**深 圳 大 学 实 验 报 告**

**课程名称：­ 计算机系统(1)**

**实验项目名称： 简单游戏设计：简易四子棋**

**学院： 计算机与软件学院**

**专业： 软件工程**

**指导教师：李志**

**报告人：郑彦薇 学号：2020151022 班级： 软工01班**

**实验时间： 2021/6/3**

**实验报告提交时间： 2021/6/9**

**教务处制**

**一、实验目的**

**分析和理解指定的需解决问题。**

**利用LC-3的汇编代码设计实现相关程序。**

**通过LC-3仿真器调试和运行相关程序并得到正确的结果。**

**二、实验内容**

**在LC-3中实现简易四子棋的游戏，通过键盘键入玩家选择的落子位置--列，输出玩家落子后的棋盘，并判断当前情况是否已得到结果（玩家一胜/玩家二胜/平局：输入次数减为0），若已得到结果，程序结束，若未得到结果，玩家可输入次数减一，重复上述操作，实现在得出结果之前对棋盘进行更改、打印、判断的循环。**

**程序从x3000开始，输入与打印及当前结果都在Console窗口中进行与显示，利用子程序完成整个个大程序。**

**三、实验步骤与结果**

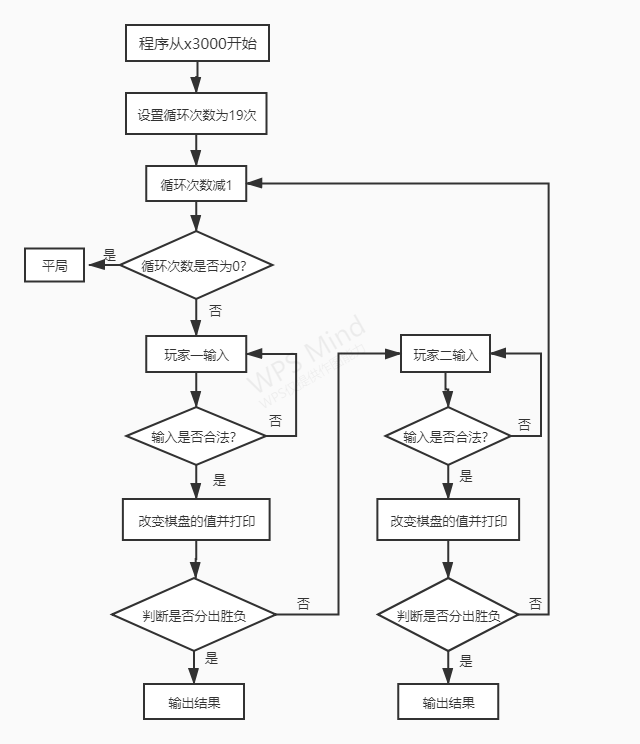
**（一）问题的解决思路：**

对简易四子棋程序的编写，包括打印棋盘、四种胜负情况或平局的判断、输入的合法性等。我们将程序分成主要的三个部分：

1. 设置6✖6的双重循环，逐次进行内循环与外循环，对大小为36的棋盘进行打印，将-设置为0，玩家一的棋子O设置为1，玩家二的棋子X设置为-1，通过正负性的判断打印出相应的内容；
2. 胜负的判断用其中一个玩家的结果来表示，设置双重循环进行枚举，统计一个判断循环内玩家一的棋子数，为4说明玩家一已获胜，为0说明玩家二获胜，不为0或4或出现“-”，说明当前四子没有连成线，难分胜负，不做判断。
3. 对输入的数的合法性进行判断，首先是输入的数字只能是1~6，且每个数字最多只能输入6次，可以通过遍历矩阵，若当前输入列数所在列已满，则输入不合法。

