Syso → Ln

```
for (int i=0 ; i<5 ; i++)
{
    Syso("shruti");
}
```

("Hi shruti");

```
for ( ① ; ② ; ④ )
{
    ③
}
```

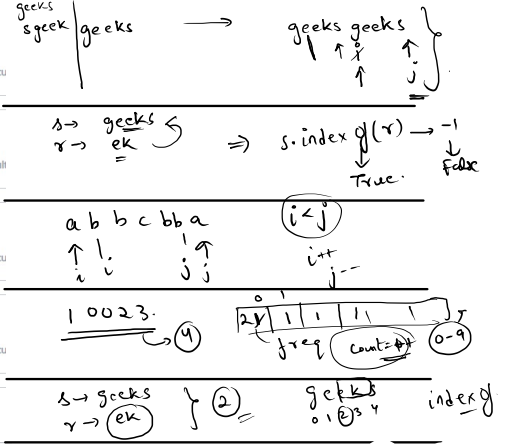
o/p
shruti
shruti

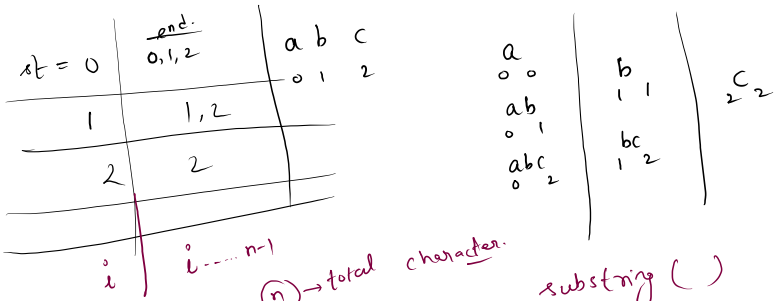
```
for (int i=0; i<5 ; i++)
{
    Syso ("shruti");
}
```

i=1	0<5.
	1<5.

Revision.

- Generate Rotation — Success Rate: 100.00% Max Score: 10 Difficu
- Target String — Success Rate: 97.62% Max Score: 10 Difficult
- Is Palindrome — Success Rate: 100.00% Max Score: 10 Difficu
- Find Unique — Success Rate: 100.00% Max Score: 10 Difficu
- Locate the Target String
- Print All Substrings





substring ( )

- 1 parameter
  - (1)
  - [1 ... till end]
- 2 parameter
  - (x, y)
  - [x, y)
  - [x, y-1]

Sum of All Substrings

Problem | Submissions | Leaderboard | Discussions

Take a **String str** as input and print the **sum** of all **substrings** of a string representing a **number**.

Eg: str="1234"

Sum = 1 + 2 + 3 + 4 + 12 + 23 + 34 + 123 + 234 + 1234 = **1670**

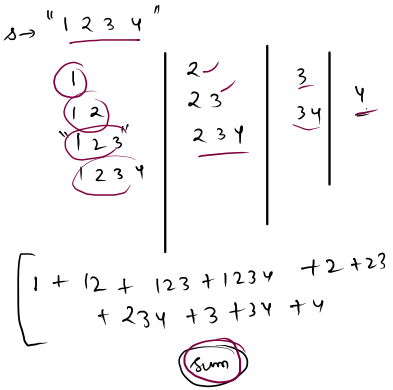
Note: Number will be in string fromat.

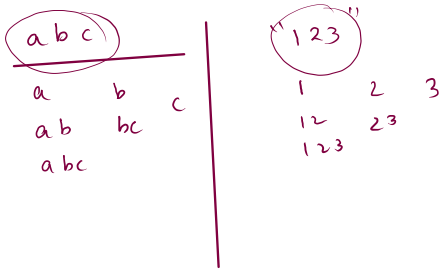
Sample Input 0

```
12345
```

Sample Output 0

```
16755
```



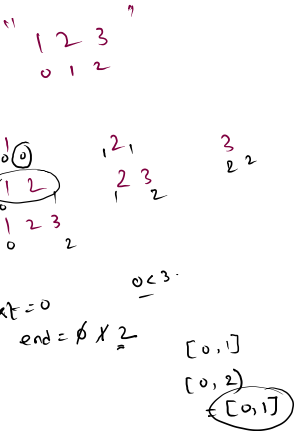



```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String s = scn.next();

    int n = s.length();
    int sum = 0;

    for(int st = 0; st < n; st++){
        for(int end = st; end < n; end++){
            String ss = s.substring(st, end+1);
            int val = Integer.parseInt(ss);
            sum += val;
        }
    }

    System.out.println(sum);
```



Desired String

Problem | Submissions | Leaderboard | Discussions

Take a string as input. Print the count of all the substrings that start with 'A' and end with 'A'. Also print the length of the longest such substring in the second line. In the third line, print that longest substring.

