西安交通大学实验报告

课程名称： 算法设计与问题求解 实验名称： 综合算法设计

学 院： 机械工程学院 实验日期 2020 年 10 月 24 日

班 级： 机械97班 姓 名： 杨逢诜 学号：2193712613

#### 一、实验内容和结果

* **题目1**

1. 输入n（<10）,输出字母表前n个字母的全排列（每个位置字母可以重复）

提示：用递归实现

【源程序】

#include<stdio.h>

char str[10]; //声明全局变量，以防函数内的运算数值丢失

void Output(int position,int length) //声明函数：输出全部全排列序列

{

if(position==length) //若字符串已经被填充满，则输出字符串

{

printf("%s ",str);

}

else //若字符串未被填充满，则先填充第一位未被填充的字符空间

{

int i;

for(i=0;i<length;i++) //遍历所有可能的填充情形

{

str[position]='A'+i; //决定填充在该存储空间的字符

Output(position+1,length); //调用自身，确定下一步将用于继续确定（所有可能的）字符抑或输出结论

}

}

}

int main() //主函数开始

{

int n; //声明序列的长度并按照提示输入之

printf("Input n:");

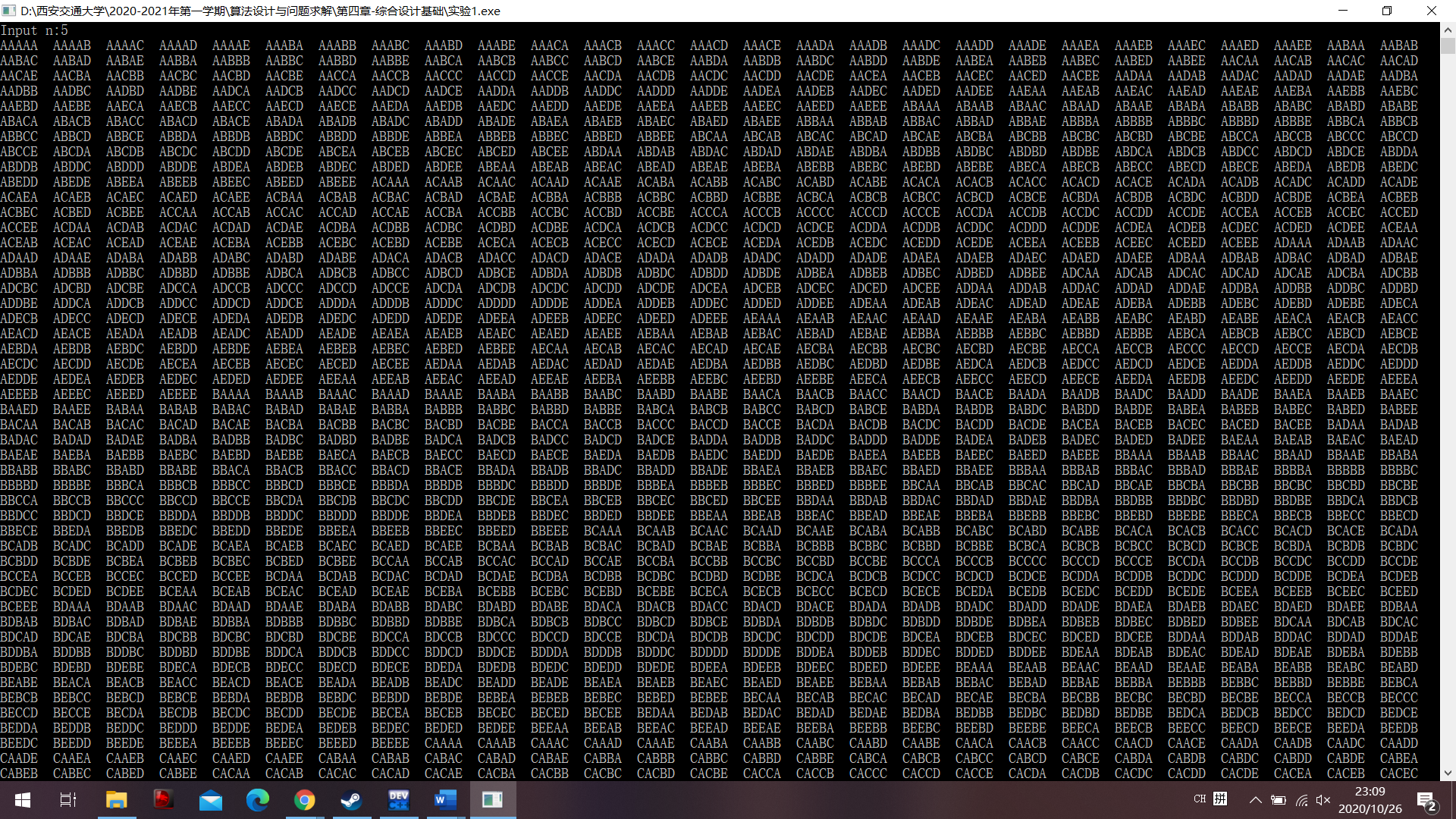
scanf("%d",&n);

