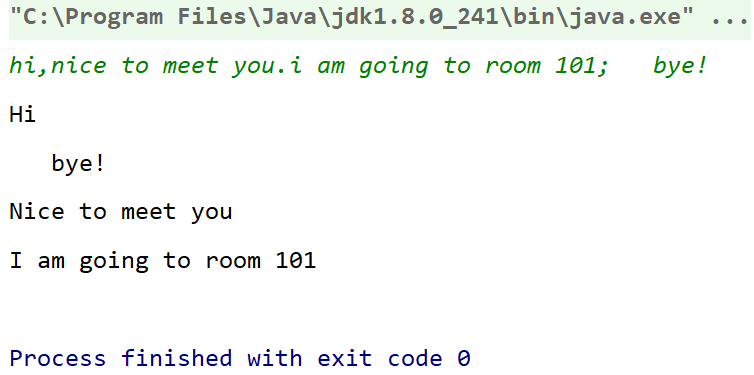
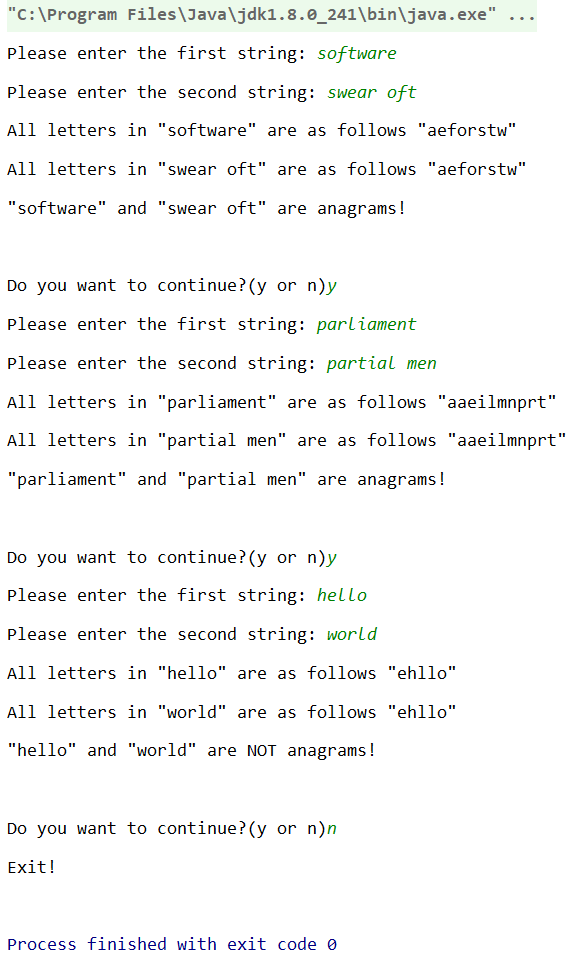
**Exercise1：**

**import java.util.Scanner;  
  
public class Exercise1 {  
 public static void main(String[] args) {  
 Scanner input=new Scanner(System.in);  
 String original=input.nextLine();  
 String[] split=original.split("[,.:;]");  
  
 for(int i=0;i<split.length-1;i++){  
 for(int j=i+1;j<split.length;j++) {  
 if (split[j].length() < split[i].length()) {  
 String a = split[i];  
 split[i] = split[j];  
 split[j] = a;  
 }  
 }  
 }//按照长度排序,注意将前后2个比较是不行的，要把每一个都和后面的所有数进行比较  
  
 for(int i=0;i<split.length;i++){  
 if(Character.isLetter(split[i].charAt(0))){//注意这里的正则表达式  
 StringBuilder splitI=new StringBuilder(split[i]);  
 String a=String.valueOf(split[i].charAt(0));  
 String A=a.toUpperCase();  
 splitI.replace(0,1,A);  
 split[i]=splitI.toString();  
 }  
 }//将第一个字符大写  
 //如果将split[i].charAt(0)先定义为char c，再对c进行修改，split[0].charAt(0)的值不会变,因为c只是一个引用  
 //charAt是一个方法，是不可以对其进行运算的，只能赋值给其他或者直接打印  
  