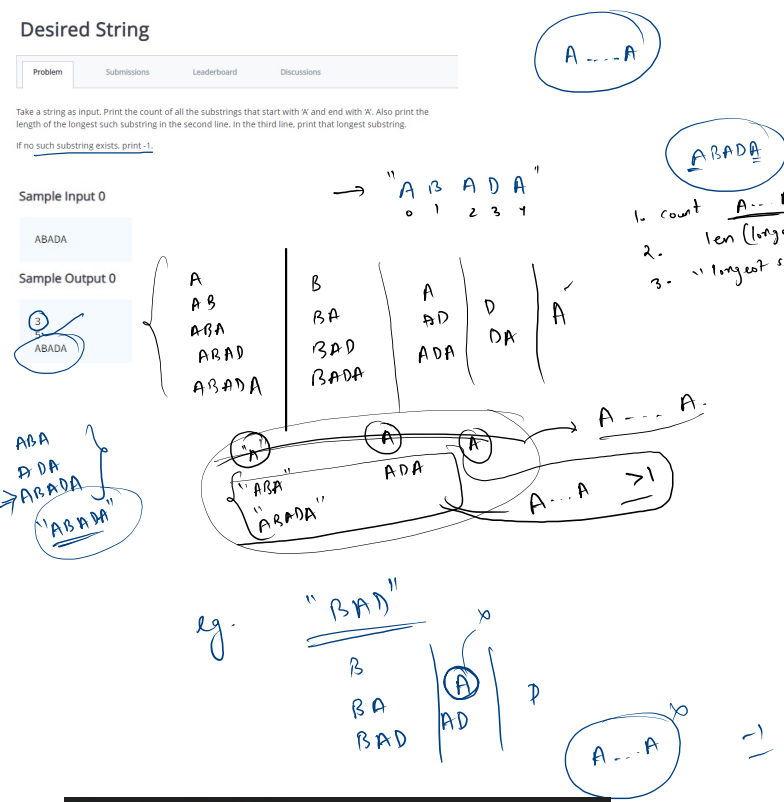
If no such substring exists, print -1.

Sample Input 0

```
ABADA
```

Sample Output 0

```
3
ABADA
```




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

    int n = s.length();
    int count = 0;      // how many ss are there of pattern A...A and also len > 1
    String maxString = "";      // it will store the ss of pattern A..A which is longest


    for(int st = 0; st < n; st++){
        for(int end = st; end < n; end++){
            String ss = s.substring(st, end+1);

            if(ss.length() > 1 && ss.charAt(0) == 'A' && ss.charAt(ss.length()-1) == 'A'){
                count++;
                if(ss.length() > maxString.length()){
                    maxString = ss;
                }
            }
        }
    }

    if(count > 0){
        System.out.println(count);
        System.out.println(maxString.length());
        System.out.println(maxString);
    }
    else{
        System.out.println(-1);
    }
```

Count Consecutive Ones

Problem | Submissions | Leaderboard | Discussions

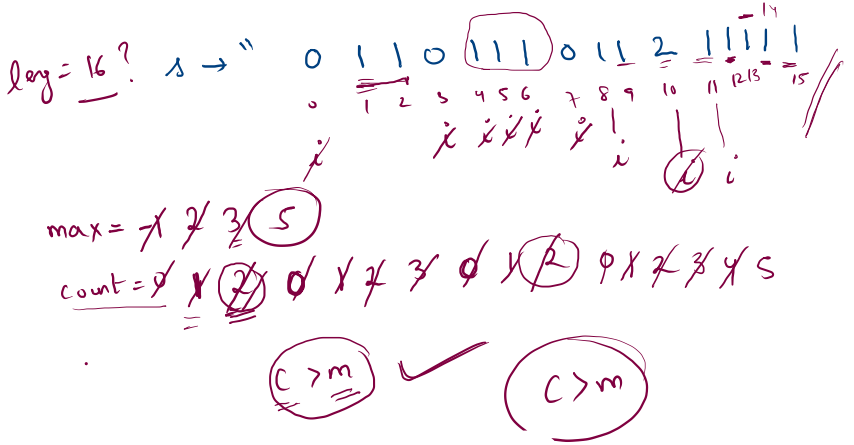
Given a string str as a string input. Count the maximum consecutive ones in the string.

Sample Input 0