**（二）流程图：**

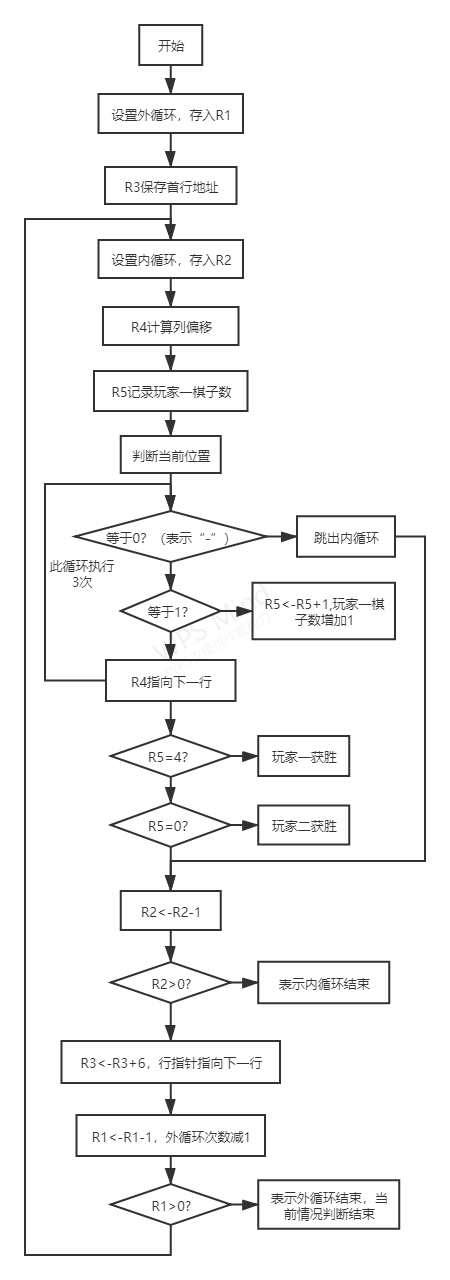
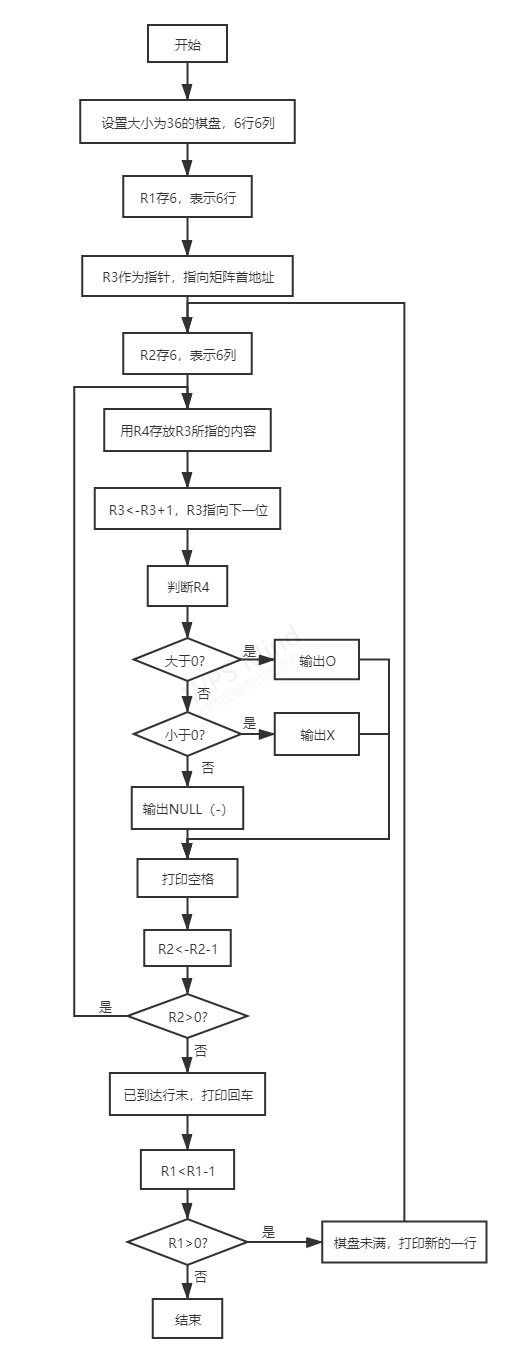
根据上述思路，我们可以得到解决问题的流程图如下：