 for(int i=0;i<split.length;i++){  
 System.out.println(split[i]);  
 }  
  
 }  
}**



**Exercise2：**

**import java.util.ArrayList;  
import java.util.****Collections;  
import java.util.Scanner;  
  
public class Exercise2 {  
 public static ArrayList<Character> sort(String s1){  
 char[] s1LetterandBlank=s1.toCharArray();  
 ArrayList<Character> s1Letter=new ArrayList<>();  
  
 for(int i=0;i<s1LetterandBlank.length;i++){  
 if(Character.isLetter(s1LetterandBlank[i])){  
 s1Letter.add(s1LetterandBlank[i]);  
 }  
 }//删除空格和标点  
  
 Collections.sort(s1Letter);//按字母顺序排序  
  
 return s1Letter;  
 }  
  
 public static boolean areAnagram(String s1,String s2){  
 boolean result=false;  
 sort(s1);  
 sort(s2);  
  
 if(sort(s1).equals(sort(s2))){  
 result=true;  
 }  
  
 return result;  
 }  
  
 public static String conversion(ArrayList<Character> Letter){  
 char[] a=new char[Letter.size()];  
 for(int i=0;i<Letter.size();i++){  
 a[i]=Letter.get(i);  
 }  
 String s=String.valueOf(a);  
 return s;  
 }  
  
 public static void main(String[] args) {  
 Scanner input=new Scanner(System.in);  
 String determinate;  
 do {  
 System.out.print("Please enter the first string: ");  
 String s1 = input.nextLine();  
 System.out.print("Please enter the second string: ");  
 String s2 = input.nextLine();  
  
 System.out.println("All letters in \"" + s1 + "\" are as follows \"" + conversion(sort(s1)) + "\"");  
 System.out.println("All letters in \"" + s2 + "\" are as follows \"" +conversion(sort(s1)) + "\"");  
  
 if (areAnagram(s1, s2)) {  
 System.out.println("\"" + s1 + "\"" + " and " + "\"" + s2 + "\"" + " are anagrams!");  
 } else {  
 System.out.println("\"" + s1 + "\"" + " and " + "\"" + s2 + "\"" + " are NOT anagrams!");  
 }  
  
 System.out.println();  
 System.out.print("Do you want to continue?(y or n)");  
 determinate=input.nextLine();  
 }while (determinate.equals("y"));  
  
 System.out.println("Exit!");  
 }  
}**



**Exercise3：**

**import java.util.Random;  
import java.util.Scanner;  
import java.util.ArrayList;  
  
public class ClassRoom {  
 static int cnt=0;  
 int id=200+cnt;  
 int seatsNumber;  
 int[][] TheArragementOfTheClassRoom=new int[5][7];//成员变量必须在构造方法之外创建，构造方法只是赋值  
  
 ClassRoom(){  
 Random r=new Random();  
 seatsNumber=r.nextInt(90)+41;  
 for(int i=0;i<5;i++) {  
 for (int j = 0; j < 7; j++) {  
 TheArragementOfTheClassRoom[i][j]=r.nextInt(6);  
 }  
 }  
 }//构造方法  
  
 public int getID(ClassRoom cr){  
 int id=cr.id;  
 return id;  
 }  
 public int getSeatsNumber(ClassRoom cr){  
 int seatsNumber=cr.seatsNumber;  
 return seatsNumber;  
 }  
 public int[][] getArrangement(ClassRoom cr){  
 int[][] arragement=cr.TheArragementOfTheClassRoom;  
 return arragement;  
 }  
 public void toString(ClassRoom cr){  
 System.out.printf("ClassRoom[id:%d,seats:%d] plan: Room #%d with seats(%d)",cr.id,cr.seatsNumber,cr.id,cr.seatsNumber);  
 System.out.println();  
 }  
  
