VALID ANAGRAM

☆ In this program, we are given two strings s1 and s2 and our job is to return true if s2 is a valid anagram of s1, or false if not.

A valid anagram is a phrase or a sentence that is formed by rearranging the same letters.
Eg : "ave" and "cab"

There are two possible solutions, one using a hashmap | hashtable and the other using no space.

We can create hashmap for the string and check if it satisfies the other one as well. For the solution with no space, we just sort both strings and compare them. If they are equal, that means the same set of alphabets were used.

Pseudocode :

```
validAnagramHashmap (s1, s2) {
    if (s1.length() != s2.length()) {
        return false
    }
    map hm
    for (i = 0 → s1.length()-1) {
```

```
        if (!hm.find(s1[i])) {
            hm[s1[i]] = 1
        } else {
            hm[s1[i]] += 1
        }
    }
    for(i = 0 -> s2.length()-1) {
        if (!hm.find(s2[i])) {
            return false
        } else {
            hm[s2[i]] -= 1
        }
    }
    for(i = 0 -> hm.size()) {
        if (hm[i] != 0) {
            return false
        }
    }
    return true
}
validAnagramSorting(s1, s2) {
    sort(s1)
    sort(s2)
    if(s1 == s2) {
        return true
    } else {
        return false
    }
}
```