项目说明文档

数据结构课程设计

——排课软件

作 者 姓 名： 董震宇

学 号： 1852143

指 导 教 师： 张 颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

**目录**

[一、项目分析 4](#_Toc27425553)

[1. 项目内容 4](#_Toc27425554)

[2. 项目要求 4](#_Toc27425555)

[二、概述 4](#_Toc27425556)

[三、设计 5](#_Toc27425557)

[1.数据结构设计 5](#_Toc27425558)

[2.类结构设计 5](#_Toc27425559)

[3.成员与操作设计 5](#_Toc27425560)

[（1）课程结构体 5](#_Toc27425561)

[（2）Arrangement类 5](#_Toc27425562)

[四、具体实现 6](#_Toc27425563)

[1.总体思路 6](#_Toc27425564)

[2.读取信息 6](#_Toc27425565)

[（1）简要思路 6](#_Toc27425566)

[（2）代码 7](#_Toc27425567)

[3.更新“入度”数组 7](#_Toc27425568)

[（1）简要思路 8](#_Toc27425569)

[（2）算法流程图 8](#_Toc27425570)

[（3）核心代码 8](#_Toc27425571)

[4.拓扑排序 9](#_Toc27425572)

[（1）简要思路 9](#_Toc27425573)

[（2）算法流程图 9](#_Toc27425574)

[（3）核心代码 10](#_Toc27425575)

[5.文件输出课程表 11](#_Toc27425576)

[（1）简要思路 11](#_Toc27425577)

[（3）核心代码 12](#_Toc27425578)

[五、运行测试 13](#_Toc27425579)

[1.成功运行 14](#_Toc27425580)

[（1）输入文件显示 14](#_Toc27425581)

[（2）控制台 14](#_Toc27425582)

[（3）输出文件 14](#_Toc27425583)

[2.课程总数与各学期课程数之和不相等 16](#_Toc27425584)

[六、注意事项 16](#_Toc27425585)

[1.关于读取文件的格式 16](#_Toc27425586)

[2.输出文件 16](#_Toc27425587)

[七、自我小结 16](#_Toc27425588)

一、项目分析

1. 项目内容

大学的每个专业都要进行排课。假设任何专业都有固定的学习年限，每学年含两学期，每个专业开设的课程都是确定的，而且课程在开设时间的安排必须满足先修关系。每门课程有哪些先修课程是确定的。每门课恰好占一个学期，假定每天上午与下午各有5节课。是在这样的前提下设计一个教学计划编制程序。

2. 项目要求

（1）输入数据包括：个学期所开的课程数（必须使每学期所开的课程数之和与课程总数相等），课程编号，课程名称，周学时数，指定开课学期，先决条件。如指定开课学期为0，表示有电脑自行指定开课学期。

（2）如输入数据不合理，比如每学期所开的课程数值和与课程总数不相等，应显示适当的提示信息。

（3）用文本文件存储输入数据，并且读入计算机。

（4）用文本文件存储产生的各学期的课表。

二、概述

我们把大学全部课程看作一个大的工程，每节课的学习就是工程的一些活动，完成了这些活动，这个工程就可以完成了。

课程与课程之间存在先修关系，我们可以利用有向图来表示这种先修关系。在有向图中，顶点表示课程学习活动，有向边课程学习活动之间的先修关系。这个有向图就被称为顶点表示活动的网络，即AOV网络。

由于本题要求较多，将全部课程按照先修关系排好序后，还需要根据课程指定开课学期、学时以及每学期课程数，将所有课程分配到课程表中。所以进行拓扑排序后，再根据要求进行操作，从而实现项目要求。

三、设计

1.数据结构设计

由于课程之间存在先修关系，所以排课实际上可以看做是活动的网络，故采用图数据结构。而课程表的输出采用二维数组数据结构。

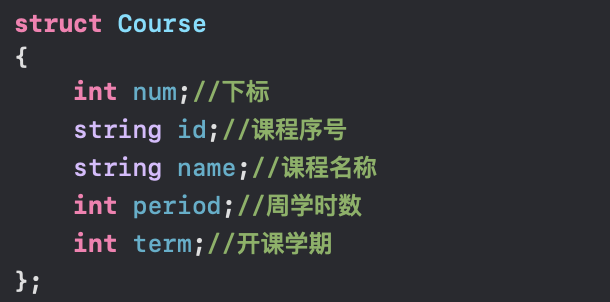
2.类结构设计

设计一个课程结构体，其中成员包括课程序号、课程名称、开课学期、周学时数等。

设计一个排课类，包括各函数实现以及课程信息及辅助信息的实现。

3.成员与操作设计

（1）课程结构体



num是指该课程在邻接矩阵中的下标，也是AOV或AOE活动网络的序号。

（2）Arrangement类



其中关于课程的信息有课程总数、每学期课程数、课程之间的先修关系 （邻接矩阵），以及辅助存储：入度为0的队列、入度数组和课程先后顺序，课程表。

函数部分通过ReadInfo()从文件中读入课程所有信息，由于采用AOV活动网络的处理——进行拓扑排序，所以有辅助函数getIndegree()，更新入度数组。在建立课程表时，也要根据课程的指定信息将课程插入表中，固有辅助函数InsertCourse()。

