Is Palindrome

(Imp)

Easy

- "abcba"
- "atta"
- "naman"
- "madam"

str = "radar";

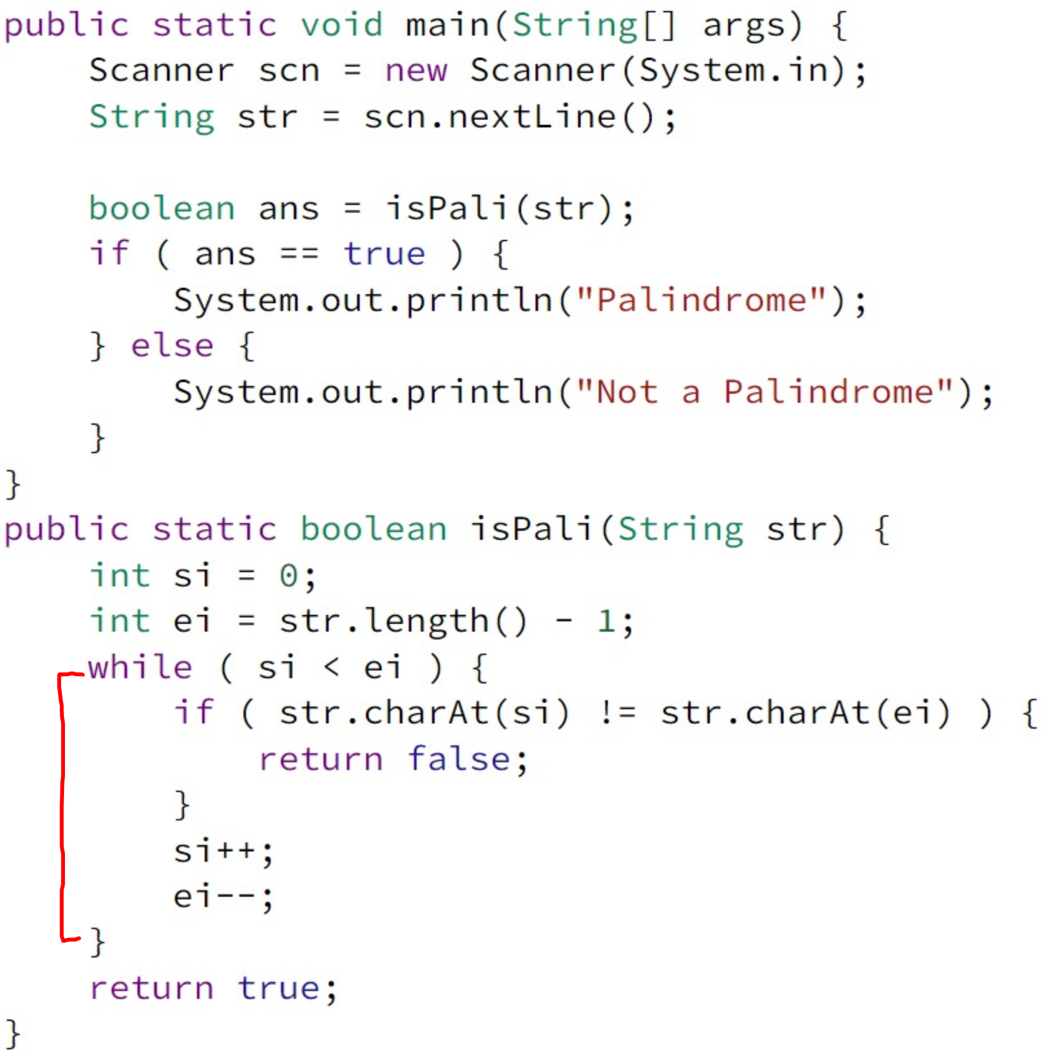
⇈

si ei

```
1) loop until si < ei.
    1.1) char at si != char at ei
            return false
    1.2) si++ , ei--
2) return true
```

