import java.util.Scanner;  
import java.math.BigDecimal;  
import java.util.ArrayList;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner input = new Scanner(System.*in*);  
  
  
 System.*out*.println("Please input some strings");  
 String h = input.nextLine();  
  
 Pattern pattern = Pattern.*compile*("\\d{3}");  
 Matcher matcher = pattern.matcher(h);  
  
 while(matcher.find()) {  
 System.*out*.println(matcher.group());  
 }  
  
  
 String ff = input.nextLine();  
  
 if (ff.substring(0, 1).equals("0")) {  
 ff.replace("0","1");  
 System.*out*.println(ff);  
 }  
 else if (ff.substring(0, 1).equals("1")) {  
 ff.replace("1","2");  
 System.*out*.println(ff+1);  
 }  
 else if (ff.substring(0, 1).equals("2")) {  
 ff.replace("2","3");  
 System.*out*.println(ff+2);  
 }  
 else if (ff.substring(0, 1).equals("3")) {  
 ff.replace("3","4");  
 System.*out*.println(ff+3);  
 }  
 else if (ff.substring(0, 1).equals("4")) {  
 ff.replace("4","5");  
 System.*out*.println(ff+4);  
 }  
 else if (ff.substring(0, 1).equals("5")) {  
 ff.replace("5","6");  
 System.*out*.println(ff+5);  
 }  
 else if (ff.substring(0, 1).equals("6")) {  
 ff.replace("6","7");  
 System.*out*.println(ff+6);  
 }  
 else if (ff.substring(0, 1).equals("7")) {  
 ff.replace("7","8");  
 System.*out*.println(ff+7);  
 }  
 else if (ff.substring(0, 1).equals("8")) {  
 ff.replace("8","9");  
 System.*out*.println(ff+8);  
 }  
 else if (ff.substring(0, 1).equals("9")) {  
 ff.replace("9","10");  
 System.*out*.println(ff+9);  
 }  
  
  
 //()表示组如(d(a)(c)且(表示第几组中间的(a)就是第二组因为是第二个(  
 //需求1: 判断一个字符串开始字符和结束字符  
 // \\1： 把首字母拿出来再次使用  
 String regex1 = "(.).+\\1";  
 System.*out*.println("m123m".matches(regex1));  
 System.*out*.println("b456b".matches(regex1));  
 System.*out*.println("c123c".matches(regex1));  
 System.*out*.println("w123w".matches(regex1));  
 System.*out*.println("u123p".matches(regex1));  
  
 //需求2: 判断一个字符串开始部分和结束部分是否一致？可以有多个字符  
 String regex2 = "(.+).+\\1";  
 System.*out*.println("w123w1".matches(regex1));  
 System.*out*.println("b2456b2".matches(regex1));  
 System.*out*.println("c123c".matches(regex1));  
 System.*out*.println("wg123wg".matches(regex1));  
 System.*out*.println("u123p".matches(regex1));  
  
 //需求3: 判断一个字符串的开始部分和结束部分是否一致? 开始部分内的字符也需要一致  
 //序号需要变成2了因为(.)现在是第二个括号  
 String regex3 = "((.)\\2\*).+\\1";  
 System.*out*.println("ww123ww".matches(regex1));  
 System.*out*.println("b2456b2".matches(regex1));  
 System.*out*.println("c123c".matches(regex1));  
 System.*out*.println("wg123wg".matches(regex1));  
 System.*out*.println("u123p".matches(regex1));  
  
 //将相同的字母变成一个  
 //+至少一次  
 String str = "ppppggggrrr";  
 String ow = str.replaceAll("(.)\\1+","$1");  
 System.*out*.println(ow);  
  
 //输入数字  
 int d = input.nextInt();  
 //转成String类型  
 String w = String.*valueOf*(d);  
 //找到w的长度  
 int o = w.length();  
 for (int i = 0; i < o; i++) {  
 System.*out*.print(w.charAt(i) + " ");  
 }  
 System.*out*.println();  
 //double t = input.nextDouble();  
 BigDecimal d1 = new BigDecimal("5.88");  
 System.*out*.println(d1);  
  
 ArrayList<Integer> list = new ArrayList<>();  
 int[][] f = new int[3][3];  
 for (int i = 0; i < f.length; i++) {  
 for (int j = 0; j < f[i].length; j++) {  
 f[i][j] = (i + 1) + (j + 1);  
 list.add(f[i][j]);  
 }  
 }  
  
 *Switch*(5);  
  
 }  
  
  
//斐波纳契数列  
 public static int t(int m) {  
 if (m == 1 || m == 2) {  
 return 1;  
 }  
 return *t*(m - 1) + *t*(m - 2);  
 }  
  
 public static int Switch(int m) {  
 return switch (m) {  
 case 1 -> *t*(m);  
 case 2 -> *t*(m);  
 case 3 -> *t*(m);  
 case 4 -> *t*(m);  
 case 5 -> *t*(m);  
 default -> *t*(m);  
 };  
}  
  
 public static boolean a(int[][] o) {  
  
 int cnt = 0;  
 for (int i = 0; i < o.length; i++) {  
 for (int j = 0; j < o[i].length; j++) {  
 if (o[i][j] < 10) {  
 cnt++;  
 } else {  
 cnt = 0;  
 }  
 }  
 }  
 if (cnt > 0) {  
 return true;  
 }  
 return false;  
 }  
  
 //input 9 digits' number  
 public static boolean b(int p) {  
 boolean t = true;  
 boolean g = true;  
 String o = String.*valueOf*(p);  
 for (int i = 0; i < o.length(); i++) {  
 String u = String.*valueOf*(o.charAt(i));  
 boolean m = u.matches("[1-8]");  
 if (!m) {  
 g = false;  
 break;  
  
 }  
 }  
  
 return g;  
 }  
  
  
 public static int o(int m) {  
 String h = String.*valueOf*(m); // 将数字转换为字符串  
 int q = h.length(); // 获取数字的位数  
 int c = 0;  
  
 // 遍历数字的每一位  
 for (int i = 0; i < q; i++) {  
 // 将字符转换为整数并计算幂次  
 int digit = Character.*getNumericValue*(h.charAt(i));  
 c += (int) Math.*pow*(digit, q); // 计算每一位的 q 次方和  
 }  
  
 // 检查幂次和是否等于原数  
 if (c == m) {  
 return 1;   
 }  
 return 0; // 不是阿姆斯特朗数，返回 0  
 }  
  
}