四、具体实现

1.总体思路

读入信息——》拓扑排序——》按照课时等要求排课——》文件输出课程表。

2.读取信息

（1）简要思路

将课程信息分为4个部分，第一部分是课程总数和每个学期课程数，如果每个学期课程数之和与课程总数不相等则抛出异常，要求用户检查，结束程序。第二部分是课程信息，按照序号、名称、学时、开课学期一次排列。第三部分是先修关系，如c02的先修课程是c01和c03，则输入c02 c01 c02 c03，每次建立一个先修关系，输入-1表示先修关系结束。

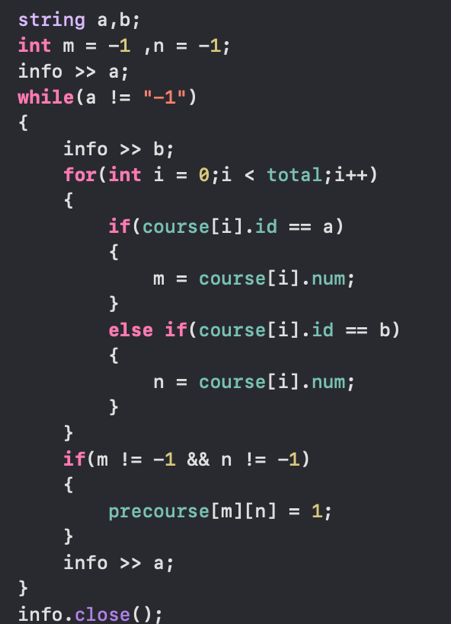
（2）代码

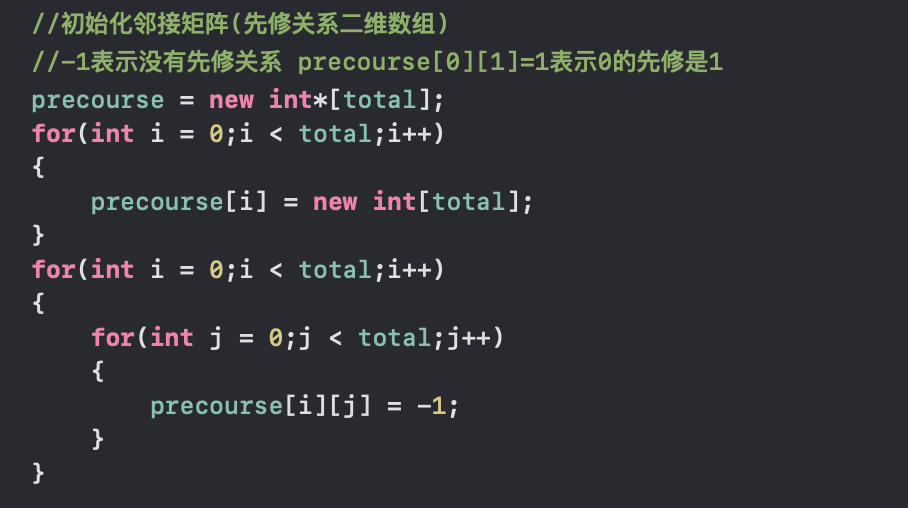
第一部分：用ifstream()打开文件并开始读取总数与各个学期课程数，若不相等则退出程序。