**模块一：打印棋盘的具体流程：**

**模块二：四种情况判断（| \ / —）**

**说明：四种情况区别在于内循环和外循环的次数不同，以及在判断某个位置时指针的具体偏移情况有所区别，其他一致，这里用一个流程图为代表**

****

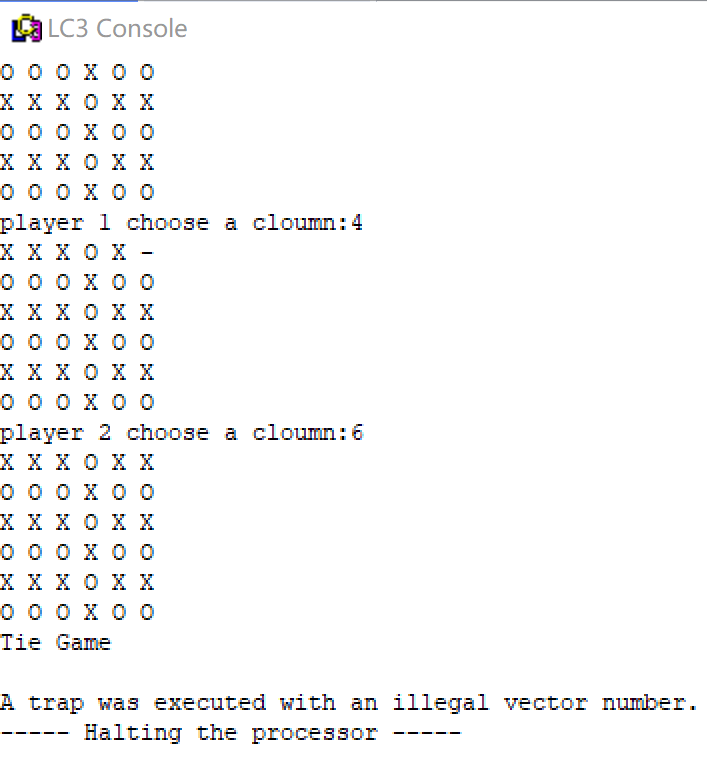
**（三）问题解决（编程实现）**

**根据上述流程图，我们可以编写出打印棋盘与判断的子程序，再通过总流程图对主程序进行编写，再将子程序与主程序合并为一个大程序。