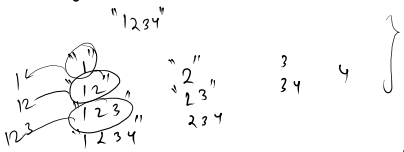


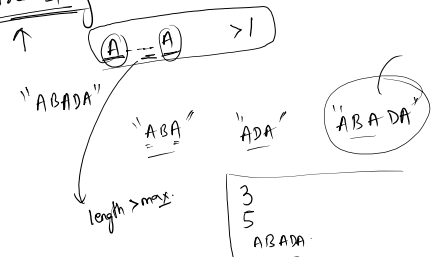
sum of all ss



$$\text{ans} = 1 + 12 + 123 + 1234 + 2 + 23 + 234$$

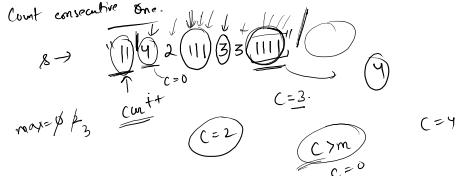
"112" \rightarrow Integer.parseInt("112")

Desired String:

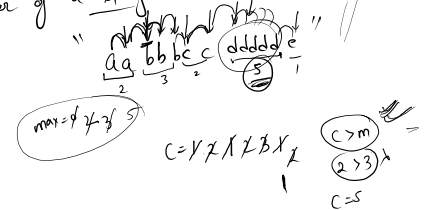


3
5
ABADA

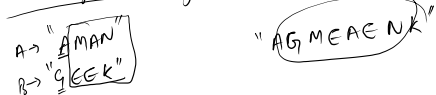
Count consecutive one.



Power of a string



Merge all string:



Long Pressed Name

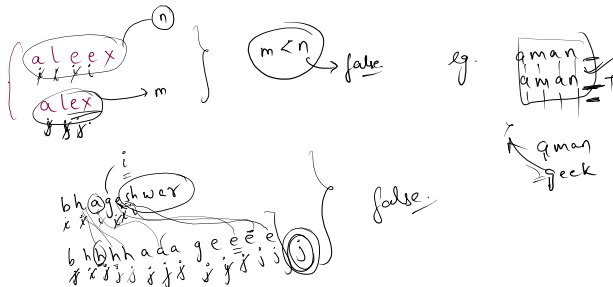
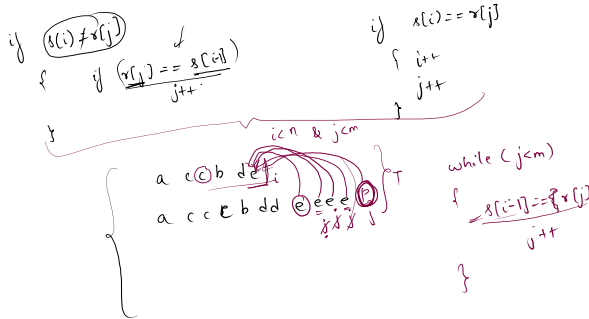
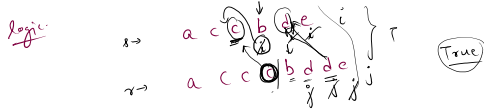
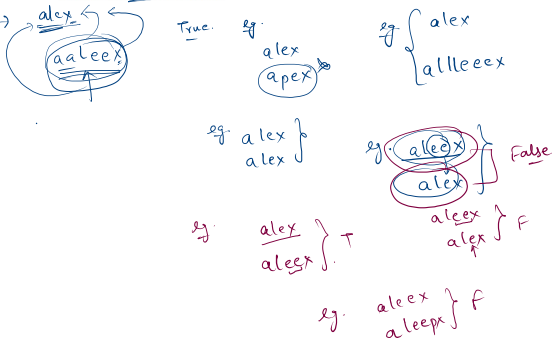
Your friend is typing his name into a keyboard. Sometimes, when typing a character c, the key might get long pressed, and the character will be typed 1 or more times.
You examine the typed characters of the keyboard. Return True if it is possible that it was your friend's name, with some characters (possibly none) being long pressed.

Sample Input 0

alex
aleex

Sample Output 0

true



```

public class Solution {
    public static boolean isMatch(String name, String typed){
        int i = 0;
        int j = 0;

        int n = name.length();
        int m = typed.length();

        if( m < n ){
            return false;
        }

        while(i < n && j < m){
            if(name.charAt(i) == typed.charAt(j)){
                i++;
                j++;
            }
            else if(i > 0 && name.charAt(i-1) == typed.charAt(j)){
                j++;
            }
            else{
                return false;
            }
        }

        while(i < m){
            if(name.charAt(i-1) == typed.charAt(j)){
                j++;
            }
            else{
                return false;
            }
        }

        return i < n ? false : true;
    }
}

```

5 < 6

Handwritten diagram illustrating the matching process for the strings "pypln" and "pypln".

Top row (name): p y p l n (with indices 0, 1, 2, 3, 4 above)

Bottom row (typed): p y p l n (with indices 0, 1, 2, 3, 4 below)

Annotations:

- $n=6$ (circled)
- $i=5$ (circled)
- $m=8$ (circled)
- $j=8$ (circled)
- Red arrows indicate the progression of pointers i and j.

```

if(i < n){
    // if there is some char remaining in first string and 2nd is completed
    return false;
}
else{
    return true;
}

// return i < n ? false : true;
// return i == n ? true : false;

// return i == n;
}

```

Is it a Pangram String?

smaller case
upper case.

the quick Brown fox jumps over the lazy dog Aman

if (ch != ' ')

c = 4+

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | | | | | | |
| q | r | s | t | u | v | w | x | y | z | | | | | | |

if (freq[i] == true)
c++;
}

if c == 26
yes | No

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String s = scn.nextLine();

    boolean[] freq = new boolean[26];

    for(int i = 0; i < s.length(); i++){
        char ch = s.charAt(i);
        if(ch != ' '){
            if(ch >= 'A' && ch <= 'Z'){
                int idx = ch - 'A';
                freq[idx] = true;
            }
            else if(ch >= 'a' && ch <= 'z'){
                int idx = ch - 'a';
                freq[idx] = true;
            }
        }
    }

    int c = 0;
    for(int i = 0; i < 26; i++){
        if(freq[i] == true){
            c++;
        }
    }
    if(c == 26){
        System.out.println("YES");
    }
    else{
        System.out.println("NO");
    }
}
```