```
-10^4 <= nums[i] <= 10^4

nums is sorted in non-decreasing order
```

Output Format

Return the desired answer.

Sample Input 0

```
6
1 1 1 2 2 3
```

Sample Output 0

```
5
```

Explanation 0

Your function should return k = 5, with the first five elements of nums being 1, 1, 2, 2 and 3 respectively. It does not matter what you leave beyond the returned k (hence they are underscores).

Sample Input 1

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
         Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int arr[] = new int[n];
        for(int i=0;i<n;i++)arr[i]=sc.nextInt();

        int ans =2;
        for(int i =2;i<n;i++){
            if(arr[i]!=arr[ans-2]){
                arr[ans]=arr[i];
                ans++;
            }
        }
        System.out.print(ans);
    }
}
```

```java
public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int[]arr = new int[n];
        for(int i = 0; i < n; i++){
            arr[i] = scn.nextInt();
        }
       int ans = duplication(n, arr);
        System.out.println(ans);
    }
    public static int duplication(int n, int[]arr){
        int finalCount = 0,tempVal=Integer.MIN_VALUE;
        for(int i = 0; i < n; i++){
            int count = 0;
            if(tempVal==arr[i])continue;
            for(int j = 0; j < n; j++){
                if(arr[i] == arr[j]){
                    tempVal=arr[i];
                    count++;
                }
            }
            if(count < 3){
                finalCount += count;
            }else{
                finalCount += 2;
            }
        }
        return finalCount;
    }
}
```

```java
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to S
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    //sc.nextLine();
    String[] str = new String[n];
    for(int i = 0; i < n; i++){
        str[i] = sc.next();
    }
    for(int i = 0; i < n; i++){
        if(isPalindrom(str[i])){
            System.out.println(str[i]);
            return;
        }
    }
    System.out.println("empty");
}

public static boolean isPalindrom(String str1){
    int i = 0, j = str1.length()-1;
    boolean bl = true;
    while(i < j){
        if(str1.charAt(i) != str1.charAt(j)){
            bl = false;
            break;
        }
        i++;j--;
    }
    return bl;
}
}
```

Palindrom

arr[i]

Sagar

-> ragas

DaD

Dad

jacar

racar

Given a string str, return the longest palindromic substring in str.

NOTE :-

After answering the question, attempt the related question in the linked resource to improve your understanding of the question . Clickhere

Input Format

string str as an input.

Constraints

1 <= str.length <= 10^2

Output Format

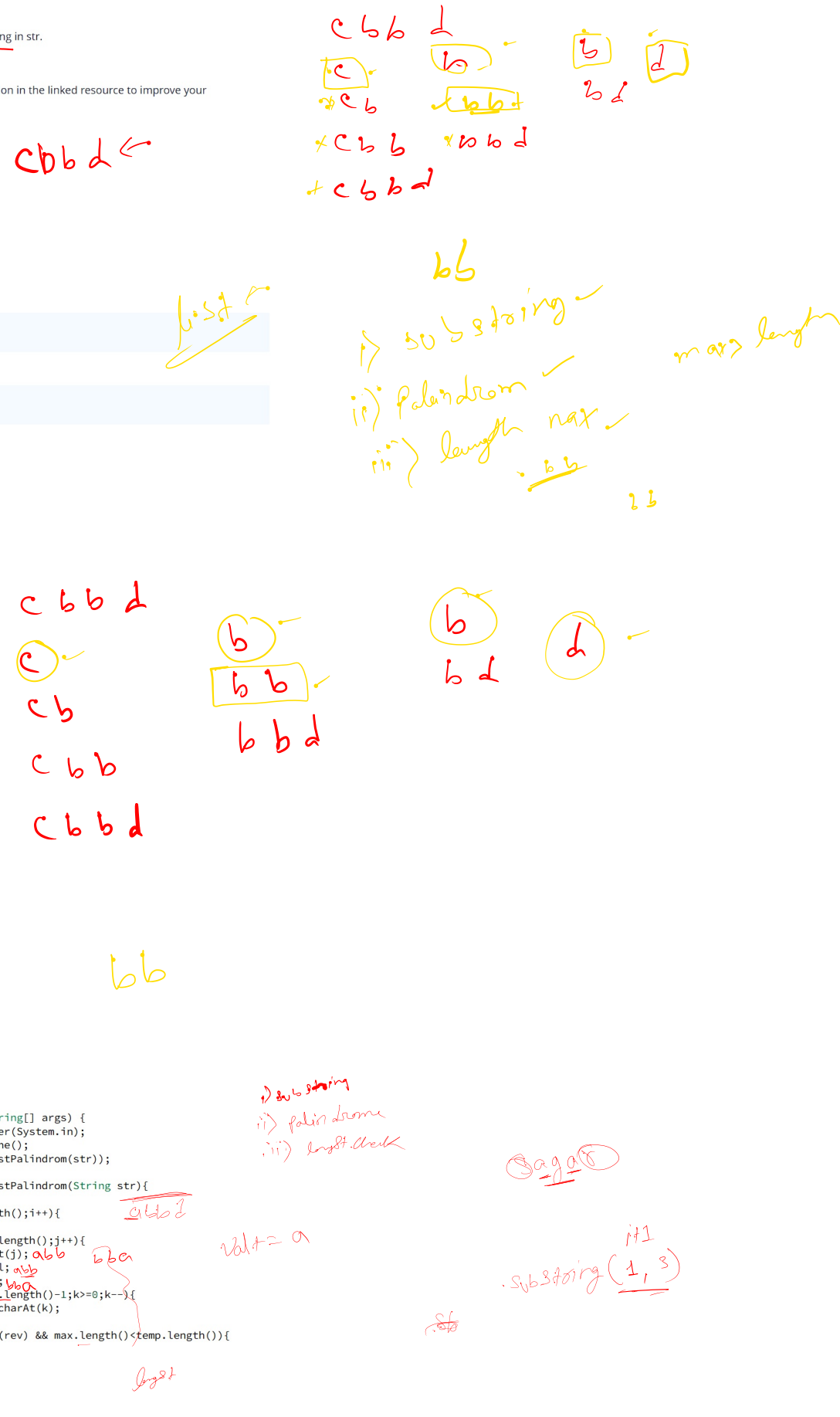
return the longest palindromic substring.

Sample Input 0

```
cbbd
```

Sample Output 0

```
bb
```



```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        System.out.print(longestPalindrom(str));
    }
    public static String longestPalindrom(String str){
        String max = "";
        for(int i=0;i<str.length();i++){
            String val ="";
            for(int j=i;j<str.length();j++){
                val+=str.charAt(j);
                String temp=val;
                String rev ="";
                for(int k=temp.length()-1;k>=0;k--){
                    rev+=temp.charAt(k);
                }
                if(temp.equals(rev) && max.length()<temp.length()){
                    max=temp;
                }
            }
        }
        return max;
    }
}
```