

Given an array of strings words, return the first palindromic string in the array. If there is no such string, return an string "empty".

A string is palindromic if it reads the same forward and backward.

#### Input Format

- first line is the length of array of string
- array of string

#### Constraints

- $1 \leq \text{words.length} \leq 100$
- $1 \leq \text{words}[i].\text{length} \leq 100$
- $\text{words}[i]$  consists only of lowercase English letters.

#### Output Format

Return the palindromic string from the array, print "empty" if string is not possible.

#### Sample Input 0

```
5  
abc car ada racecar cool
```

#### Sample Output 0

```
ada
```

#### Explanation 0

The first string that is palindromic is "ada".  
Note that "racecar" is also palindromic, but it is not the first.

abc, car, ada, racecar, cool  
abc, car, ada, racecar, cool  
abc, car, ada, racecar, cool  
abc, car, ada, racecar, cool  
abc, car, ada, racecar, cool

2 - - - 3  
↓

function Palindrom  
i=0, j=str.length-1  
while(i<j) {  
s.charAt(i) != s.charAt(j)  
return false;  
}  
i++;  
j--;  
}  
return true

```
1 import java.io.*;  
2 import java.util.*;  
3  
4 public class Solution {  
5  
6     public static void main(String[] args) {  
7         /* Enter your code here. Read input from STDIN. Print output to S  
8         Scanner sc = new Scanner(System.in);  
9         int n = sc.nextInt();  
10        //sc.nextLine();  
11        String[] str = new String[n];  
12        for(int i = 0; i < n; i++){  
13            str[i] = sc.next();  
14        }  
15        for(int i = 0; i < n; i++){  
16            if(isPalindrom(str[i])){  
17                System.out.println(str[i]);  
18                return;  
19            }  
20        }  
21        System.out.println("empty");  
22    }  
23  
24    public static boolean isPalindrom(String str1){  
25        int i = 0, j = str1.length()-1;  
26        boolean bl = true;  
27        while(i < j){  
28            if(str1.charAt(i) != str1.charAt(j)){  
29                bl = false;  
30                break;  
31            }  
32            i++;j--;  
33        }  
34        return bl;  
35    }  
36 }
```

A palindrome is a term that can be read the same way forwards and backwards, irrespective of any punctuation or capitalization changes. To determine whether a phrase is a palindrome, the phrase is first converted to lowercase, and all non-alphanumeric characters are removed, leaving only letters and numbers.

#### Input Format

string str as an input.

#### Constraints

$1 \leq \text{str.length} \leq 2 * 10^5$

str consists only of printable ASCII characters.

#### Output Format

return true or false.

#### Sample Input 0

A man, a plan, a canal: Panama

#### Sample Output 0

true

## Submitted Code

Language: Java 15

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         String str = sc.nextLine();
9
10        int i = 0;
11        int j = str.length() - 1;
12        while(i < j){
13            char ilo = Character.toLowerCase(str.charAt(i));
14            char jlo = Character.toLowerCase(str.charAt(j));
15
16            if((ilo < 'a' || ilo > 'z') && (ilo < '0' || ilo > '9')){
17                i++;
18            }else if((jlo < 'a' || jlo > 'z') && (jlo < '0' || jlo > '9')){
19                j--;
20            }else if(ilo != jlo){
21                System.out.print(false);
22                return;
23            }else{
24                i++;
25                j--;
26            }
27        }
28
29        System.out.print(true);
30    }
31 }
32 }
```

Print the count of numbers in a given string.

Eg. 132ab2cd45ef6. The above string has 4 numbers.

#### Input Format

A String

#### Constraints

1<=str.length()<=100000

#### Output Format

An integer value

#### Sample Input 0

132ab2cd45ef6

#### Sample Output 0

4

#### Explanation 0

132, 2, 45 and 6 are number present in string

Language: Java 15

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         String str = sc.nextLine();
9         int count=0;
10        boolean already= false;
11        for(int i=0;i<str.length();i++){
12            char ch= str.charAt(i);
13            if(Character.isDigit(ch)){
14                if(!already){
15                    count++;
16                    already= true;
17                }
18            }else{
19                already=false;
20            }
21        }
22        System.out.print(count);
23    }
24 }
```

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         String str = sc.nextLine();
9         int count=0;
10        boolean already= false;
11        for(int i=0;i<str.length();i++){
12            char ch= str.charAt(i);
13            if(ch>='0' && ch<='9'){
14                if(!already){
15                    count++;
16                    already= true;
17                }
18            }else{
19                already=false;
20            }
21        }
22        System.out.print(count);
23    }
24 }
```

Handwritten annotations and diagrams:

- Red handwritten text: "without inbuilt"
- Red handwritten string: "132ab2cd45ef6" with numbers 1, 2, 3, 4 above the digits 1, 3, 2, 6 respectively.
- Blue handwritten diagram: A circle with '1' inside, crossed out with an 'X', and a '3' next to it.
- Blue handwritten diagram: A circle with '1' inside, with arrows pointing to '2' and '3'.
- Blue handwritten diagram: A circle with '1' inside, with an arrow pointing to '3'.