第二部分：依次读入课程信息



第三部分：读入先修关系并关闭文件



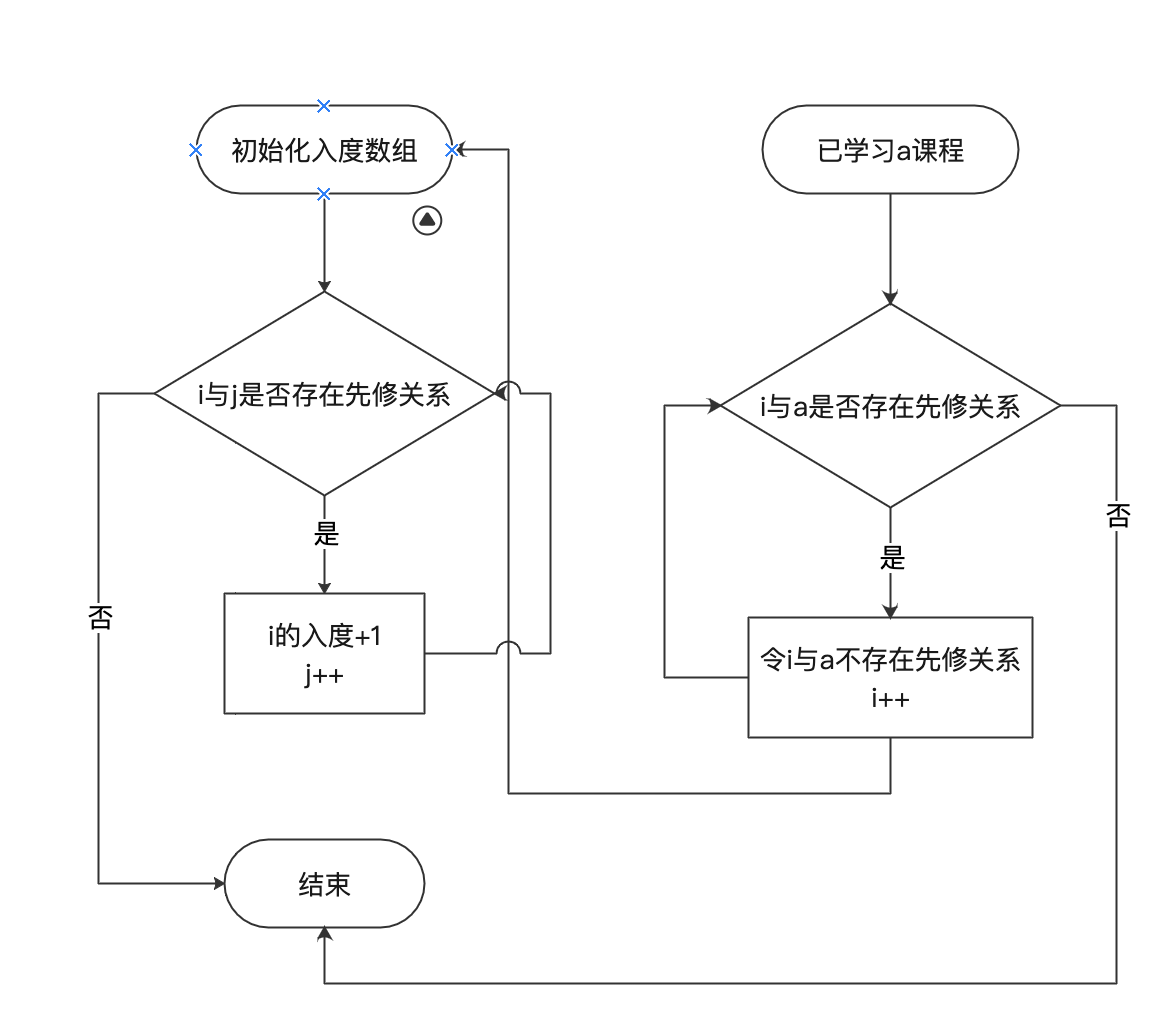
3.更新“入度”数组

（1）简要思路

初始化：indegree数组全部初始化为0，若previous[i][j] == 1（i，j之间有先修关系），则indegree[i]++，邻接矩阵中，某一顶点的入度为矩阵一行中所有的“1”相加。

