**深 圳 大 学 实 验 报 告**

**课程名称：­ 计算机系统**

**实验项目名称： 简易四子棋**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师：**

**报告人： 学号： 班级：**

**实验时间： 2022年6 月 8 日 星期 三**

**实验报告提交时间： 2022年6 月 10 日 星期五**

**教务部制**

|  |
| --- |
| **实验目的：**   1. 分析和理解实验指定的问题。 2. 学习并熟悉掌握子程序的编写、使用。 3. 利用LC-3实现较为复杂的程序 |
| **实验内容与实验要求:**  本实验需要在LC-3中实现简易版四子棋的游戏，两位选手通过键盘和输出窗口轮流交互操作，棋盘由6 X 6的网格组成。  游戏规则如下：  两位选手依次轮流落子；  选手不能悔棋；  有子的地方不能继续落子；  直到有一方的四个棋子能够连成一条水平线、垂直线或者是对角线；  如果棋盘已满，无人获胜，则平局。  游戏最初时应该打印空的棋盘，可以用ASCII码"-" （即ASCII 码 x002D）来表示该处为空，"O"(ASCII 码 x004F)表示第一位选手的棋子，"X" (ASCII 码 x0058)来表示第二位选手的棋子，为了让棋盘更易于观察，在各列间加一个空格，第6列之后不要添加，初始棋盘应该如下：  - - - - - -  - - - - - -  - - - - - -  - - - - - -  - - - - - -  - - - - - -  选手一始终先下第一步棋，然后两者轮流落子，在每次落子之后，应该打印该选手的信息，提示他落子，以选手一为例，应该打印信息如下：  Player 1, choose a column:  为了明确选手的落子的位置，该选手应该输入数字1-6，然后回车，数字1-6指示在落子所在的列，从左到右，无需输入行号，程序应默认从行号6到行号1递减的顺序填入该棋子，若前后输入的列号相同，则行号减一。例如，如果选手第一次在左起第二列落子，应该输入2，然后回车，则该棋子落在行6列2处，当后面输入的列号再次为2时，则将棋子落子行5列2处，以此类推，详情见后续示例输出。程序应该确保选手输入的数字对应正确的列的范围，如果输入不合理，应该输出一条错误信息，提示该选手继续输入，例如，如果对于选手一：  Player 1, choose a column: D  Invalid move. Try again.  Player 1, choose a column: 7  Invalid move. Try again.  Player 1, choose a column:  程序应该一直提示该选手，知道输入正确的数字，当用户输入完成，程序应通过显示回馈给选手，然后通过换行符(ASCII 码 x000A)换行。  当选手输入成功后，程序应打印更新后的棋盘，并检查是否有人获胜，如果没人获胜，则轮到下一位输入。  当其中一位获胜或平局时，游戏结束，程序显示最后的棋盘情况并终止（Halt）。例如，如果选手二有四子相连，应该输出：  Player 2 Wins.  如果平局，程序应该输出：  Tie Game. |
| **实验步骤**   1. 分析题目，题目要求设计一个简易四子棋，根据题意画出游戏进行的流程图，如下。      1. 将会循环用到的模块设计为子程序。   我设计了五个子程序，分别是打印棋盘，判断|、判断—、判断\、判断/。  子程序1：打印棋盘；  用双重循环，判断矩阵中的元素，>0是选手一的子，<0是选手二的子，=0是还没有下棋。注意每行最后输出换行。  子程序2：判断竖着的；  只需判断前三行的就行，用一个3x6的循环，从第一行第六个开始，指针加6即可判断下面的元素。  子程序3：判断横着的  只需判断前三列的就行，用一个6X3的循环，从第一行的第三个开始，指针加1即可判断右边的元素。  子程序4：判断斜着的\  只需要判断左上角的3X3九个格子，从第一行的第三个开始，指针加7即可指向右下角的那个元素。  子程序5：判断斜着的/  只需要判断右上角的3X3九个格子，指针加5即可指向左下角的那个元素。  在子程序2-5中，判断为玩家一 的棋子后，计数器加一，若判断完毕后玩家一的棋子数为4，则玩家一获胜。若玩家一棋子为0，玩家二的棋子为4，则玩家二获胜；   1. 在主程序中接收输入的数据，输出提示信息。 2. 因为总代码行数超过了256，所以需要跳转。 3. 将各个子程序与主程序结合起来，下面为最终的代码。   ;主程序  .ORIG X3000  BRnzp pre\_MAIN  ;子程序1 打印棋盘  ;-------------------------------------  DISPLAY ST R0,SAVER0;  ST R1,SAVER1;  ST R2,SAVER2;  ST R3,SAVER3;  ST R4,SAVER4;  ST R5,SAVER5;  ST R6,SAVER6;  ST R7,SAVER7;  AND R3,R3,0; (R3=i)  ADD R3,R3,#6; 令R3=6  LEA R6,mat0x;  loop1 AND R4,R4,#0; (R4=j)R4=1时要换下一行，要输出回车；  ADD R4,R4,#6;    loop2 LDR R1,R6,#0 ;矩阵的数据存入R1  ADD R6,R6,#1 ;下一个数据  ADD R1,R1,#0 ;  BRn player2 ;(<0,输出“X ”,玩家2是X）  BRz nothing ;（ =0，输出- ）  BRp player1 ;(>0,输出“O”，玩家一是O )  player1 LD R0,PLAYO;  trap x21  BRnzp SorE; ;跳转到输出空格  player2 LD R0,PLAYX;  trap x21  BRnzp SorE;  nothing LD R0,PLAYnon;  trap x21    SorE LD R0,SPACE;  TRAP X21 ;out  ADD R4,R4,#-1 ;如果不是这行最后一个，继续输出  BRp loop2;    LD R0, ENTER; 打印回车 ;是这行的最后一个的话，换行  TRAP x21  ADD R3, R3, #-1  BRp loop1 ; 循环1到此结束      LD R0, SAVER0 ; 恢复寄存器  LD R1, SAVER1  LD R2, SAVER2  LD R3, SAVER3  LD R4, SAVER4  LD R5, SAVER5  LD R6, SAVER6  LD R7, SAVER7  RET  ;----------------------------------------------------  ;子程序2 竖着的|  CASE1 ST R0,SAVER0;  ST R1,SAVER1;  ST R2,SAVER2;  ST R3,SAVER3;  ST R4,SAVER4;  ST R5,SAVER5;  ST R6,SAVER6;  ST R7,SAVER7;  AND R1,R1,#0;  ADD R1,R1,#3; R1=i,R1=3,外层循环;  LEA R3,mat0x;  J1\_L1 AND R2,R2,#0; R2=j,R2=6,内层循环；  ADD R2,R2,#6;  J1\_L2 ADD R4,R3,#-1;  ADD R4,R4,R2; R4指向这一行的第六个，然后第五个第四个第三个第二个第一个  AND R5,R5,#0; R5中放有几个P1的棋；  LDR R0,R4,#0; 看这一格里是啥棋 1 or -1 or 0;  BRz j1\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1 ;P1的子  BRz J1ADD1 ;  BRnzp j1\_ad1  J1ADD1 ADD R5, R5, #1  j1\_ad1 ADD R4, R4, #6 ; R4指向下一行  LDR R0, R4, #0  BRz j1\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J1ADD2 ;  BRnp j1\_ad2  J1ADD2 ADD R5, R5, #1  j1\_ad2 ADD R4, R4, #6 ; R4指向下一行  LDR R0, R4, #0  BRz j1\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J1ADD3 ;  BRnp j1\_ad3  J1ADD3 ADD R5, R5, #1  j1\_ad3 ADD R4, R4, #6 ; R4指向下一行  LDR R0, R4, #0  BRz j1\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J1ADD4 ;  BRnp j1\_ad4  J1ADD4 ADD R5, R5, #1  j1\_ad4 ADD R6, R5, #-4 ; 只要能走到这里必定为4个1或-1  BRz pre\_P1WIN ; 如果R5为4则p1赢  ADD R5, R5, #0  BRz pre\_P2WIN ; 如果R5为0则p2赢  j1\_l2ed ADD R2, R2, #-1  BRp J1\_L2 ; 内循环到此结束    ADD R3, R3, #6 ; R3指向下一行  ADD R1, R1, #-1  BRp J1\_L1 ; 外循环到此结束  ; 判断|结束    LD R0, SAVER0 ; 恢复寄存器  LD R1, SAVER1  LD R2, SAVER2  LD R3, SAVER3  LD R4, SAVER4  LD R5, SAVER5  LD R6, SAVER6  LD R7, SAVER7  RET  ;---------------------  ;子程序3 判断横着的—  CASE2 ST R0,SAVER0;  ST R1,SAVER1;  ST R2,SAVER2;  ST R3,SAVER3;  ST R4,SAVER4;  ST R5,SAVER5;  ST R6,SAVER6;  ST R7,SAVER7;  AND R1,R1,#0;  ADD R1,R1,#6; R1=i,R1=6,外层循环;  LEA R3,mat0x;  J2\_L1 AND R2,R2,#0; R2=j,R2=3,内层循环；  ADD R2,R2,#3;  J2\_L2 ADD R4,R3,#-1;  ADD R4,R4,R2; R4指向这一行的第三个第二个第一个  AND R5,R5,#0; R5中放有几个P1的棋；  LDR R0, R4, #0  BRz j2\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1 ;P1的子，R5加一  BRz J2ADD1  BRnzp