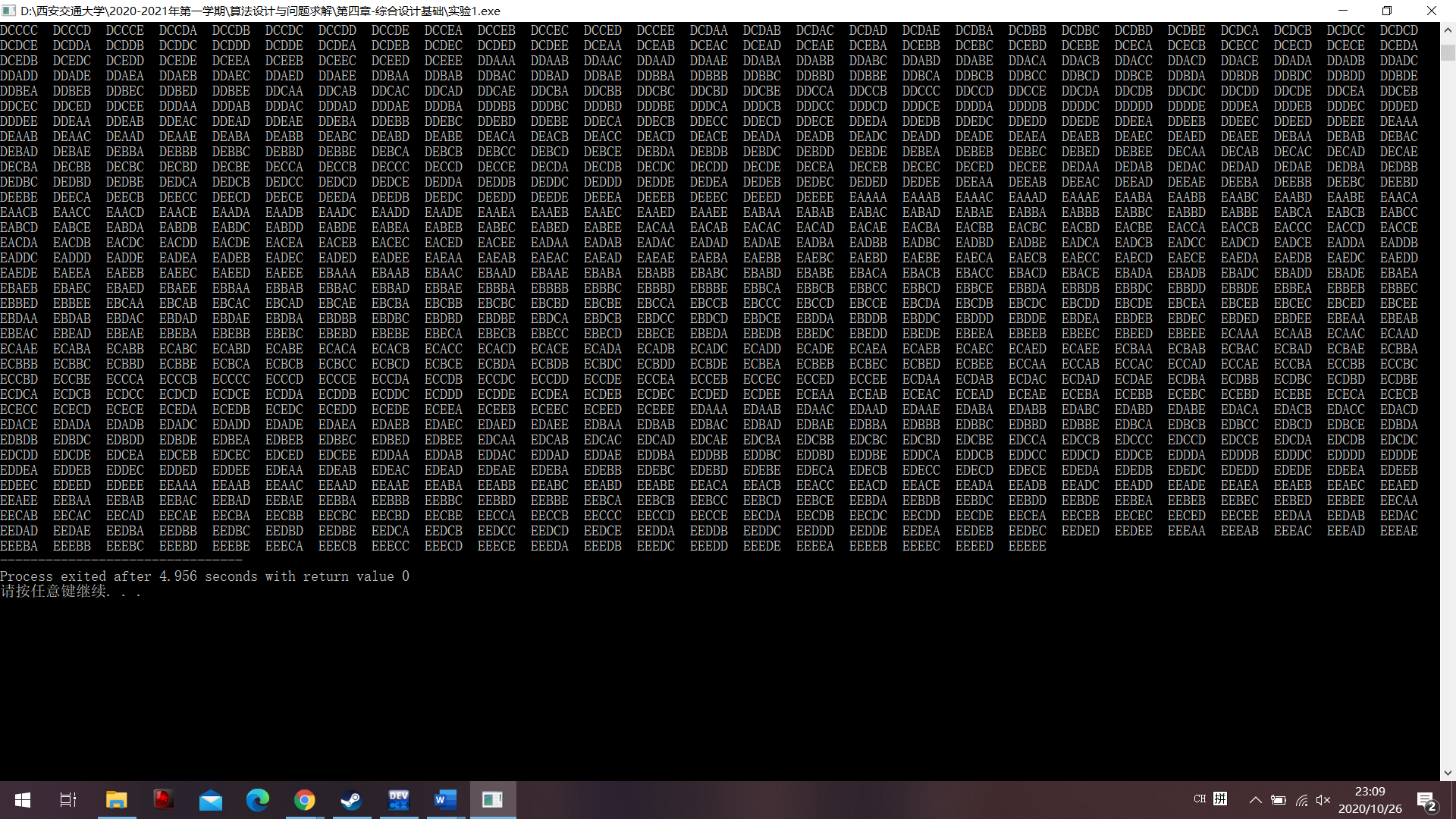
Output(0,n); //针对此长度输出字母的全排列序列，主函数结束

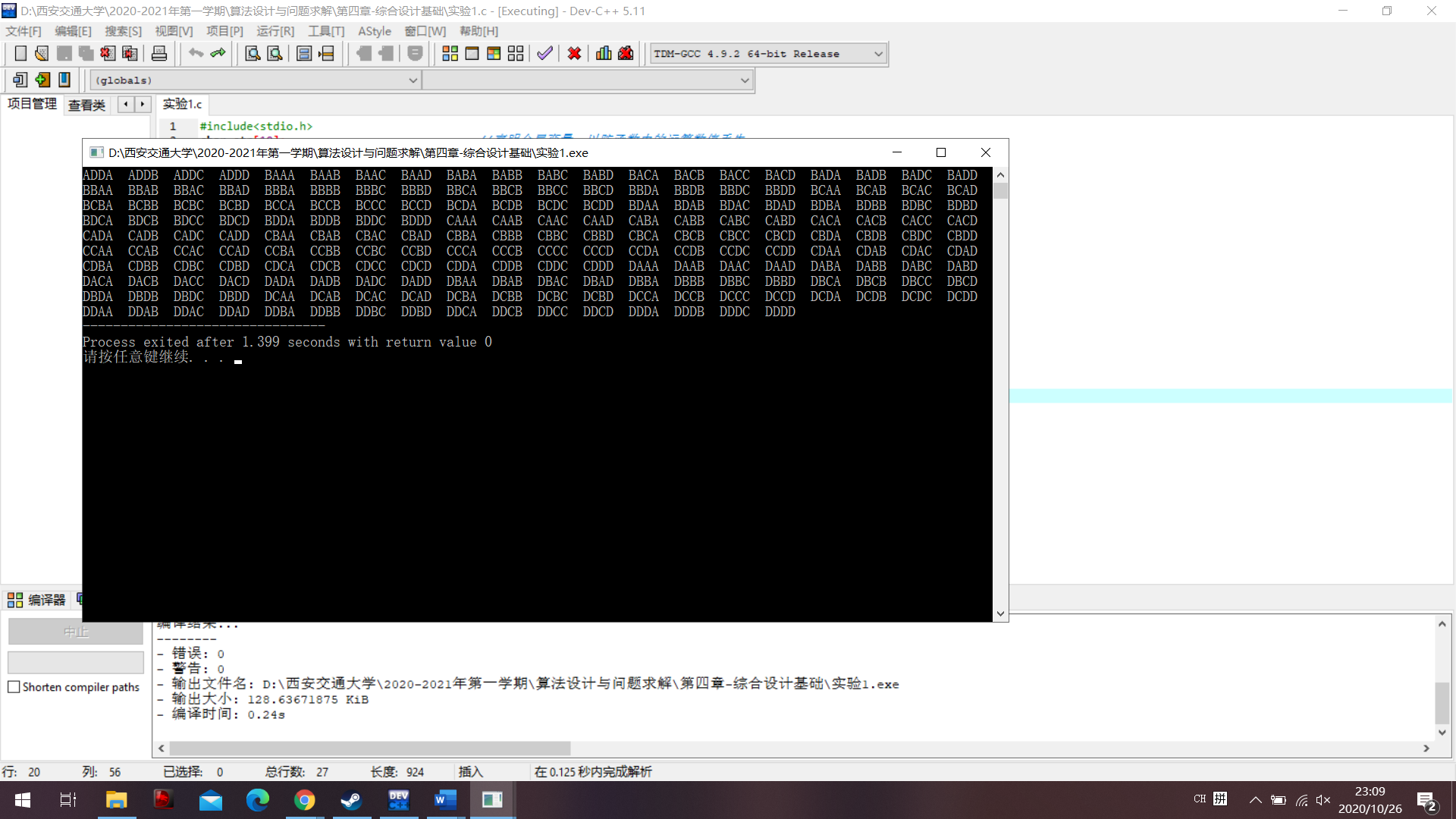
return 0;

}

【运行结果】







* **题目2**

2. 编写五子棋程序，在屏幕上用‘-’表示空位，用两个不同字符（比如A，B）表示白黑子。黑白双方轮流下棋，直到某方获胜为止。

【源程序】

#include<stdio.h>

#include<string.h>

//1指征胜利，0指征没有胜负之分

void PutInPiece(char player,int x,int y,char checkerboard[][20]) //声明函数：放置棋子

{

if(checkerboard[x][y]=='-') //若该位置为空，则将玩家的代表符存储到对应位置的存储单元

{

checkerboard[x][y]=player;