更新：若已经学习某门课程，则将与这门课程相关的所有先修关系改为-1，然后再次调用上面的初始化函数重新获得入度数组。

（2）算法流程图



（3）核心代码

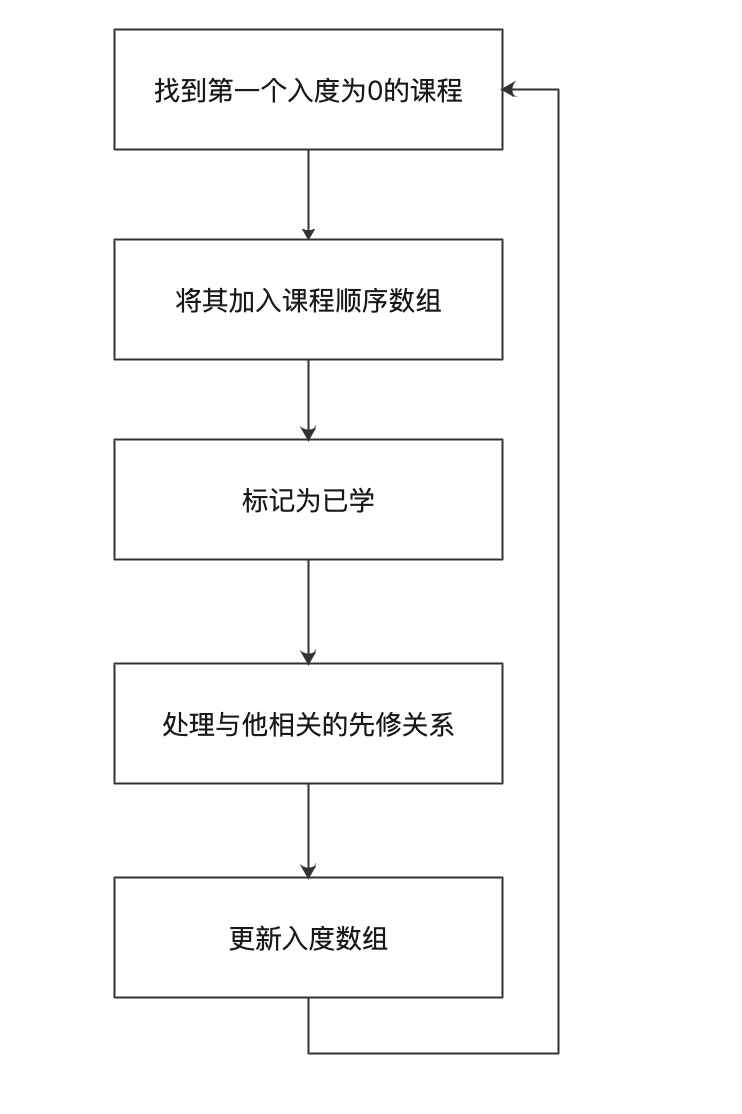


4.拓扑排序

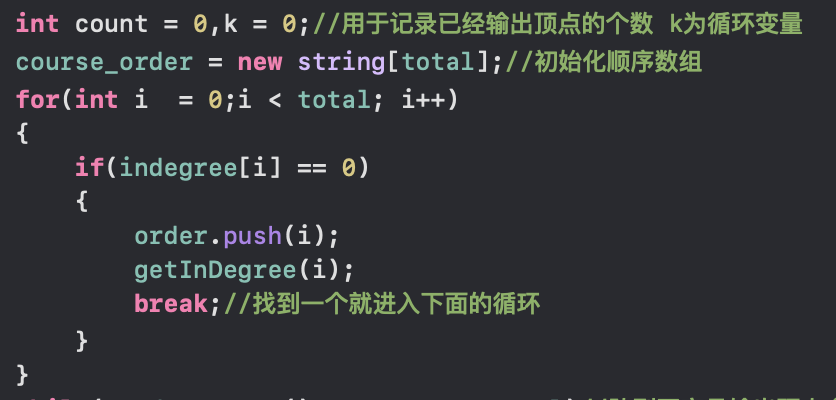
（1）简要思路

在全部课程中，找到一个入度为0的课程，将他加入课程顺序数组中，删去这个课程（标记为已经学过），并且处理它的先修关系，然后更新入度数组，循环进行以上步骤，直至全部课程学完。如果课程没有学完但是已经没有入度为0的课程了，则说明AOV网络中存在有向环。

