaa`
`A` or `a`	3	4
`G` or `g`	1	1
`N` or `n`	1	1
`M` or `m`	2	1
`R` or `r`	1	1

The two strings don't contain the same number of `a`'s and `m`'s, so we print "Not Anagrams".

Sample Input 2

```
Hello
hello
```

Sample Output 2

```
Anagrams
```

Change Theme Language Java 7

```
import java.util.Scanner;

public class Solution {

    static boolean isAnagram(String a, String b) {
        String s1=a.toLowerCase();
        String s2=b.toLowerCase();
         if (a.length() != b.length()) {
            return false;
        }
        int[] n = new int[26];

        for (int i = 0; i < a.length(); i++) {
            char ch = s1.charAt(i);
            n[ch - 'a']++;
        }

        for (int i = 0; i < b.length(); i++) {
            char ch = s2.charAt(i);
            n[ch - 'a']--;

            if (n[ch - 'a'] < 0) {
                return false;
            }
        }
        return true;
```

Line: 7 Col: 28

Upload Code as File Test against custom input Run Code Submit Code

You have earned 10.00 points! 69% 128/150
You are now 22 points away from the 4th star for your java badge.

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6

Compiler Message

Success

Input (stdin) Download

```
anagram
margana
```

Expected Output Download

```
Anagrams
```

Two strings, a and b, are called anagrams if they contain all the same characters in the same frequencies. For this challenge, the test is not case-sensitive. For example, the anagrams of `CAT` are `CAT`, `ACT`, `tac`, `TCA`, `aTC`, and `CtA`.

Function Description

Complete the isAnagram function in the editor.

isAnagram has the following parameters:

- string a: the first string
- string b: the second string

Returns

- boolean: If a and b are case-insensitive anagrams, return true. Otherwise, return false.

Input Format

The first line contains a string a.

The second line contains a string b.

Constraints

- $1 \leq length(a), length(b) \leq 50$
- Strings a and b consist of English alphabetic characters.
- The comparison should NOT be case sensitive.

Sample Input 0

```
anagram
margana
```

Sample Output 0

```
Anagrams
```

Explanation 0

Character	Frequency: `anagram`	Frequency: `margana`
`A` or `a`	3	3
`G` or `g`	1	1
`N` or `n`	1	1
`M` or `m`	1	1
`R` or `r`	1	1

The two strings contain all the same letters in the same frequencies, so we print "Anagrams".

Sample Input 1

```
anagramm
marganaa
```

Sample Output 1

```
Not Anagrams
```

Explanation 1

Character	Frequency: `anagramm`	Frequency: `marganaa`
`A` or `a`	3	4
`G` or `g`	1	1
`N` or `n`	1	1
`M` or `m`	2	1
`R` or `r`	1	1

The two strings don't contain the same number of `a`'s and `m`'s, so we print "Not Anagrams".

Sample Input 2

```
Hello
hello
```

Sample Output 2

```
Anagrams
```

Change Theme Language Java 7

```
1  import java.util.Scanner;
2
3  public class Solution {
4
5      static boolean isAnagram(String a, String b) {
6          String s1=a.toLowerCase();
7          String s2=b.toLowerCase();
8           if (a.length() != b.length()) {
9              return false;
10         }
11         int[] n = new int[26];
12
13         for (int i = 0; i < a.length(); i++) {
14             char ch = s1.charAt(i);
15             n[ch - 'a']++;
16         }
17
18         for (int i = 0; i < b.length(); i++) {
19             char ch = s2.charAt(i);
20             n[ch - 'a']--;
21
22             if (n[ch - 'a'] < 0) {
23                 return false;
24             }
25         }
26         return true;
```

Line: 7 Col: 28

Test against custom input

You have earned 10.00 points! 69% 128/150
You are now 22 points away from the 4th star for your java badge.

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Compiler Message

```
Success
```

Input (stdin)

```
1  anagram
2  margana
```

Expected Output

```
1  Anagrams
```