本次实验中，主要实现了图类和相关功能，图用邻接矩阵的方式存储，主要功能有判断图是不是有向图、无向图、有向无环图、连通图，求得图的拓朴排序结果，以及得到以点1为起点，到其他所有点的单源最短路径。

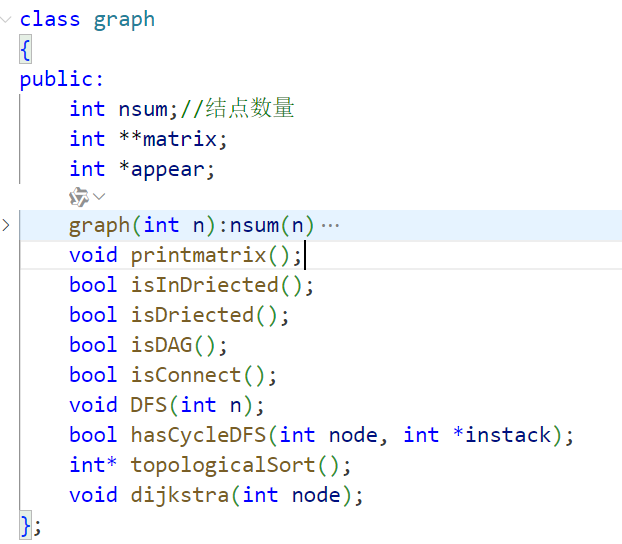
程序使用命令行参数输入，输出结果将被保存到与main.exe同目录下的result.txt中，输入格式是：先输入两个int型变量，分别代表结点数和边数，接下来m行数据均有3个int型变量，分别是边的起点、终点和边的权值。例如：



就代表一共有5个点、6条边，6后面的数字每三个就代表了一条边的数据

接下来简单介绍程序的实现方法

下图为图类的基本结构，其中,printmartix输出了图的邻接矩阵，isInDriected和isDriected分别判断图是不是无向图和有向图，isDAG判断图是不是有向无环图，isConnect判断图是不是连通图，DFS和hasCycleDFS是用于求是否是有向无环图和连通图的辅助搜索函数，topologicalSort是求拓朴排序的函数，dijkstra是求点node的单源最短函数，这里如果没有的话将输出“”



以下为一个输出结果，图中含有5个点，6条边，输出数据中有它的邻接矩阵，这个图是有向图，是有向无环图，但不是连通图，拓朴排序的结果是5 1 4 2 3，点1到2和3 的最短路径分别是2和9，无法到达4和5，因此不存在最短路径



实现思路：

判断是否是有向图和无向图时，遍历整个邻接矩阵，如果对任意的i，j，都有i到j的边权值等于j到i的边权值，就是无向图，如果存在一对ij使得i到j的边权值不为MAXINT（也就是存在一条边）且i到j的边权值≠j到i的边权值，就是有向图。

判断是否是有向无环图的时候，先判断是否是无向图，若是就直接return false，否则开始依次遍历每个点，通过hasCycleDFS和栈，每次把当前结点所有有边且未访问的邻居结点入栈，如果栈中已存在该结点，就说明不是有向无环图。

判断是否是连通图时， 从结点1开始进行依次深度优先搜索，然后检查所有结点，如果存在未被访问的结点，就说明图有不止一个连通分量，也就不是连通图。

求拓扑排序时，先计算所有结点的入度，然后依次寻找入度为0的点，存到结果数组并删除该结点和与该结点关联的边，并把与这些边关联的结点入度-1，最终把结果数组翻转即为拓扑排序结果。

求单源最短路径即为dijkstra算法，简单默写代码即结束。