code

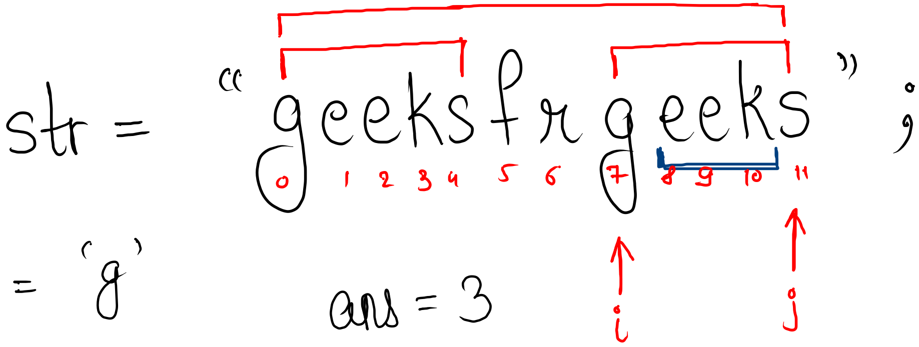
T.C = O(str.length())



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

    boolean ans = isPali(str);
    if ( ans == true ) {
        System.out.println("Palindrome");
    } else {
        System.out.println("Not a Palindrome");
    }
}
public static boolean isPali(String str) {
    int si = 0;
    int ei = str.length() - 1;
    while ( si < ei ) {
        if ( str.charAt(si) != str.charAt(ei) ) {
            return false;
        }
        si++;
        ei--;
    }
    return true;
}
```

Find Distance B/W Two Characters



str = "geeksfrgeeks";

0 1 2 3 4 5 6 7 8 9 10 11

i

j

ch1 = 'g'

ch2 = 's'

ans = 3

Note :-

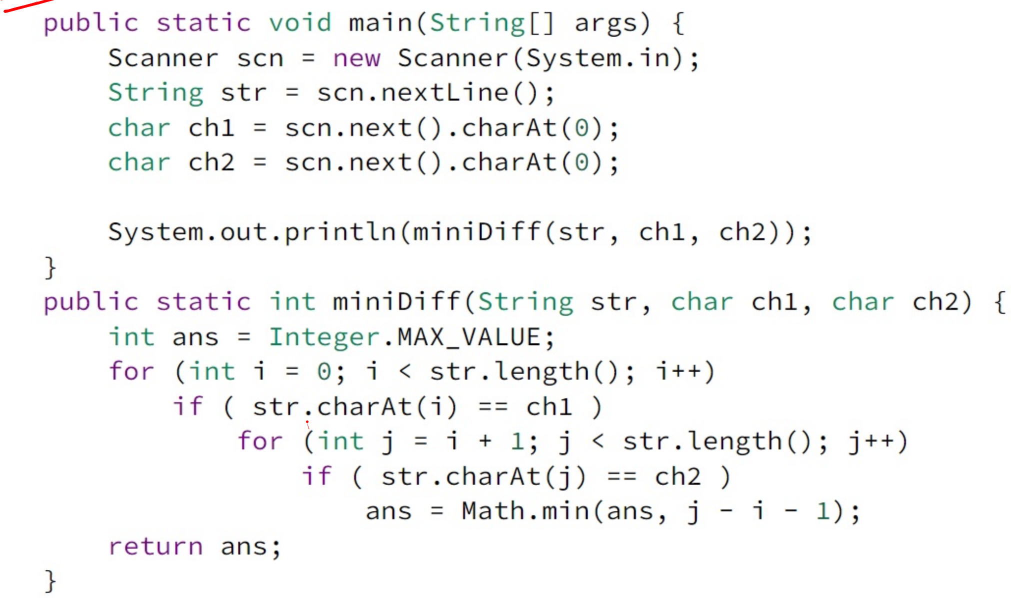
diff b/w 2 pointers excluding both is

$j-i-1$

code

```
ans = ∞
1) loop from 0 to n-1
    1.1) check if char at i == ch1
        1.1.1) loop from (i+1) to (n-1)
            1.1.1.1) check if char at j == ch2
                ans = Math.min(ans, j-i-1);
```

code

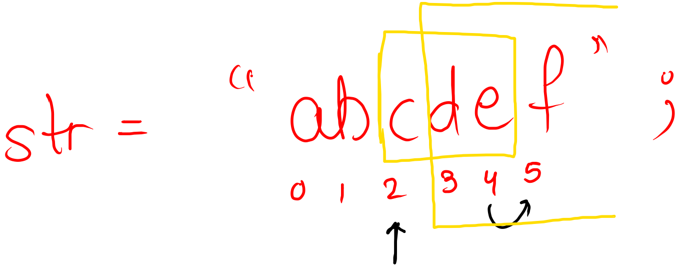


