

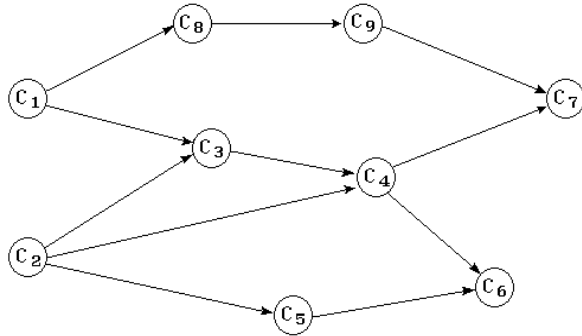
考试说明：

以下有三道大题，第一题是必做题，剩下两题任选其一完成考试。

第一题总分 50 分，第二题总分 50 分，第三题总分 50 分。

不鼓励同时完成三道大题，否则在后两题中取较低分数统计总分，请特别注意。

一、输入一个有向图，计算并输出该有向图所可能拥有的所有的拓扑序列。如果该有向图没有拓扑序列，也要输出“没有拓扑序列”的提示。



具体要求如下：

(1) 将以上有向图用邻接矩阵表示出来，存放在数组 `int graph[9][9]` 之中；（5 分）

(2) 设立一个整型数组 `inDegree[]`，用来存储所有结点的入度，在拓扑排序之前将所有顶点的入度计算出来并存储在该数组中；（5 分）

(3) 在拓扑排序的过程中，不允许重复扫描 `inDegree` 数组以发现入度为 0 的顶点，而是应该另外开辟一个队列（或堆栈）来存储那些入度为 0 的顶点序号，并动态修改该队列（或堆栈）；（5 分）

(4) 利用递归算法输出所有可能的拓扑序列，递归函数原型如：`void topSort(int graph[][9], int inDegree[], int path[], ZeroSet &zs, int k)`，其中 `graph` 是邻接矩阵，`inDegree` 存放所有顶点的入度，`path` 存放当前找到的拓扑序列，`zs` 用来存放所有当前入度为 0 的结点的序号，数据类型 `ZeroSet` 可自己定义，`k` 是数组 `path` 中当前需要填充的元素的下标，当 `k==9` 时表示发现了一条完整的拓扑序列，可以将其输出；（12 分）

(5) 需要输出该有向图的全部 215 条拓扑序列，例如以如下方式输出；（3 分）

```
214: C2, C1, C5, C3, C8, C4, C9, C7, C6
215: C2, C1, C5, C3, C8, C4, C9, C6, C7
```

(6) 如果有向图没有拓扑序列，要能够发现并给出提示；（3 分）

(7) 编写任意其他必要的函数，不许将所有代码都写在 `main()` 函数中；（6 分）

(8) 能够正确地申请空间和释放空间；（6 分）

(9) 算法的时间复杂度低，代码拥有良好的风格。（5 分）

二、六度空间理论产生于 20 世纪 60 年代，由美国心理学家米尔格伦提出，是当前风靡世界的 SNS（社交网络服务）公司所追求的重要理念，其意义在于揭示：你和一个陌生人之间都有可能通过一系列的介绍互相认识。这里对关系的“度”进行定义，如果一个人 A 和 B 认识，那么我们称二者之间的关系为 1 度关系，如果 A 和 C 原来不认识，但 A 和 B 互相认识，且 B 和 C 互相认识，那么定义 A 和 C 之间的关系为 2 度关系。一般地，如果两个人 X 和 Y 本来不认识，但通过 k 个人能将二者联系起来，那么就称 X 和 Y 之间的关系为 k+1 度关系。这里所说的关系都是双向的而不是单向的。一个人和自己的关系度数规定

为 0。

给定  $N$  个人，以及这  $N$  个人中已经存在的一些“1 度关系”，请针对每个结点计算其关系网中符合“六度空间”理论的结点占结点总数的百分比。

**【输入格式】**

第 1 行包含两个正整数  $N$  和  $M$ ，代表有  $N$  个人，以及  $M$  个“1 度关系”。

第 2 行到第  $M+1$  行每行给出一对正整数，分别是每个“度关系”中两个结点的编号（结点从 1 到  $N$  编号）。

**【输出格式】**

对每个结点输出与该结点距离不超过 6 的结点数占结点总数的百分比，精确到小数点后 2 位。每个结点输出一行，格式为“结点编号: (空格) 百分比%”。

**【输入样例】**

```
10 9
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
```

**【输出样例】**

```
70.00%
80.00%
90.00%
100.00%
100.00%
100.00%
100.00%
100.00%
90.00%
80.00%
70.00%
```

(1) 能够正确地从键盘或文件中获得输入数据；(5 分)

(2) 将输入数据转换成数据结构中的图，可以用邻接矩阵的结构存储，也可以用邻接表的结构存储；(5 分)

(3) 根据  $N$  和  $M$  的值，正确地动态申请空间，使用完毕后正确地释放空间；(5 分)

(4) 要求选用合适的非递归算法，在算法中选用合适的辅助数据结构，算法的时间复杂度要尽可能低；(18 分)

(5) 对一个浮点型变量  $x$  而言，`printf("%.2f%%\n", x*100)` 可以正确的百分比格式将其输出。要求输出所有结点的百分比。在计算百分比的时候，总人数为  $N-1$ ，即把当前结点排除在总人数之外；(7 分)

(6) 以上的输入样例和输出样例只是用来展示格式，它们之间没有内在逻辑上的联系。要求对不同的测试用例都输出正确的结果；(2 分)

(7) 正确地将算法分解到功能独立的函数中去，不许将所有代码都写在 `main()` 函数中。(8 分)

三、一棵二叉树用二叉链表的结构存储，设计一个算法计算该二叉树的宽度。也就是：哪一层上的结点数最多，二叉树的宽度就是这一层的结点数目。具体要求如下：

- (1) 不允许使用递归，而是必须用迭代（循环）的方式实现算法；（7 分）
- (2) 如果使用队列或栈作为辅助数据结构，要使用尽可能少的辅助空间完成任务；（7 分）
- (3) 能够正确地申请空间和释放空间；（7 分）
- (4) 不允许使用数组保存各个层的结点数目；（7 分）
- (5) 正确地将算法分解到功能独立的函数中去，不许将所有代码都写在 `main()` 函数中；（7 分）
- (6) 要实际构造至少一棵二叉树验证算法的正确性；（10 分）
- (7) 算法的时间复杂度低，代码拥有良好的风格。（5 分）