

上机题：

☆1. 要求用逆邻接表实现有向图。根据一个有向图的顶点序列和有向边序列构造该有向图。用逆拓扑排序算法判断该有向图是否存在回路。逆拓扑排序算法是：（1）在有向图中选择一个没有后继的顶点并输出。（2）从有向图中删除该顶点和所有以它为终点的有向边。（3）重复（1）和（2），直至当前有向图为空，或非空但不存在没有后继的顶点，前者说明原有有向图不存在回路，后者说明存在回路。

☆2. 已知由 n ($n \geq 2$) 个正整数构成的集合 $A = \{a_k | 0 \leq k < n\}$ ，将其划分为两个不相交的子集 A_1 和 A_2 ，元素个数分别是 n_1 和 n_2 ， A_1 和 A_2 中元素之和分别为 S_1 和 S_2 。设计一个尽可能高效的划分算法，满足 $|n_1 - n_2|$ 最小且 $|S_1 - S_2|$ 最大。要求：

- （1）给出算法的基本设计思想。
- （2）根据设计思想，采用 C 或 C++ 语言描述算法，关键之处给出注释。
- （3）说明你所设计算法的平均时间复杂度和空间复杂度。

☆3. 对于有向图，如果一个顶点的出度大于入度，则这个顶点称为 KJ 顶点。有向图用邻接矩阵存储，数据结构定义如下：

```
1  typedef struct {
2      int numVertex, numEdge;    //顶点数，边数
3      char VertexList[MAXV];    //顶点表
4      int Edge[MAXV][MAXV];    //邻接矩阵
5  } MGraph;
```

要求实现函数 `int printVertices (MGraph G)`，输出有向图中所有 KJ 顶点，并返回 KJ 顶点的总数。