```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    char ch1 = scn.next().charAt(0);
    char ch2 = scn.next().charAt(0);

    System.out.println(miniDiff(str, ch1, ch2));
}
public static int miniDiff(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;
}
```

T.C = O(N), N = str.length()

⇒ Substring



```
str = "abcdef";
       0 1 2 3 4 5
           ↑
```

```
str.substring(2,5);
str.substring(3,6); ✓
```

Ques Generate all possible substrings

str = "abcd";

0 1 2 3

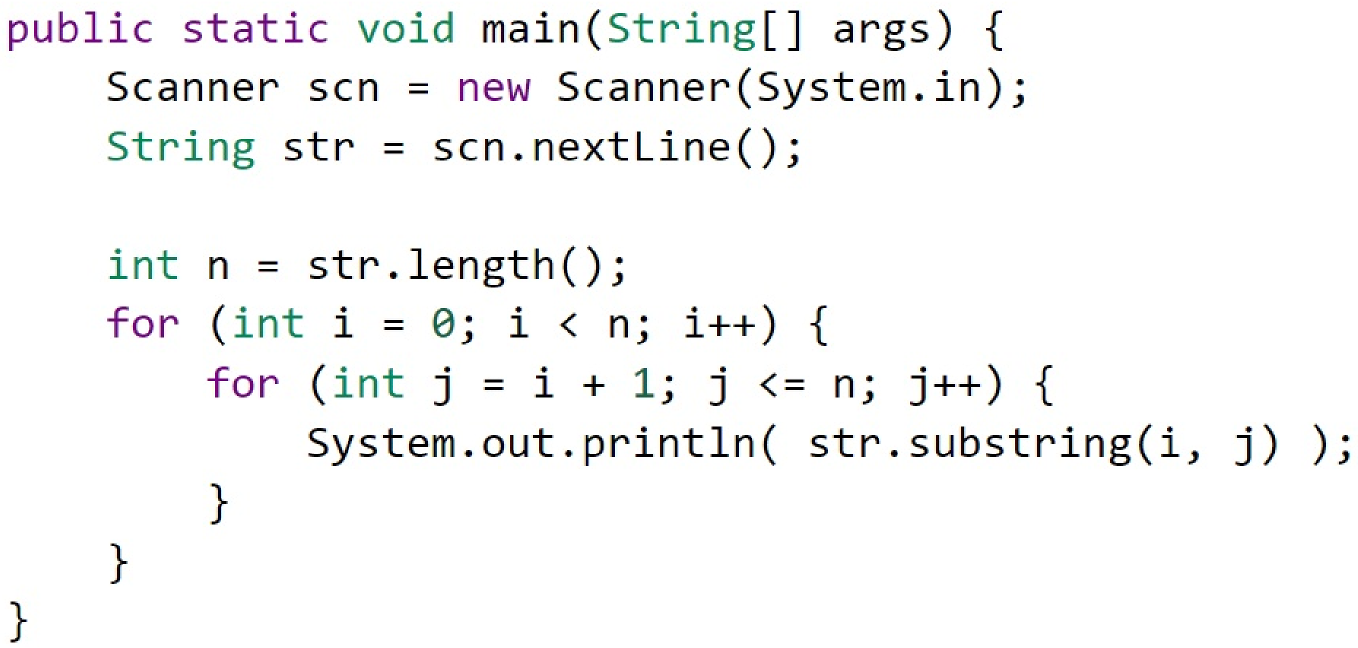
substrings

	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)

code

