Find Distance B/W Two Characters

str = " geeksforgeeks "

0 1 2 3 4

ch1 = 'g'

ch2 = 's'

psudo code

1) loop with i pointer start from 0

1.1) check if char at i is ch1

1.1.1) loop with j pointer start from (i+1)

1.1.1.1) check if char at j is ch2

ans = min (j - i - 1, ans);

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    char ch1 = scn.next().charAt(0);
    char ch2 = scn.next().charAt(0);
    System.out.println(findDistance(str, ch1, ch2));
}
public static int findDistance(String str, char ch1, char ch2) {
    int ans = Integer.MAX_VALUE;
    for (int i = 0; i < str.length(); i++)
        if ( str.charAt(i) == ch1 )
            for (int j = i + 1; j < str.length(); j++)
                if ( str.charAt(j) == ch2 )
                    ans = Math.min( ans, j - i - 1 );
    return ans;
}
```

⇒ Substring (subpart of a string)

Syntex

str = "AaBZ123bcd"

0 1 2 3 4 5 6 7 8 9

1

str.substring(3,7); // Z123

str.substring(1,4); // aBZ

str.substring(6); // 3bcd

Note:- T.C of substring O(N)

Str = "abcd"

0 1 2 3

all substring

	(i, j)
a	(0,1)
ab	(0,2)
abc	(0,3)
abcd	(0,4)
b	(1,2)
bc	(1,3)
bcd	(1,4)
c	(2,3)
cd	(2,4)
d	(3,4)

psudo code

1) loop i start at 0

1.1) loop j start at i+1

code

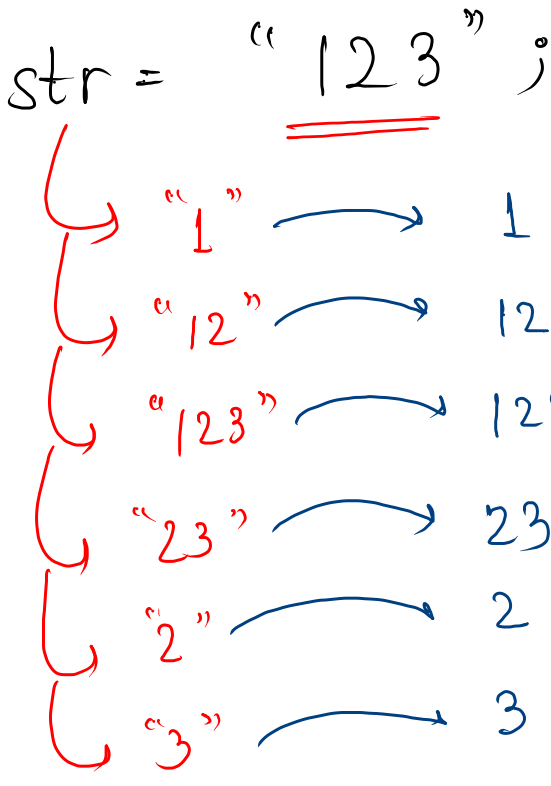
```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();

    for (int i = 0; i < str.length(); i++) {
        for (int j = i + 1; j <= str.length(); j++) {
            System.out.println( str.substring(i, j) );
        }
    }
}
```

↳ O(N)

$T.C = O(N^3)$

where N is len of str.

Sum of All Substrings



Note:-

1) Integer.parseInt(str)

2) Integer.valueOf(str)

Code

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

    int ans = 0;
    for (int i = 0; i < str.length(); i++) {
        for (int j = i + 1; j <= str.length(); j++) {
            String subs = str.substring(i, j);
            ans = ans + Integer.parseInt(subs);
        }
    }
    System.out.println(ans);
}
```

Desired String

str = "ABADA"

substrings

- "A" X
  - "AB"
    - "ABA"
      - "ABAD"
        - "ABADA"
- "B"
  - "BA"
    - "BAD"
      - "BADA"
- "A" X
  - "AD"
    - "ADA"
- "D"
  - "DA"
- "A" X

Ques 1) count substring with "A" at start and end

2) print longest such substring

3) print len of longest substring

code

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

    int count = 0;
    int len = 0;
    String maxSubstring = "";
    for (int i = 0; i < str.length(); i++) {
        for (int j = i + 1; j <= str.length(); j++) {
            String subs = str.substring(i, j);
            if ( subs.length() != 1 && subs.charAt(0) == 'A' && subs.charAt(subs.length() - 1) == 'A' ) {
                count++;
                if ( subs.length() > len ) {
                    maxSubstring = subs;
                    len = subs.length();
                }
            }
        }
    }
    if ( count == 0 ) {
        System.out.println(-1);
    } else {
        System.out.println(count);
        System.out.println(len);
        System.out.println(maxSubstring);
    }
}
```