（注意在程序中对流程图语言描述的棋子代表符号进行赋值）**

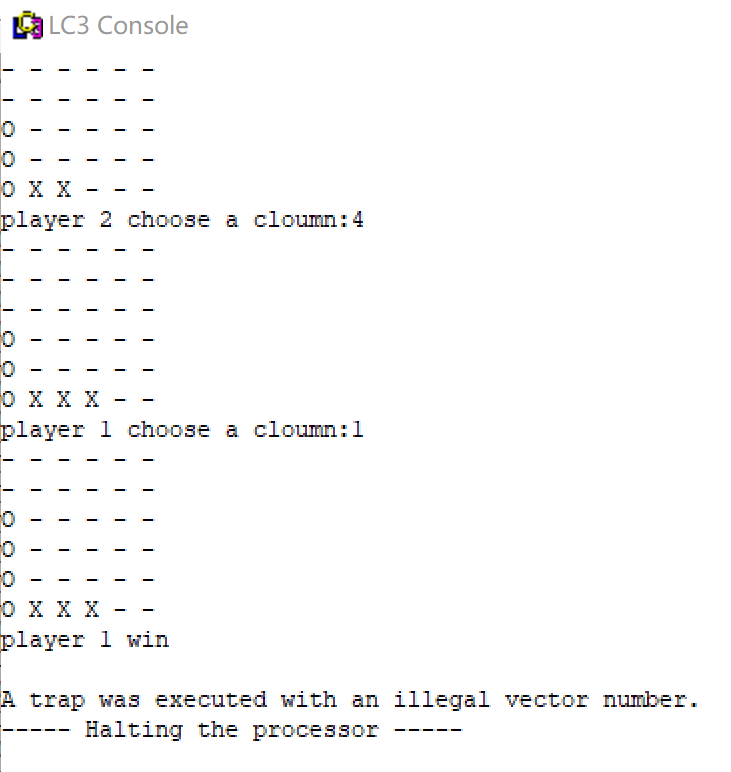
**具体程序编写见所带附件connect4.asm**

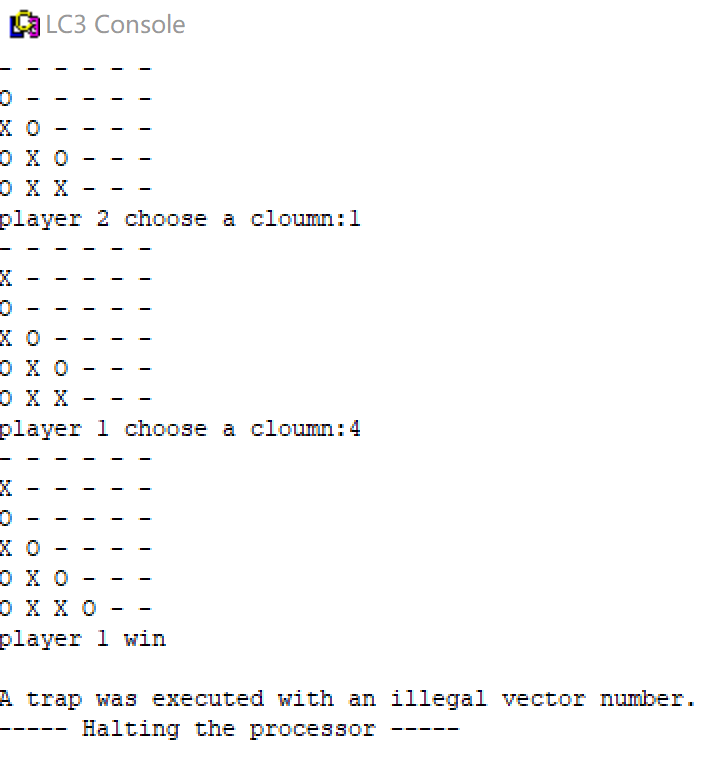
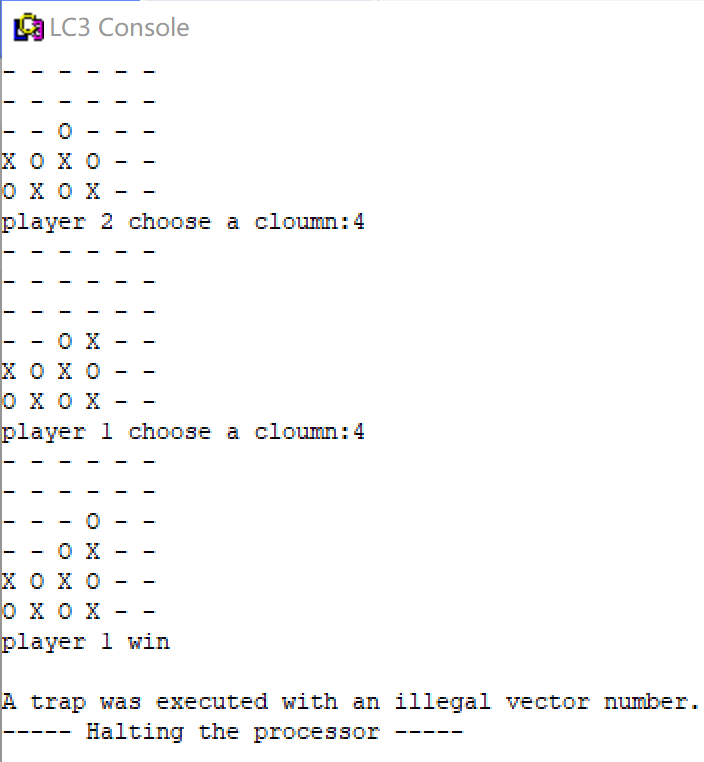
**结果展示：**

