**Problem Statement: Anagram Checker**

An anagram is a word or phrase formed by rearranging the letters of another word or phrase. For example, the word “listen” is an anagram of “silent”. In this problem, you are given two strings and you need to determine if they are anagrams of each other. Here, we would be considering numbers and alphabets to form a valid set of characters to be considered for checking. Rest all the characters should be ignored in both strings. Also, we need to consider the frequency of the characters as well. Example: “abcd” and “dbacc” are not anagrams as the frequency of ‘c’ is not matching in the given strings.

Write a Python function called `is\_anagram` that takes two strings as input and returns a boolean `True` if they are anagrams of each other and Boolean `False` otherwise.

The function should consider the following:

* Ignore any non-alphabetic and non-numeric characters (such as spaces or punctuation) when checking for anagrams. To reiterate, ignore any character which is not [0-9],[a-z],[A-Z].
* Treat uppercase and lowercase letters as the same. The function should be case-insensitive.

**Example:**

Input: "Listen", "Silent"

Output: True

Input: "AbcD", "ddabc"

Output: False

Input: "A bcD", "BAC D"

Output: True

Input: "Hello", "World"

Output: False

**Note:**

* You can assume that the input strings will contain at least one alphabetic or numeric character.
* You are not allowed to use any built-in Python functions or libraries for checking anagrams. You should implement the logic yourself.