 public void printClassRoom(ClassRoom clsrms){  
 System.out.printf("ClassRoom[id:%d,seats:%d] plan: ",id,seatsNumber);  
 System.out.println();  
 System.out.print(" ");  
 for(int i=0;i<7;i++){  
 System.out.print("Day"+(i+1)+" ");  
 }  
 System.out.println();  
 for(int i=0;i<4;i++){  
 System.out.print((i+1)+"("+(2\*i+1)+"\_"+(2\*i+2)+")"+" ");  
 for(int j=0;j<7;j++){  
 System.out.print(TheArragementOfTheClassRoom[i][j]+" ");  
 }  
 System.out.println();  
 }  
 System.out.print("5(9\_10) ");  
 for(int j=0;j<7;j++){  
 System.out.print(TheArragementOfTheClassRoom[4][j]+" ");  
 }  
 System.out.println();  
 }//打印教室安排  
  
 public boolean isValidforExam(ClassRoom cr,int day,int time,int seats){  
 boolean result=false;  
 if (cr.TheArragementOfTheClassRoom[day-1][time-1] == 0 && seats <= cr.seatsNumber) {  
 result=true;  
 }  
 return result;  
 }//判断合格  
  
 public static ArrayList<ClassRoom> collectValidRoom(ArrayList<ClassRoom> clsrm,int day,int time,int seats) {  
 ArrayList<ClassRoom> okRooms=new ArrayList<>();  
 for(int i=0;i<clsrm.size();i++) {  
 if(clsrm.get(i).isValidforExam(clsrm.get(i),day,time,seats)){  
 okRooms.add(clsrm.get(i));  
 }  
 }  
 return okRooms;  
 }//查看合格的教室有哪些  
  
  
 public static void main(String[] args) {  
 Scanner input = new Scanner(System.in);  
  
 ArrayList<ClassRoom> clsrms = new ArrayList<>();  
 System.out.println("Please input the number of classrooms you want to generate: ");  
 int num = input.nextInt();  
  
 for (int i = 0; i < num; i++) {  
 cnt++;//静态成员变量的访问时不需要类或对象的。  
 ClassRoom cr = new ClassRoom();  
 clsrms.add(cr);  
 }  
  
 System.out.println("There are " + clsrms.size() + " Classrooms are generated");  
 for(int i=0;i<num;i++) {  
 clsrms.get(i).printClassRoom(clsrms.get(i));  
 }//先打印出所有的教室  
  
 System.out.println("Please input the info of a classroom which you want to appoint(day time seats): ");  
 int day = input.nextInt();  
 int time = input.nextInt();  
 int seats = input.nextInt();  
  
 for(int i=0;i<clsrms.size();i++){  
 if(clsrms.get(i).isValidforExam(clsrms.get(i),day,time,seats)){  
 System.out.printf("ClassRoom[id:%d,seats:%d] plan: Room #%d with seats(%d) on day\_%d\_Time\_%d is Valid for EXAM",clsrms.get(i).id,clsrms.get(i).seatsNumber,clsrms.get(i).id,clsrms.get(i).seatsNumber,day,time);  
 System.out.println();  
 }else {  
 System.out.printf("ClassRoom[id:%d,seats:%d] plan: Room #%d with seats(%d) on day\_%d\_Time\_%d is NOT Valid for EXAM",clsrms.get(i).id,clsrms.get(i).seatsNumber,clsrms.get(i).id,clsrms.get(i).seatsNumber,day,time);  
 System.out.println();  
 }  
 }//打印各教室合格的情况  
  
 ArrayList<ClassRoom> okRooms = collectValidRoom(clsrms, day, time, seats);  
 System.out.println("There are " + okRooms.size() + " rooms valid for book");  
 for(int i=0;i<okRooms.size();i++) {  
 okRooms.get(i).printClassRoom(okRooms.get(i));  
 }  
  
 input.close();  
 }  
}**