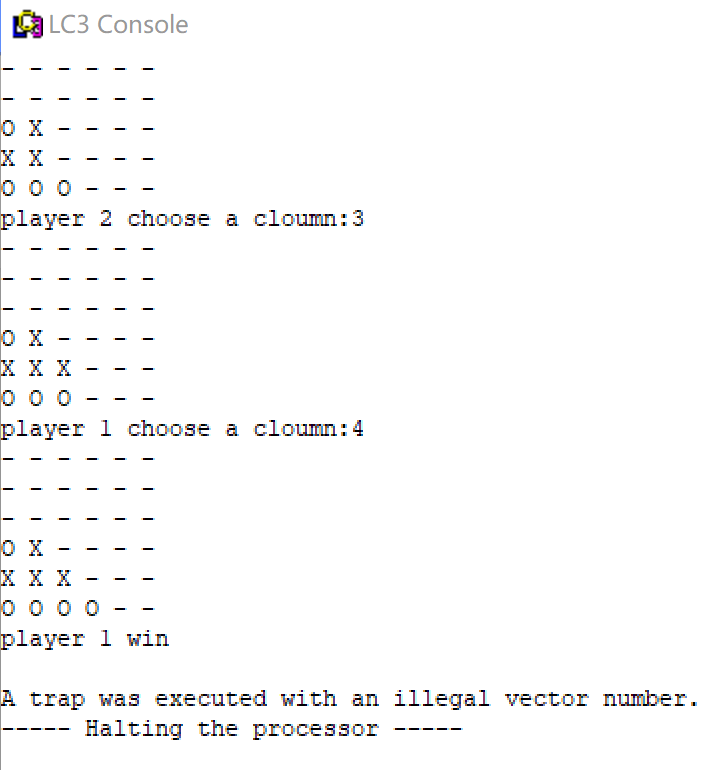
1. **Tie Game：**

****

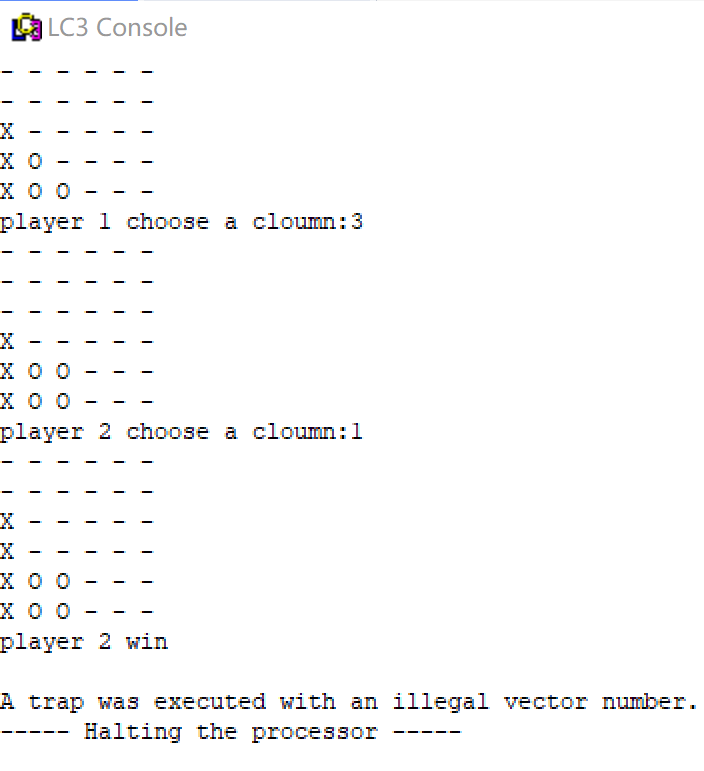
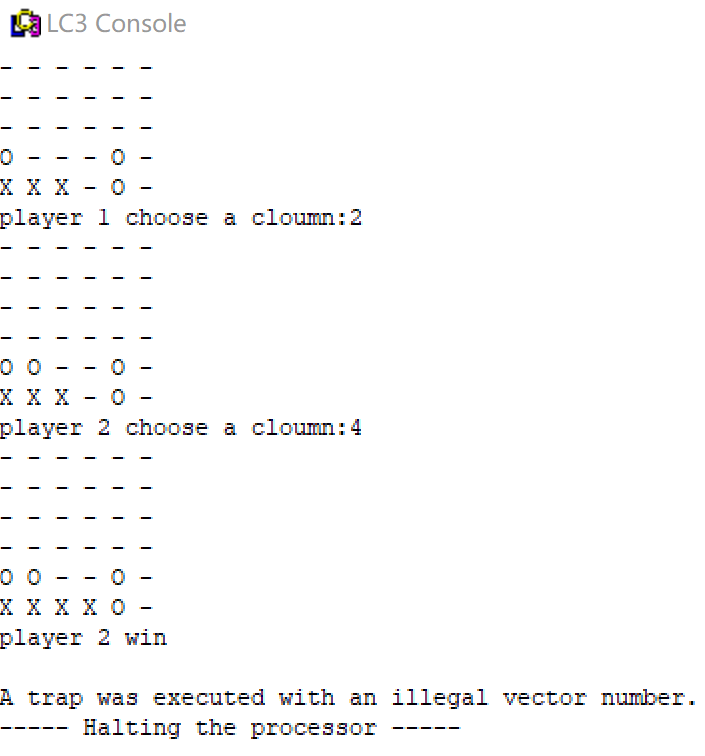
**（二）Player1 wins：**

