**Java Anagrams**

<https://www.hackerrank.com/challenges/java-anagrams/problem>

Two strings, *a* and *b*, are called anagrams if they contain all the same characters in the same frequencies. For example, the anagrams of CAT are CAT, ACT, TAC, TCA, ATC, and CTA.

Complete the function in the editor. If *a* and *b* are case-insensitive anagrams, print "Anagrams"; otherwise, print "Not Anagrams" instead.

**Input Format**

The first line contains a string denoting *a*.  
The second line contains a string denoting *b*.

**Constraints**

* *1 <= length(a), length(b) <= 50*
* *Strings a and b consist of English alphabetic characters.*
* *The comparison should NOT be case sensitive.*

**Output Format**

Print "Anagrams" if *a* and *b* are case-insensitive anagrams of each other; otherwise, print "Not Anagrams" instead.

**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

**Explanation 2**

| **Character** | **Frequency:**Hello | **Frequency:**hello |
| --- | --- | --- |
| E or e | 1 | 1 |
| H or h | 1 | 1 |
| L or l | 2 | 2 |
| O or o | 1 | 1 |

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