```
0110111011211111
```

Sample Output 0

```
5
```




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

    int curr = 0;
    int max = -1;

    for(int i = 0; i < s.length(); i++){
        if(s.charAt(i) == '1'){
            curr++;
        }
        else{
            if(curr > max){
                max = curr;
            }
            curr = 0;
        }
    }
    if(curr > max){
        max = curr;
    }



    System.out.println(max);
```

Power of a String

Problem | Submissions | Leaderboard | Discussions

Take a **String str** as input and calculate the **Power** of the string.

Power of a string is defined as the **maximum length** of **substring** that contains only one **unique** character.

-> A **substring** is a continuous sequence of characters within a string.

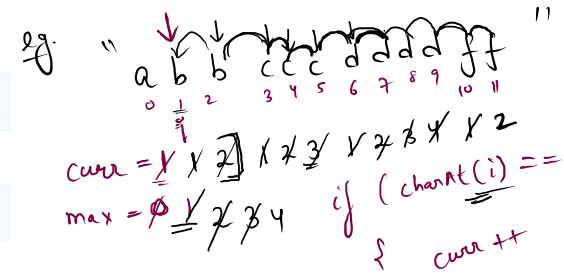
Note: All characters in the string are in **lowercase**.

Sample Input 0

```
abbcccddddeeeeeffgghheeccccc
```

Sample Output 0

```
5
```



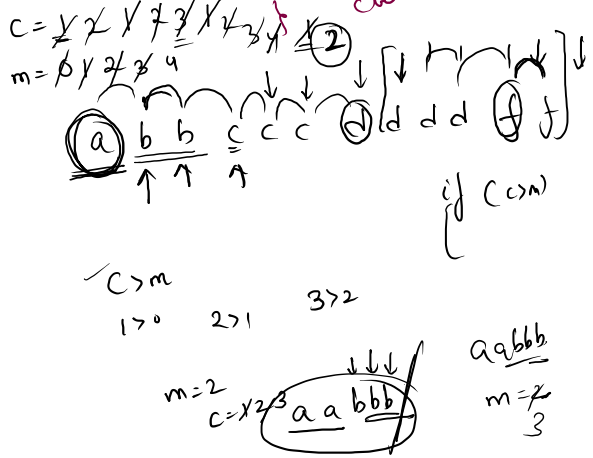

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String s = scn.next();

    int curr = 1;
    int max = 0;

    for(int i = 1; i < s.length(); i++){
        if(s.charAt(i) == s.charAt(i-1)){
            curr++;
        }
        else{
            if(curr > max){
                max = curr;
            }
            curr = 1;
        }
    }
    if(curr > max){
        max = curr;
    }



    System.out.println(max);
}
```



Merge Strings Alternatively

Problem | Submissions | Leaderboard | Discu

Take two strings as input.

Merge both the strings **alternatively**.

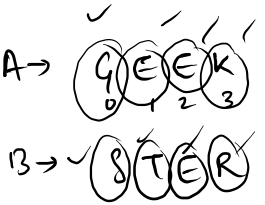
Note: Length of strings will be same.

Sample Input 0

```
GEEK
STER
```

Sample Output 0

```
GSETEEKR
```



" "

G S E T E E K R.

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
       Scanner scn = new Scanner(System.in);
        String A = scn.next();
        String B = scn.next();

        String C = "";

        for(int i = 0; i < A.length(); i++){
            C += A.charAt(i);
            C += B.charAt(i);
        }
        System.out.println(C);

    }
}
```

A- G E E K
0 1 2 3
B- S T E R

C = "GSETEEKR"

i = 0 1 2 3 4

0 < 4

4 < 4 ✗