j2\_ad1  J2ADD1 ADD R5, R5, #1  j2\_ad1 LDR R0, R4, #1 ;指向右边那个  BRz j2\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J2ADD2  BRnzp j2\_ad2  J2ADD2 ADD R5, R5, #1  j2\_ad2 LDR R0, R4, #1 ;指向右边那个  BRz j2\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J2ADD3  BRnzp j2\_ad3  J2ADD3 ADD R5, R5, #1  j2\_ad3 LDR R0,R4,#3  BRz j2\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J2ADD4  BRnzp j2\_ad4  J2ADD4 ADD R5, R5, #1  j2\_ad4 ADD R6, R5, #-4 ; 只要能走到这里必定为4个1或2  BRz pre\_P1WIN ; 如果p1计数器为4则p1赢  ADD R5, R5, #0  BRz pre\_P2WIN ; 如果p1计数器为0则p2赢  j2\_l2ed ADD R2, R2, #-1  BRp J2\_L2 ; 内循环到此结束    ADD R3, R3, #6 ; R3指向下一行  ADD R1, R1, #-1  BRp J2\_L1 ; 外循环到此结束  ; 判断-结束    LD R0, SAVER0 ; 恢复寄存器  LD R1, SAVER1  LD R2, SAVER2  LD R3, SAVER3  LD R4, SAVER4  LD R5, SAVER5  LD R6, SAVER6  LD R7, SAVER7  RET  ;-----------------------------------------  pre\_MAIN BRnzp preMAIN  PLAYX .FILL 0x58 ;X  PLAYnon .FILL 0x2D ;-  PLAYO .FILL 0x4F ;O  SPACE .FILL 0x20 ;空格  ENTER .FILL 0X0D ;回车  SAVER0 .FILL #0  SAVER1 .FILL #0  SAVER2 .FILL #0  SAVER3 .FILL #0  SAVER4 .FILL #0  SAVER5 .FILL #0  SAVER6 .FILL #0  SAVER7 .FILL #0  ;矩阵存放  mat0x .FILL #0  .FILL #0  .FILL #0  .FILL #0  .FILL #0  .FILL #0  mat1x .FILL #0  .FILL #0  .FILL #0  .FILL #0  .FILL #0  .FILL #0  mat2x .FILL #0  .FILL #0  .FILL #0  .FILL #0  .FILL #0  .FILL #0  mat3x .FILL #0  .FILL #0  .FILL #0  .FILL #0  .FILL #0  .FILL #0  mat4x .FILL #0  .FILL #0  .FILL #0  .FILL #0  .FILL #0  .FILL #0  mat50 .FILL #0  mat51 .FILL #0  mat52 .FILL #0  mat53 .FILL #0  mat54 .FILL #0  mat55 .FILL #0  preMAIN BRnzp MAIN;  pre\_P1WIN BRnzp P1WIN;  pre\_P2WIN BRnzp P2WIN;  ;子程序4判断斜着的、  CASE3 ST R0,SAVER0;  ST R1,SAVER1;  ST R2,SAVER2;  ST R3,SAVER3;  ST R4,SAVER4;  ST R5,SAVER5;  ST R6,SAVER6;  ST R7,SAVER7;  AND R1,R1,#0;  ADD R1,R1,#3; R1=i,R1=3,外层循环;  LEA R3,mat0x;  J3\_L1 AND R2,R2,#0; R2=j,R2=3,内层循环；  ADD R2,R2,#3;  J3\_L2 ADD R4,R3,#-1;  ADD R4,R4,R2; R4指向这一行的第三个，然后第二个，第一个  AND R5,R5,#0;  LDR R0, R4, #0  BRz j3\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J3ADD1  BRnzp j3\_ad1  J3ADD1 ADD R5, R5, #1  j3\_ad1 ADD R4, R4, #6  LDR R0, R4, #1 ; R4指向左上角  BRz j3\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J3ADD2  BRnzp j3\_ad2  J3ADD2 ADD R5, R5, #1  j3\_ad2 ADD R4, R4, #6  LDR R0, R4, #2 ; R4指向左上角  BRz j3\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J3ADD3  BRnzp j3\_ad3  J3ADD3 ADD