（2）算法流程图



（3）核心代码





5.文件输出课程表

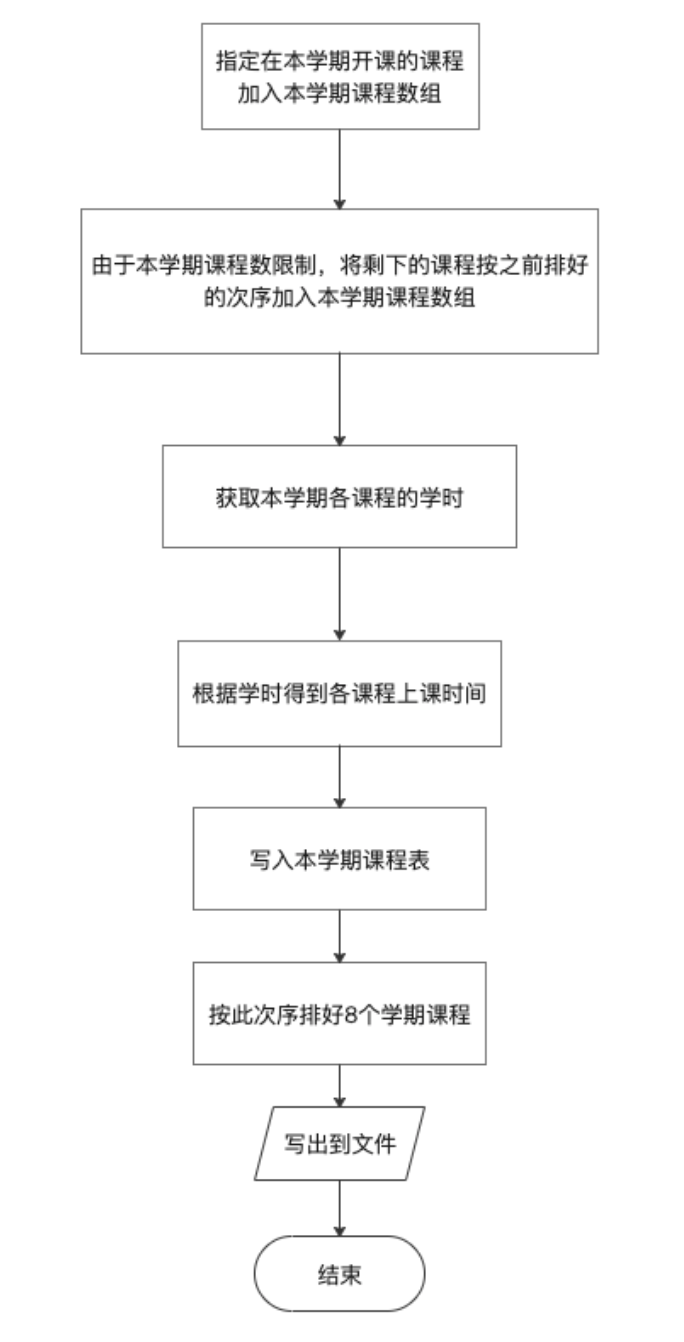
（1）简要思路

因为有8个学期，所以循环进行8次。

在循环内，新建本学期课程数组、本学期课程学时数组以及初始化本学期课程表。然后进行本学期排课。

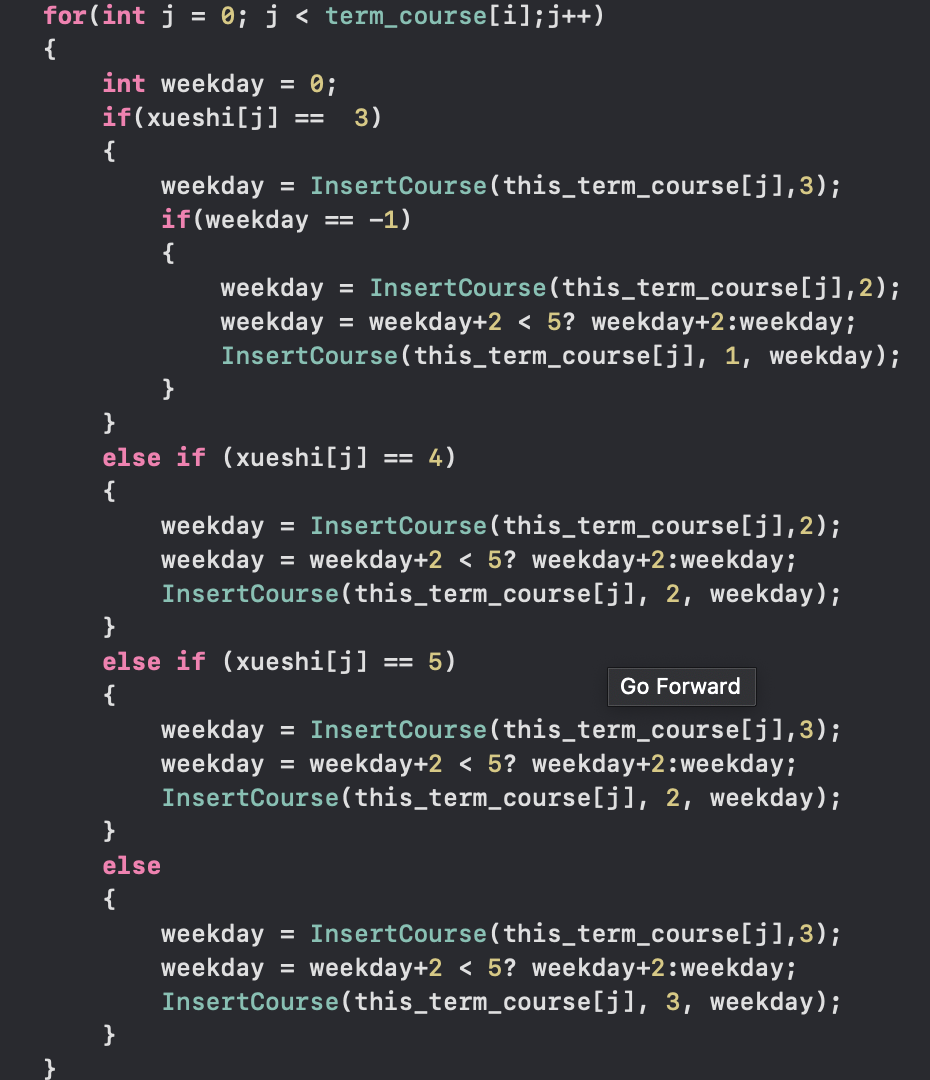
先将指定在本学期开课的课程加入到本学期课程数组中，然后将排好序且未指定开课学期的课程加入到本学期课程数组中，并记录他们的学时。根据不同的学时，运用InsertCourse()函数获取该课程按要求应该排的星期，再将他们插入课程表中。最后将课程表写出到文件中。

