**Day 1**

'''

Mr Sayyad is given a list of words. Mr Sayyad observed a similarity among

the list of words, there exists a common subword in every word in the list,

from the index-0 (from the beginning of the word).

Your task is to find out the common subword which is largest.

Input Format:

-------------

Single line space separated integers, the list[].

Output Format:

--------------

Print the string, the largest common word.

Sample Input-1:

---------------

money monkey monday

Sample Output-1:

----------------

mon

Sample Input-2:

---------------

corn cook book clock

Sample Output-2:

----------------

"" //empty string

=== Write your **python** code below ===

***Soln:***

**n=input().split()**

**l=0**

**n.sort()**

**minimum=min(len(n[0]),len(n[-1]))**

**res=""**

**for i in range(0,minimum):**

**if(n[0][i]==n[-1][i]):**

**res=res+n[0][i]**

**else:**

**break**

**print(res)**

/\*Suman is given two words W1 and W2.

His task is to derive W2 from W1, withthe following operations:

- Replace all the occurrences of a letter in W1 with any other letter.

- Repeat the above step to derive W2 from W1.

Your task is to check whether W2 can be derived from W1 or not.

If yes, print 'true', Otherwise print 'false'.

NOTE:

-----

NO two letters in W1 should map to same character

You may assume both W1 and W2 have the same length.

Input Format:

-------------

Line-1: A String,a word W1

Line-2: A String,a word W2

Output Format:

--------------

Print a boolean value

Sample Input-1:

-------------------

paper

title

Sample Output-1:

---------------------

true

Explanation:

----------------

'p' is replaced with 't', 'a' is with 'i', 'e' is with 'l', and 'r' with 'e'.

Sample Input-2:

-------------------

memo

demo

Sample Output-2:

---------------------

false

\*/

My soln:

import *java.util.\**;

*public* *class* Suman {

*public* *static* void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String s1 = sc.next();

        String s2 = sc.next();

        System.out.println(fun(s1, s2));

    }

*public* *static* boolean fun(String s1, String s2) {

        Map<Character, Character> k = new HashMap<Character, Character>();

        if (s1.length() != s2.length()) {

            return false;

        }

        for (int i = 0; i < s1.length(); i++) {

            if (!k.containsKey(s1.charAt(i))) {

                k.put(s1.charAt(i), s2.charAt(i));

            } else {

                if (k.get(s1.charAt(i)) == s2.charAt(i)) {

                    k.put(s1.charAt(i), s2.charAt(i));

                } else {

                    return false;

                }

            }

        }

        return true;

    }

}

Soln:

