Lab 10：“debug是快乐的话那我们大抵是乐极生悲了”

Sec0 :上次lab解析 — Hardwood Species

Tips：本问题有海量的输入，请使用scanf而不是cin，以避免Time Exceeded（但是不纳入评分考虑）

设物种名为x的树木的棵数为h[x]，树木的总棵数为n。我们首先根据输入信息统计每类树木的棵数以及树木总棵数，然后按照树木名的字典序输出每类物种在树中所占的比例。

冒泡排序，快速排序…

（当然偷懒的话使用map类关联容器，使得表元素自动按照物种的字典序排列。）

Sec 1: debug

https://blog.csdn.net/taibudong1991/article/details/124005247?ops_request_misc=%257B%2522request%255Fid%2522%253A%252216689571451680018276102 7%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=166895714516800182761027&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-2-124005247-null-null.142^v65^control,201^v3^control_2,213^v2^t3_control2&utm_term=C语言调试基础&spm=1018.2226.3001.4187

Sec2： 递归与回溯

程序调用自身的编程技巧称为递归（Recurision），是子程序在其定义或说明中直接调用自身的一种方法。

递归就是将一个大型复杂的问题层层转化成一个与原问题相似的规模较小的问题来求解，因此只需要少量的程序代码就可以描述出解题过程中所需要的多次重复计算，使得程序更为简洁和清晰。

在递归过程中，系统将每一层的返回点、局部量等用栈储存。递归时，当前层的返回点和局部变量入栈；回溯时，当前层的返回点和局部变量出栈。利用这种特性设计递归算法，程序就会变得十分简洁。需要注意的是，如果递归过程无法达到递归便捷或递归次数过多，则会造成栈溢出。

递归算法一般用来解决三类问题：

1）函数的定义是递归的（如阶乘n！或者Fibonacci函数）

2）数据结构形式按递归定义（如树的遍历和图的深度优先搜索等）

3）问题的解法是递归的（如回溯法）

特别需要指出的是，回溯法采用递归方法求解问题，堪称递归算法的一个应用典范。回溯法从初始状态出发，运用题目给出的条件、规则，按照纵深搜索的顺序递归扩展所有可能的情况，从中找出满足题意要求的解答。因此在无法建立数学模型、无法使用解析方法直接求解的情况下，通常可以使用回溯法寻求答案。

Task1 ： The Settlers of Catan

考察：递归与回溯

在1995年，德国在Settlers of Catan举办了一场游戏，玩家们要在一座岛屿的未知荒野上，通过构建道路、定居点来控制这个岛屿。

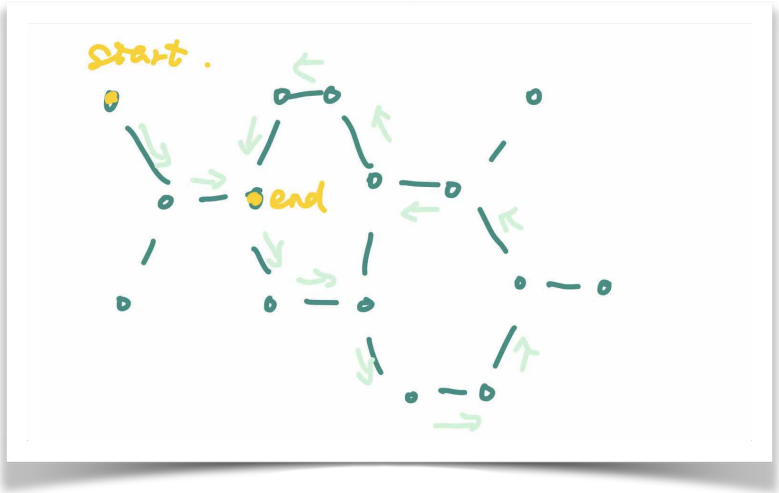
你受雇于一家软件公司（什么，在这个经济萧条的时代你居然能找到工作？），公司刚刚决定开发这款游戏的PC版，要求你实现这款游戏的一条特殊规则：当游戏结束时，建造了最长道路的玩家获得额外的两分。

问题是，玩家通常建造复杂的道路网络，而不是一条线性路径。因此，确定最长的路是不容易的（虽然玩家们通常可以用马上看出）。

与原始的游戏相比，我们仅要解决一个简化了的问题：给出一个节点（城市）的集合和一个边（路段）的集合，这些连接点的变得长度为1。

最长的道路被定义为网络中每条边都不会经过两次的最长路径，虽然节点可以被经过超过一次。

例如，下图中的网络包含了一条长度为12的道路。



输入

输入包含一个或多个测试用例。

每个测试用例的第一行给出两个整数：节点数n（2<=n<=25）和边数（1<=m<=25）。接下来的m行描述m条边，每条边由这条边所连接的两个节点的编号表示，节点编号从0到n-1。边是无向边。节点的度（一个节点连接的边的数量）最多为3。道路网不一定是连通的。

输入以n和m取0结束。

输出

对每个测试用例，在单独一行输出最长的道路的长度。

样例输入	样例输出
3 2	2
0 1	12
1 2	
15 16	
0 2	
1 2	

样例输入	样例输出
2 3	
3 4	
3 5	
4 6	
5 7	
6 8	
7 8	
7 9	
8 10	
9 11	
10 12	
11 12	
10 13	
12 14	
0 0	

Task 2 : The Sultan's Successors（八皇后问题变体）

考察：递归与回溯，格式化输出

Nubia的苏丹没有子女，所以她决定，在她去世的时候，把她的国家分成k个不同的部分。每个部分将由一些测试中表现最好的人来继承（当你继承了财产，你就不需要在上面那家软件公司打工了不是吗），有可能某个人继承多个部分或者全部。为了确保最终只有智商最高的人成为她的继承者，苏丹设计了一个巧妙的测试。在一个喷泉飞溅和充满异香的大厅里，放着k个国际象棋棋盘。每一个棋盘的方格从1到99范围内的数字进行编号，并提供8个宝石做的皇后棋子。每一个潜在的继承人的任务是将8个皇后放在棋盘上，使得没有一个皇后可以攻击另一个皇后，并且对于棋盘上所选择的皇后所占据的方格，要求方格内的数字的总和要和苏丹选择的数字一样高。（如果你不熟悉国际象棋的规则，这就是说，在棋盘上的每一行和每一列只能有一个皇后，并且在每条对角线上，最多只能有一个皇后。）

请你编写一个程序，输入棋盘的数量以及每个棋盘的详细情况，并确定在这些条件下每个棋盘可能的最高得分。（苏丹是一个好棋手，也是一个优秀的数学家，她给出的数字是最高的。）

输入

输入首先在一行中给出棋盘的数量k，然后给出k个64个数字组成的集合，每个集合由8行组成，每行8个数字，每个数字是小于100的正整数。棋盘的数量不会多于20.

输出

输出给出k个数字，表示你的k个得分，每个得分一行，向右对齐，5个字符的宽度

样例输入	样例输出
1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 ... 57 58 59 60 61 62 63 64	260