}

else printf("WRONG PLACE!"); //若该位置已经有其他玩家的标记，则终止放置进程并提示

}

int CheckingVictory(char player,char checkerboard[][20]) //声明函数：胜利判定

{

int i,j,k; //声明计数变量，前两个用于确定位置，后一个用于判定五子条件

for(i=1;i<=19;i++) //遍历：棋盘上的每个点；注意：由于遍历从左上角开始，所以判定时不必考虑五子向上延伸的情形

{

for(j=1;j<=19;j++)

{

if(checkerboard[i][j]==player) //找到该玩家放置的一个棋子后开始判定进程

{

if(checkerboard[i][j+1]==player&&j<=15) //判定是否能够在竖向连成五子-启动条件：在竖向有连续的两个棋子；最后一个约束条件用于防止调用过程中发生越界

{

for(k=2;k<=5;k++) //若有两个连续棋子，则查看下方是否有连续的五子，若无则直接终止进程，若有则直接判定胜利

{

if(checkerboard[i][j+k]!=player)

{

break;

}

}

if(k==5)

{

return 1;

}

}

if(checkerboard[i+1][j]==player&&i<=15) //判定是否能够在横向连成五子-启动条件：在横向有连续的两个棋子；最后一个约束条件用于防止调用过程中发生越界

{

for(k=2;k<=5;k++) //若有两个连续棋子，则查看右方是否有连续的五子，若无则直接终止进程，若有则直接判定胜利

{

if(checkerboard[i+k][j]!=player)

{

break;

}

}

if(k==5)

{

return 1;

}

}

if(checkerboard[i+1][j+1]==player&&i<=15&&j<=15) //判定是否能够在右斜向连成五子-启动条件：在右斜向有连续的两个棋子；最后一个约束条件用于防止调用过程中发生越界

{

for(k=2;k<=5;k++) //若有两个连续棋子，则查看右下方是否有连续的五子，若无则直接终止进程，若有则直接判定胜利

{

if(checkerboard[i+k][j+k]!=player)

{

break;

}

}

if(k==5)

{

return 1;

}

}

if(checkerboard[i-1][j+1]==player&&i>=5&&i<=15) //判定是否能够在右斜向连成五子-启动条件：在右斜向有连续的两个棋子；最后一个约束条件用于防止调用过程中发生越界

{

for(k=2;k<=5;k++) //若有两个连续棋子，则查看左下方是否有连续的五子，若无则直接终止进程，若有则直接判定胜利

{

if(checkerboard[i+k][j+k]!=player)

{

break;

}

}

if(k==5)

{

return 1;

}

}

}

}

}

return 0; //若遍历整个棋盘后没有找到满足成功条件的五子，则输出0以表示“无法判定”

}

void PrintBoard(char checkerboard[][20]) //声明函数：输出棋盘

{

int i,j; //声明计数变量用于控制循环

for(j=0;j<=19;j++) //将棋盘数组中的全部元素输出，注意换行

{

for(i=0;i<=19;i++)

{

printf("%c",checkerboard[i][j]);

}

printf("\n");

}

}

int main() //主函数开始

{

char checkerboard[20][20]; //声明棋盘数组，声明玩家标识符

char black='A',white='B';

int i,j,x,y; //声明整型变量，前两个用于控制循环和判定胜负，后两个用于暂时存放棋子位置

checkerboard[0][0]=' '; //构造棋盘原点

for(i=1;i<=19;i++) //构造棋盘横轴

{

checkerboard[i][0]='0'+(i-1)%10;

}

for(i=1;i<=19;i++) //构造棋盘纵轴

{

checkerboard[0][i]='0'+(i-1)%10;

}

for(i=1;i<=19;i++) //用空标识符“-”填充棋盘上的其他空间

{

for(j=1;j<=19;j++)

{

checkerboard[i][j]='-';

}

}

PrintBoard(checkerboard); //输出棋盘，游戏开始

do{

printf("Black:"); //黑方先走，根据提示输入要下的棋子的位置

scanf("%d %d",&x,&y);

PutInPiece(black,x,y,checkerboard); //将棋子放入棋盘数组中，并判定是否满足胜利条件；若满足则直接终止循环

PrintBoard(checkerboard);

i=CheckingVictory(black,checkerboard);

if(i==1)

{

break;

}

printf("White:"); //白方后走，根据提示输入要下的棋子的位置

scanf("%d %d",&x,&y);

PutInPiece(white,x,y,checkerboard); //将棋子放入棋盘数组中，并判定是否满足胜利条件

PrintBoard(checkerboard);

j=CheckingVictory(white,checkerboard);

}while(i==0&&j==0); //若白方满足胜利条件则进程终止

if(i==1) //输出胜负信息，主函数（游戏）结束

{

printf("Black Win!!");

}

else printf("White Win!!");

}

【运行结果】

