/\*\*

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

\*/

package Problem20\_longest\_Palindromic\_Substring;

import com.sun.jdi.IntegerValue;

import java.util.\*;

import javax.print.DocFlavor;

import java.lang.String;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.\*;

import java.math.BigDecimal;

import java.math.BigInteger;

import java.nio.Buffer;

import java.util.Scanner;

import java.util.Vector;

public class Problem20\_longest\_Palindromic\_Substring{

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

String word = input.nextLine();

int len = word.length();

int start = 0, scope = 0;

for(int i=0; i<len; i++){

char element = word.charAt(i);

int index = word.indexOf(element,i+1);

if( index != -1){

StringBuilder temp = new StringBuilder();

temp.append(word.substring(i,index+1));

String str1,str2;

str1 = temp.toString();

str2 = temp.reverse().toString();

if(str1.equals(str2)){

int temp\_scope = index-i+1;

if(temp\_scope > scope){

scope = temp\_scope;

start = i;

}

}

}

}

System.out.println(word.substring(start,start + scope));

}

}