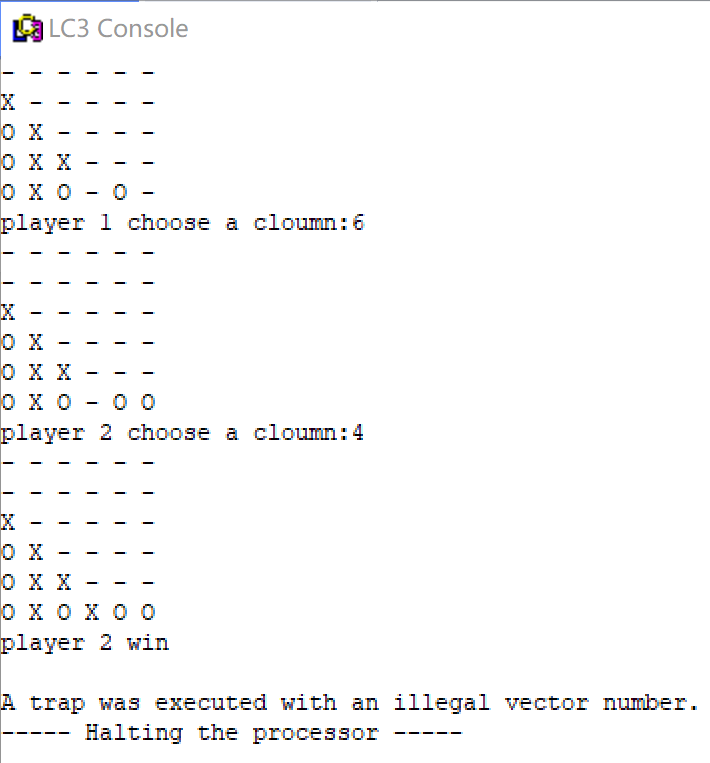
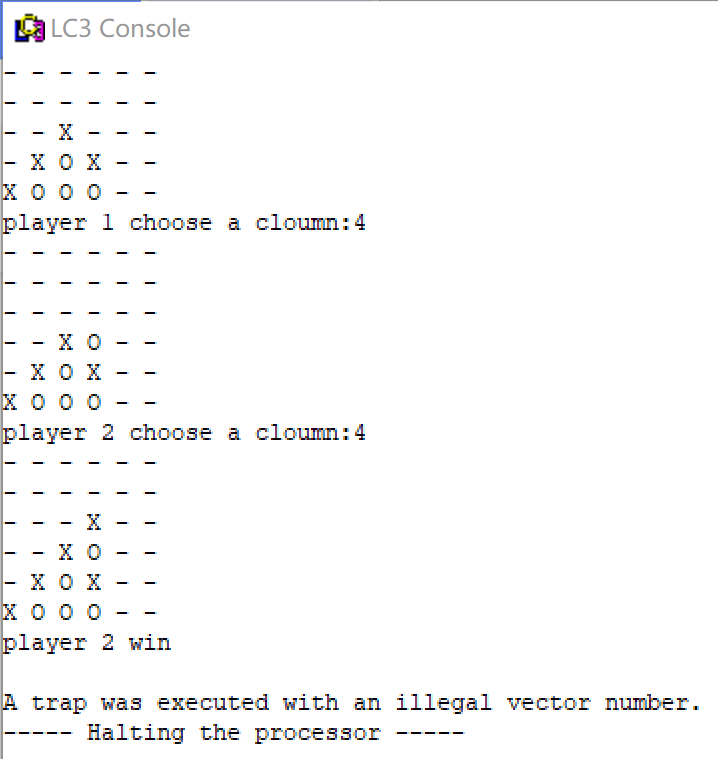
****

****

****

**（三）Player2 wins：**





**四、实验结论**

**该实验完成了对一个大程序的编写，实现了通过汇编语言实现简易四子棋游戏的设置，在解决这一问题时，我们可以得到以下结论：**

**·**在进行大程序的编写时，我们应该将大程序分成小模块，编写相应子程序，通过分块编写来增加程序的可读性和可理解度。

**·**编写程序时我们知道，该程序有500多行代码，而偏移量只有256位，为解决这一问题，我们需要在程序中间进行中转，且应适当的将一部分伪操作（.FILL）放在程序中间而不是全部写在程序最后。

**·**对寄存器的功能管理是编写大程序的关键，分配好每个寄存器的功能，可以帮助我们更好的进行程序的编写及方法设计。

**·**在该程序中，我们需要注意结果保存在R7寄存器当中，因此在对程序进行编写时应注意不能改变R7的值。

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。