Count Words

str = "Welcome_to_geeksters"

str = "This_is_a___sentence"

c = 0

no. of words

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    int count = 0;
    for (int i = 0; i < str.length() - 1; i++) {
        char ch = str.charAt(i);
        char ch1 = str.charAt(i + 1);
        if ( ch == ' ' && ch1 != ' ' ) {
            count++;
        }
    }
    count++;
    System.out.println(count);
}
```

Approch :- only when there are no spaces in start or end of sent.

str = "This is a sentence"

0 1 2 3 4 5 6 7 8 9 10 11 12 ...

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    int count = 0;
    for (int i = 0; i < str.length() - 1; i++) {
        char ch = str.charAt(i);
        char ch1 = str.charAt(i + 1);
        if ( ch == ' ' && ch1 != ' ' ) {
            count++;
        }
    }
    count++;
    System.out.println(count);
}
```

c = ~~0 1 2~~ 3

	ch	ch1
i = 0	T,	h
i = 1,	h,	i
i = 2,	i,	s
i = 3,	s,	' '
i = 4,	' ',	i
i = 5,	i,	s
i = 6,	s,	' '
i = 7,	' ',	a
i = 8,	a,	' '
i = 9,	' ',	' '
i = 10,	' ',	' '
i = 11	' ',	s
i = 12		

Inbuilt function

str = "Welcome_to_geeksster"

1) str.split(' ');

create one array divided by space

our | Welcome | to | geeksster |

2) str.split('e'); ⟶ | W | lcom | _to_g | '' | kst | r |

Find Unique

str = "100234"
0 1 2 3 4 5

arr

T	T	T	T	T	F	F	F	F	F
0	1	2	3	4	5	6	7	8	9

(boolean)

'0' → 27
'1' → 28
'2' → 29
'3' → 30
.
.
.
'9' → 37

i = 0, ch = '1'
int index = '1' - '0' = 1
28 - 27

i = 1, ch = '0'
index = '0' - '0' = 0

i = 2, ch = '0'
index = '0' - '0' = 0

i = 3, ch = '2'
index = '2' - '0' = 2

i = 4, ch = '3'
index = '3' - '0' = 3

i = 5, ch = '4'
index = '4' - '0' = 4

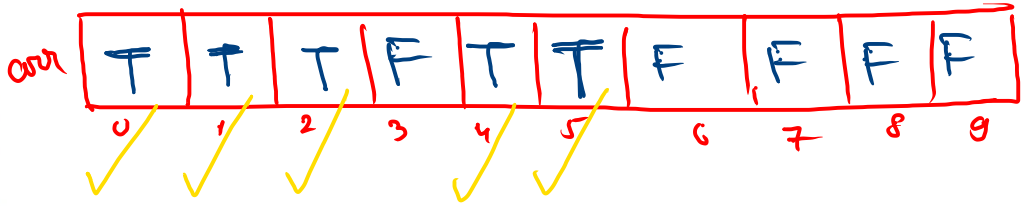
```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();
    System.out.println(countUnique(str));
}

public static int countUnique(String str) {
    boolean[] arr = new boolean[10];
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        int idx = ch - '0';
        arr[idx] = true;
    }

    int count = 0;
    for (int i = 0; i < 10; i++) {
        if (arr[i] == true) {
            count++;
        }
    }
    return (count);
}
```

str = "5401152";

0 1 2 3 4 5 6



i=0, ch='5'
idx='5'-'0'=5

i=1, ch='4'
idx='4'-'0'=4

i=2, ch='0'
idx='0'-'0'=0

i=3, ch='1'
idx='1'-'0'=1

i=4, ch='1'
idx='1'-'0'=1

i=5, ch='5'
idx='5'-'0'=5

i=6, ch='2'
idx='2'-'0'=2

5

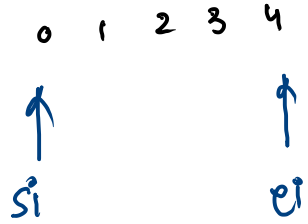
Is Palindrome

↳ 2 pointers

si = starting index

ei = ending index

String str = "radar";



si = 0

ei = str.length() - 1

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

public static String isPalindrome(String str) {
    int si = 0;
    int ei = str.length() - 1;
    while (si <= ei) {
        if ( str.charAt(si) != str.charAt(ei) ) {
            return "Not a Palindrome";
        }
        si++;
        ei--;
    }
    return "Palindrome";
}
```