R5, R5, #1  j3\_ad3 ADD R4, R4, #6  LDR R0, R4, #3 ; R4指向左上角  BRz j3\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J3ADD4  BRnzp j3\_ad4  J3ADD4 ADD R5, R5, #1  j3\_ad4 ADD R6, R5, #-4 ; 只要能走到这里必定为4个1或2  BRz pre\_P1WIN ; 如果p1计数器为4则p1赢  ADD R5, R5, #0  BRz pre\_P2WIN ; 如果p1计数器为0则p2赢  j3\_l2ed ADD R2, R2, #-1  BRp J3\_L2 ; 内循环到此结束    ADD R3, R3, #6 ; R3指向下一行  ADD R1, R1, #-1  BRp J3\_L1 ; 外循环到此结束  ; 判断\结束    LD R0, SAVER0 ; 恢复寄存器  LD R1, SAVER1  LD R2, SAVER2  LD R3, SAVER3  LD R4, SAVER4  LD R5, SAVER5  LD R6, SAVER6  LD R7, SAVER7  RET  ;-----------------------------------------------------  SAVER00 .FILL #0  SAVER11 .FILL #0  SAVER22 .FILL #0  SAVER33 .FILL #0  SAVER44 .FILL #0  SAVER55 .FILL #0  SAVER66 .FILL #0  SAVER77 .FILL #0  ;子程序5：判断斜着的/  CASE4 ST R0,SAVER0;  ST R1,SAVER1;  ST R2,SAVER2;  ST R3,SAVER3;  ST R4,SAVER4;  ST R5,SAVER5;  ST R6,SAVER6;  ST R7,SAVER7;    AND R1,R1,#0;  ADD R1,R1,#3; R1=i,R1=3,外层循环;  LEA R3,mat0x;    J4\_L1 AND R2,R2,#0; R2=j,R2=3,内层循环；  ADD R2,R2,#3;  J4\_L2 ADD R4,R3,#-1;  ADD R4,R4,R2; R4指向这一行的第三个，然后第二个，第一个  AND R5,R5,#0;    LDR R0, R4, #0  BRz j4\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J4ADD1  BRnzp j4\_ad1  J4ADD1 ADD R5, R5, #1  j4\_ad1 ADD R4, R4, #6 ; R4指向下一行  LDR R0, R4, #-1  BRz j4\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J4ADD2  BRnzp j4\_ad2  J4ADD2 ADD R5, R5, #1  j4\_ad2 ADD R4, R4, #6 ; R4指向下一行  LDR R0, R4, #-2  BRz j4\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J4ADD3  BRnzp j4\_ad3  J4ADD3 ADD R5, R5, #1  j4\_ad3 ADD R4, R4, #6 ; R4指向下一行  LDR R0, R4, #-3  BRz j4\_l2ed ; 如果遇到0直接break  ADD R0, R0, #-1  BRz J4ADD4  BRnzp j4\_ad4  J4ADD4 ADD R5, R5, #1  j4\_ad4 ADD R6, R5, #-4 ; 只要能走到这里必定为4个1或2  BRz pre\_P1WIN ; 如果p1计数器为4则p1赢  ADD R5, R5, #0  BRz pre\_P2WIN ; 如果p1计数器为0则p2赢  j4\_l2ed ADD R2, R2, #-1  BRp J4\_L2 ; 内循环到此结束    ADD R3, R3, #6 ; R3指向下一行  ADD R1, R1, #-1  BRp J4\_L1 ; 外循环到此结束  ; 判断/结束    LD R0, SAVER0 ; 恢复寄存器  LD R1, SAVER1  LD R2, SAVER2  LD R3, SAVER3  LD R4, SAVER4  LD R5, SAVER5  LD R6, SAVER6  LD R7, SAVER7  RET  ;------------------------------------------------------------  ;  ;指针存放  p0 .FILL #0  p1 .FILL #0  p2 .FILL #0  p3 .FILL #0  p4 .FILL #0  p5 .FILL #0  ;--------------------------------------------------  ;主程序  MAIN  AND R1,R1,#0;  ADD R1,R1,#15;  ADD R1,R1,#3;  JSR DISPLAY;  ;----------------让1下棋  LEA R0, mat50 ; 列指针赋值  ST R0, p0  LEA