

Find Distance B/W Two Characters

Ex1 str = "Greekstengreekster"

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

→ ch1 = s

→ ch2 = e

Ex2

str = "cbacacccb"

0 1 2 3 4 5 6 7 8

ch1 = a

ch2 = b

ch1 = a
ch2 = b

str = "cbacacccb"
0 1 2 3 4 5 6 7 8
↑ ↑ ↑ ↑ ↑ ↑ ↑
j i j j j j j

psudo
code

nested [\hookrightarrow iterate i pointer for ch1 (For())
 \hookrightarrow iterate for j for ch2
 \hookrightarrow min(diff)

code

```
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(findDifference(str, ch1, ch2));  
}  
  
public static int findDifference(String str, char ch1, char ch2) {  
    int n = str.length();  
    int ans = Integer.MAX_VALUE;  
    for (int i = 0; i < n; i++)  
        if (str.charAt(i) == ch1 )  
            for (int j = i; j < n; j++)  
                if (str.charAt(j) == ch2 )  
                    ans = Math.min( ans, j - i - 1 );  
    return ans;  
}
```

T.C = O(N)

two
pointer

⇒ Substring

↳ continuous subpart of element

syntax

↳ str.substring(start-idx, end-idx + 1);

↳ str.substring(start-idx);

str = abcd

↳ a ↳ b ↳ c ↳ d
↳ ab ↳ bc ↳ cd
↳ abc ↳ bcd
↳ abcd

Print All Substrings

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

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

syntax

$O(N)$

Sum of All Substrings

str = "1234"

substrings

↳ "1"

↳ "12"

↳ "123"

↳ "1234"

↳ "2"

↳ "23"

↳ "234"

↳ "3" ↳ "4"

↳ "34"

Inbuilt function :-

↳ Integer.parseInt(str);

↳ Integer.valueOf(str);

Code

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

    System.out.println(sumOfSubstring(str));
}

public static int sumOfSubstring(String str) {
    int n = str.length();
    int ans = 0;
    for (int i = 0; i < n; i++) {
        for (int j = i; j < n; j++) {
            String sub = str.substring(i, j + 1);
            ans += Integer.parseInt(sub);
        }
    }
    return ans;
}
```

Desired String

- ↳ count all substring which start and end 'A'
- ↳ length longest such substring
- ↳ longest such substring

j
 \downarrow
 str = "ABAD A"
 0 1 2 3 4
 \uparrow
 i

substring = "A"
 = "AB" X
 ✓ = "ABA"
 = "ABAD" X
 ✓ = "ABADA"

str.substring(i, j+1)

1) count = 0 \neq 3

2) len = 0 \neq 5

3) ans = ~~"ABA"~~

= "ABADA"
 = "B" = "A"
 = "BA" = "AD"
 = "BAD" = "ADA" ✓
 = "BADA" = "D" = "A"
 = "DA"

code

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

public static void desiredString(String str) {
    int count = 0;
    String longestString = "";
    int len = 0;
    for (int i = 0; i < str.length(); i++) {
        for (int j = i + 1; j < str.length(); j++) {
            String sub = str.substring(i, j + 1);
            if ( sub.charAt(0) == 'A' && sub.charAt(sub.length() - 1) == 'A' ) {
                count++;
                if ( len < sub.length() ) {
                    len = sub.length();
                    longestString = sub;
                }
            }
        }
    }
    if (count != 0) {
        System.out.println(count);
        System.out.println(len);
        System.out.println(longestString);
    } else {
        System.out.println("-1");
    }
}
```