```
loop i = 0 → (n-1)
    loop j = i+1 → n
        print str.substring(i, j)
```

Code



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

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

Sum of All Substrings

↳ generate all substring and add those as a no.

str = "1234";

- ✓ ↳ "1"
- ✓ ↳ "12"
- ✓ ↳ "123"
- ✓ ↳ "1234"
- ✓ ↳ "2"
- ✓ ↳ "23"
- ✓ ↳ "234"
- ✓ ↳ "3"
- ✓ ↳ "34"
- ✓ ↳ "4"

ans = 0+1+12+123+1234
+2+23+234+3+34+4

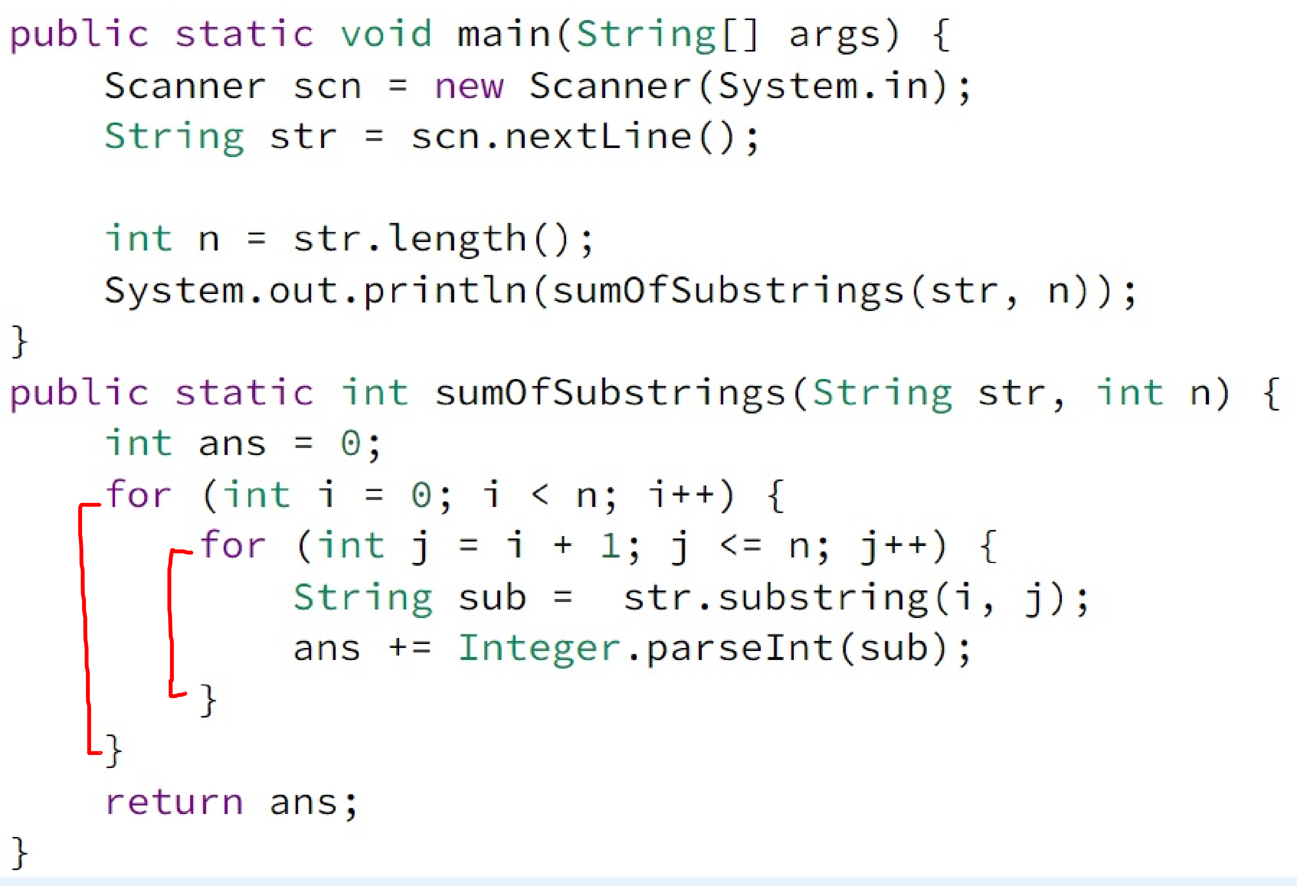
Note:-

1) Integer.parseInt(str);

2) Integer.valueOf(str);

Code

T.C = $N^2 * N = O(N^3)$



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

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

Desired String

str = "ABADA";

Substring

X ↳ A
↳ AB
✓ ↳ ABA
↳ ABAD
✓ ↳ ABADA

↳ B
↳ BA
↳ BAD
↳ BADA

X ↳ A
↳ AD
✓ ↳ ADA

↳ D
↳ DA

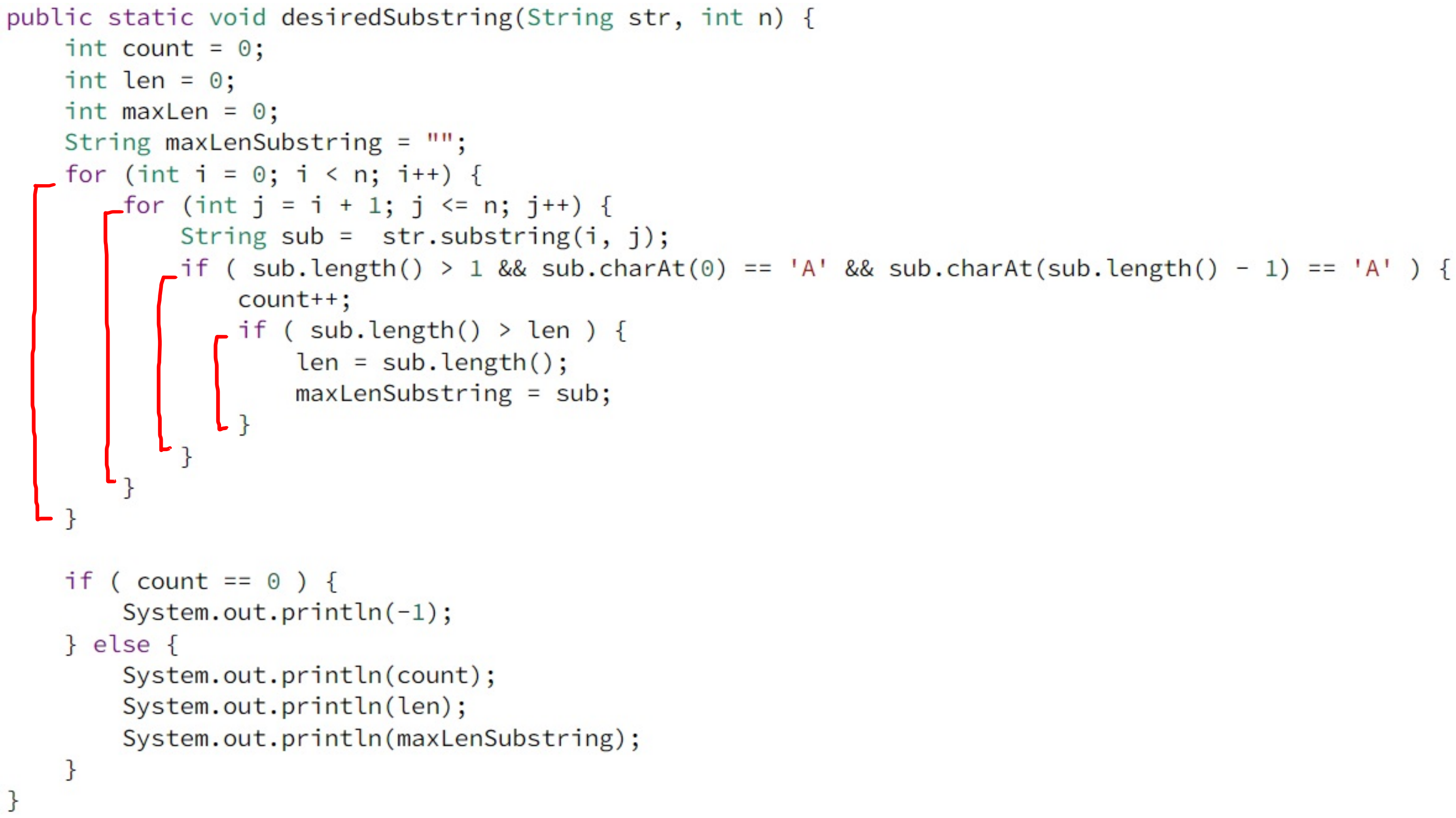
X ↳ A

ans = 3
5
ABADA

observation

1) Count of substring start ane end with 'A'
2) length of longest such sub.
3) longest sub.

code



```java
public static void desiredSubstring(String str, int n) {
    int count = 0;
    int len = 0;
    int maxLen = 0;
    String maxLenSubstring = "";
    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j <= n; j++) {
            String sub =  str.substring(i, j);
            if ( sub.length() > 1 && sub.charAt(0) == 'A' && sub.charAt(sub.length() - 1) == 'A' ) {
                count++;
                if ( sub.length() > len ) {
                    len = sub.length();
                    maxLenSubstring = sub;
                }
            }
        }
    }

    if ( count == 0 ) {
        System.out.println(-1);
    } else {
        System.out.println(count);
        System.out.println(len);
        System.out.println(maxLenSubstring);
    }
}
```