第二题：  
public class 括号深度 {  
public static void main(String[] args){  
Scanner scanner = new Scanner(System.in);  
String string = scanner.nextLine();  
int count = 0;  
while (string.contains("()") || string.contains("[]") || string.contains("{}")){  
string = string.replace("()", " ");  
string = string.replace("[]", " ");  
string = string.replace("{}", " ");  
string = string.replace(" ", "");  
count++;  
}  
if (string.equals("")){  
System.out.println(count);  
}else {  
System.out.println(0);  
}  
}  
}

第三题：  
public class 趣味游戏 {  
public static void main(String [] args){  
Scanner scanner = new Scanner(System.in);  
String string = scanner.nextLine();  
String[] strings = string.split(",");  
Set<String> set = new HashSet<>();  
//如果输入的数字不在范围内或者有重复，则输出-1。  
for (int i = 0; i < strings.length; i++){  
if (isNum(strings[i])){  
int num = Integer.valueOf(strings[i]);  
if (num <= 9 && num >= 1 ){  
set.add(strings[i]);  
}  
}  
}  
if (set.size() < strings.length){  
System.out.println(-1);  
return;  
}  
// 1）2可以当做5来使用，5也可以当做2来使用进行数字拼接，且屏幕不能同时给出2和5；  
// 2）6可以当做9来使用，9也可以当做6来使用进行数字拼接，且屏幕不能同时给出6和9。  
if (set.contains("6")){  
if (set.contains("9")){  
System.out.println(-1);  
return;  
}else {  
set.add("9");  
}  
}  
if (set.contains("9")){  
if (set.contains("6")){  
System.out.println(-1);  
return;  
}else {  
set.add("6");  
}  
}  
if (set.contains("2")){  
if (set.contains("5")){  
System.out.println(-1);  
return;  
}else {  
set.add("5");  
}  
}  
if (set.contains("5")){  
if (set.contains("2")){  
System.out.println(-1);  
return;  
}else {  
set.add("5");  
}  
}  
List<String> list = new ArrayList<>(set);  
Collections.sort(list);  
int n = list.size();  
//addList为拼接2位数 3位数 ...的集合 // 1 4 7 14 14 17 41 47  
// 1 4 7 //14 17 41 47  
int max = Integer.valueOf(list.get(n - 1));  
List<String> nums = new ArrayList<>(list);  
List<String> addList = new ArrayList<>(list); //9  
while (max > list.size() && addList.size() > 0){  
addList = getAddList(nums, addList);  
list.addAll(addList);  
}  
if (max <= list.size()){  
System.out.println(list.get(max - 1));  
}else {  
System.out.println(-1);  
}  
  
}  
// 14 14 17 41 47  
private static List<String> getAddList(List<String> nums, List<String> list) {  
List<String> newList = new ArrayList<>();  
for (int i = 0; i < list.size(); i++){  
for (int j = 0; j < nums.size(); j++){  
if (!list.get(i).contains(nums.get(j))){  
String tmp = list.get(i) + nums.get(j);  
if((tmp.contains("2") && tmp.contains("5")) || (tmp.contains("6") && tmp.contains("9"))){  
continue;  
}  
newList.add(tmp);  
}  
}  
}  
return newList;  
}  
  
private static boolean isNum(String string) {  
try {  
Integer.valueOf(string);  
return true;  
}catch (NumberFormatException e){  
return false;  
}  
}  
}