R0, mat51  ST R0, p1  LEA R0, mat52  ST R0, p2  LEA R0, mat53  ST R0, p3  LEA R0, mat54  ST R0, p4  LEA R0, mat55  ST R0, p5  ma\_lp1 ADD R0,R0,#0  AGAIN1 LEA R0, CALL1 ; 打印:请玩家2输入  TRAP x22  TRAP x20 ; 玩家2输入  TRAP x21  ADD R0, R0, #-16  ADD R0, R0, #-16  ADD R0, R0, #-16 ; 输入-48 ascii转数字    ADD R2, R0, #0 ; 输出回车  LD R0, HUICHE  TRAP x21  ADD R0, R2, #0    BRp i1t2 ; 判断输入是否>'0'  LEA R0, ERROR ; 输出错误提示  TRAP x22  BRnzp AGAIN1  i1t2 ADD R2, R0, #-7 ; R2=输入-'7'  BRn i1ed ; 判断输入是否<'7'  LEA R0, ERROR  TRAP x22  BRnzp AGAIN1    i1ed AND R2, R2, #0  ADD R2, R2, #1 ; 玩家1填数字1  LEA R3, p0 ; R3取列指针地址  ADD R3, R3, R0 ; R3列指针按照输入偏移  LDR R4, R3, #-1 ; 读取指针数据  STR R2, R4, #0 ; 将数据写入R3指向的地址  ADD R4, R4, #-6 ; 指针--  STR R4, R3, #-1 ; 将偏移后的值存回去    JSR DISPLAY  JSR CASE1 ; 判断 | 输赢  JSR CASE2 ; 判断 \ 输赢  JSR CASE3 ; 判断 / 输赢  JSR CASE4 ; 判断 - 输赢  AGAIN2 LEA R0, CALL2 ; 打印:请玩家2输入  TRAP x22  TRAP x20 ; 玩家2输入  TRAP x21  ADD R0, R0, #-16  ADD R0, R0, #-16  ADD R0, R0, #-16 ; 输入-48 ascii转数字    ADD R2, R0, #0 ; 输出回车  LD R0, HUICHE  TRAP x21  ADD R0, R2, #0    BRp i2t2 ; 判断输入是否>'0'  LEA R0, ERROR ; 输出错误提示  TRAP x22  BRnzp AGAIN2  i2t2 ADD R2, R0, #-7 ; R2=输入-'7'  BRn i2ed ; 判断输入是否<'7'  LEA R0, ERROR  TRAP x22  BRnzp AGAIN2    i2ed AND R2, R2, #0  ADD R2, R2, #-1 ; 玩家2填数字2  LEA R3, p0 ; R3取列指针地址  ADD R3, R3, R0 ; R3列指针按照输入偏移  LDR R4, R3, #-1 ; 读取指针数据  STR R2, R4, #0 ; 将数据写入R3指向的地址  ADD R4, R4, #-6 ; 指针--  STR R4, R3, #-1 ; 将偏移后的值存回去    JSR DISPLAY  JSR CASE1 ; 判断 | 输赢  JSR CASE2 ; 判断 \ 输赢  JSR CASE3 ; 判断 / 输赢  JSR CASE4 ; 判断 - 输赢  ADD R1, R1, #-1  BRp ma\_lp1 ; 循环到此结束    LEA R0,NOWIN;  TRAP X22  BR GOVER;  P1WIN LEA R0,WIN1;  TRAP X22;  BR GOVER;  P2WIN LEA R0,WIN2;  TRAP X22;  BR GOVER;  GOVER AND R0,R0,#0  HUICHE .FILL 0X0D ;回车  CALL1 .STRINGZ "Player 1, choose a column:"  CALL2 .STRINGZ "Player 2, choose a column:"  WIN1 .STRINGZ "Player 1 Wins.\n"  WIN2 .STRINGZ "Player 2 Wins.\n"  NOWIN .STRINGZ "Tie Game\n"  ERROR .STRINGZ "Invalid move. Try again.\n"  .END  以及测试的运行结果：  （1）输入不合法    （2）竖着获胜    （3）横着获胜    （4）\获胜    （5）平局 |
| 实验结论：   1. 本次实验难度很大，在做实验的第一天根本无从下手，后来听了老师的 讲解之后，明白了要先画出实验的流程图，根据流程图，将一个很复杂的程序分为若干个子程序，每块处理，更利于思考同时也是整个程序不那么冗杂。在编写各个子程序的过程中，也多多少少会有一些问题，思路上的问题去寻求他人的帮助以及在网上查询资料都可以解决，还有一些低级的错误打字错误等导致耗费许多时间，做了很多无用功。总之，在两周的努力学习之后，成功地做出了本次实验。 2. 通过本次实验，我更加熟悉的掌握了LC3汇编语言的用法。也明白了在编写程序的过程中要灵活运用标号和注释，这样可以使程序更加简便易读，出错时也方便检查。 |
| 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。