（2）算法流程图



（3）核心代码

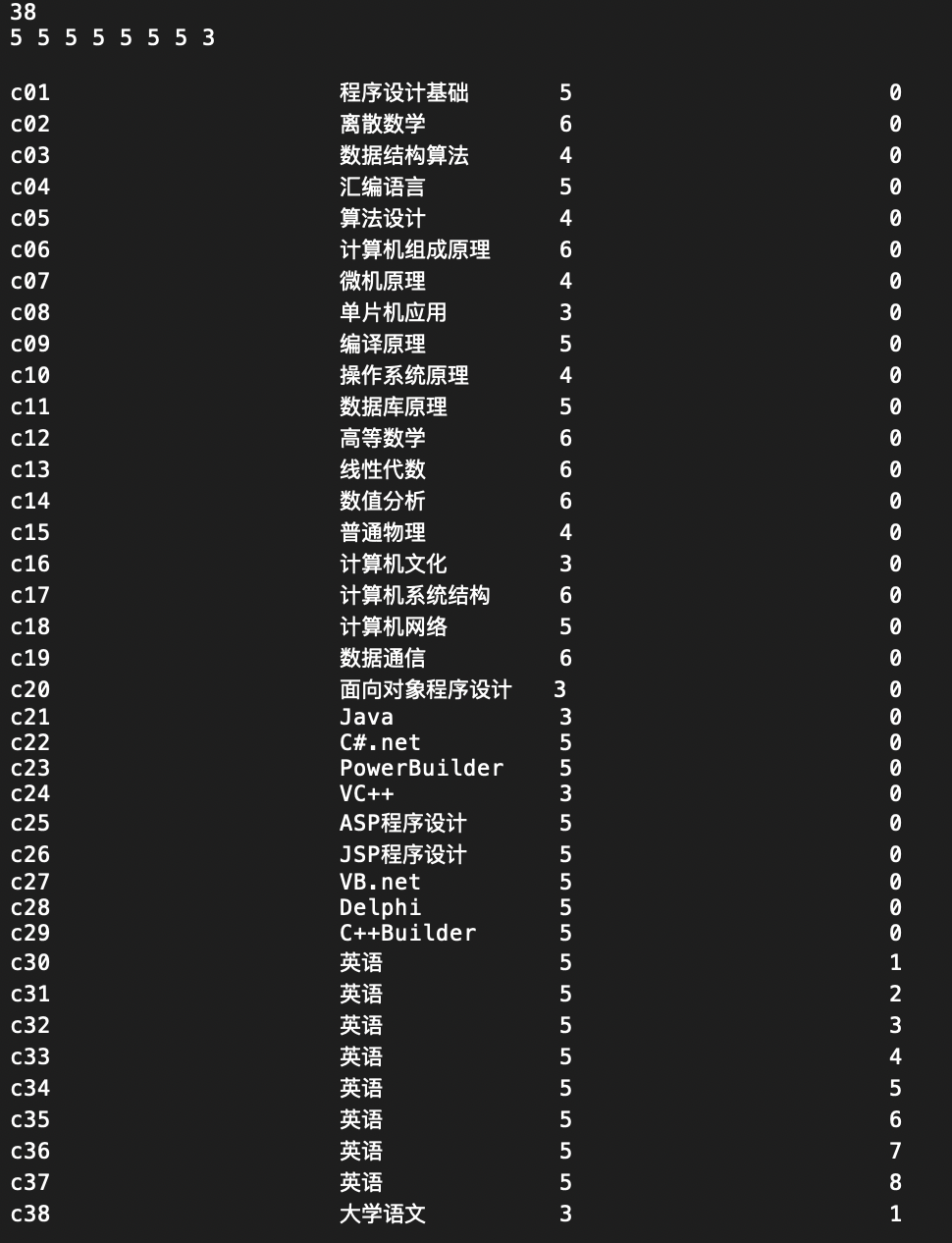
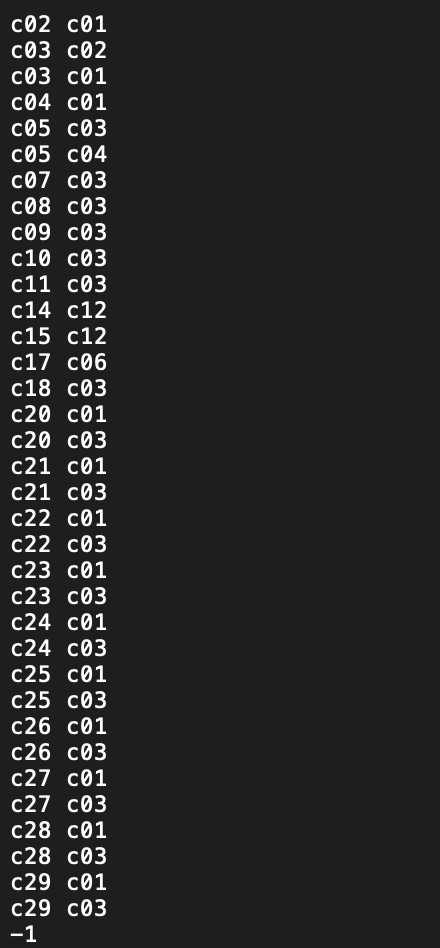




五、运行测试

1.成功运行

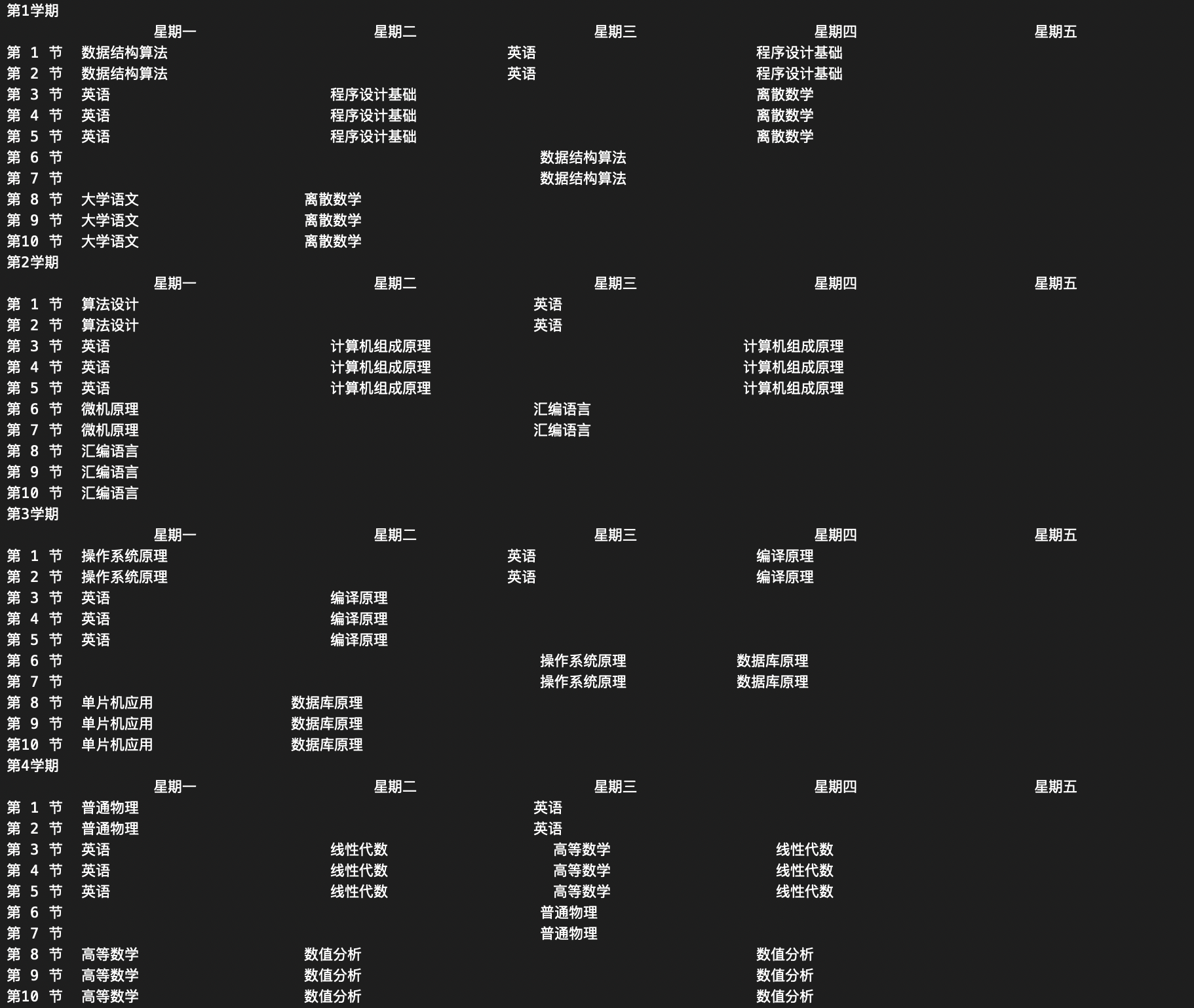
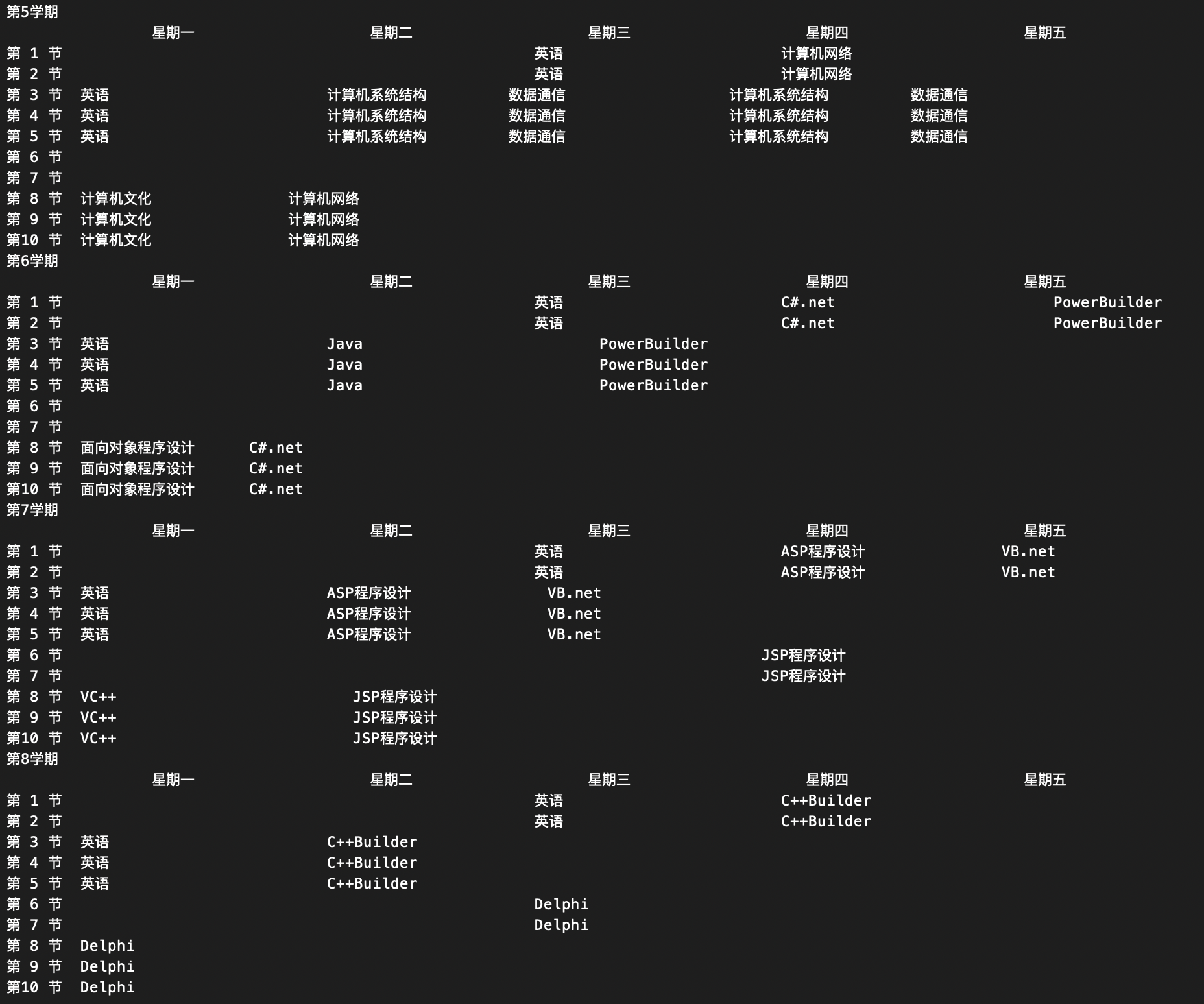
（1）输入文件显示



（2）控制台



（3）输出文件



2.课程总数与各学期课程数之和不相等



六、注意事项

1.关于读取文件的格式

课程总数

各学期课程数（8个）（这九个数字用空格分开，可以不换行）

各课程序号 名称 学时 开课学期（用空格分开，可以不换行）

先修关系格式看上图示例。

2.输出文件

输出的txt文件存放在程序目录下。

七、自我小结

本题由于要求较多，实现起来在细节处很容易弄混，比如我创建的各个辅助数组，经常搞混他们的作用。

AOV网络的拓扑排序在每输出一个顶点后，要删去这个顶点以及与它相连的全部边，再次进行排序，直至网络中所有顶点均输出或者发现网络中有有向环。

本题设计中也采用了函数重载与文件读写，将一些新的没有用过的东西加以应用，收获颇丰。

这个项目用时较长，完成的时候看到自己排好的课表也是有一种成就感。