#include <stdlib.h>

#include <time.h>

#include <stdio.h>

#include<conio.h>

#include <Windows.h>

#pragma comment(lib,"winmm.lib")

#define ROW 4

#define COL 4

int arr[ROW][COL] = { 0 };

int score = 0;

int k = 0;

//成绩记录

void reminder(int a)

{

time\_t time\_seconds = time(0);

//定义用于保存时间的变量；time()函数用于返回从1970年1月1日00：00：00至今经过的秒数

struct tm timer;//<time.h>头文件中定义的结构体

localtime\_s(&timer, &time\_seconds);//获取系统时间

FILE\* fp;//定义文件指针

errno\_t err;

//定义整形数据，用以返回错误代码（如果成功就返回0，失败则返回相应的非零错误代码）

err = fopen\_s(&fp, "records.txt", "a");

//用追加模式创建或打开文件

if (err)

{

printf("在记录数据的时候打开失败！\n");

exit(0);

}

else

{

fprintf\_s(fp, "%-8d%.2d.%.2d.%.2d:%.2d:%.2d\n",

a,

timer.tm\_year + 1900,

timer.tm\_mon + 1,

timer.tm\_mday,

timer.tm\_hour,

timer.tm\_min,

timer.tm\_sec);

fclose(fp);

}//向创建成功的文件中写入分数及游戏完成时间（若创建或打开文件失败，则打印出相应提示并退出）

}

//打印最高记录

int best(void)

{

char m = '0';

int a = 0, b = 0, max = 0, i = 0;

FILE\* fp;

errno\_t err;

err = fopen\_s(&fp, "records.txt", "a");

fclose(fp);//以追加模式创建或打开文件

err = fopen\_s(&fp, "records.txt", "r");

//用只读模式打开已存在的文件（若打开文件失败，则打印出相应提示并退出）

if (err)

{

printf("在打印历史记录的时候打开失败！\n");

exit(0);

}

else

{

fseek(fp, 0, 2);//重置文件位置指示至文件尾

k = ftell(fp) / 24;

//读取流变量k的当前位置号（已位于文件尾），并除以24以计算游戏完成次数

printf("您共有%d条游戏记录\n", k);

fseek(fp, 0, 0);//重置文件位置指示至文件头

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

{

fseek(fp, i \* 24, 0);//重置文件位置指示

fscanf\_s(fp, "%d", &a);//读取游戏分数

if (max < a)

max = a;

while (m != '\n')

m = fgetc(fp);

m++;

//将文件中记录时间的数据赋值给m，用于移动文件指示位置

fscanf\_s(fp, "%d", &b);

if (max < b)

max = b;

}//逐条读取游戏分数并比较大小

}

fclose(fp);

return max;//返回最高纪录

}

//设置光标的位置

void SetCurPos(int x, int y)

{

HANDLE hStdOutput = GetStdHandle(STD\_OUTPUT\_HANDLE);//调用句柄

COORD coord = { x, y };

SetConsoleCursorPosition(hStdOutput, coord);//设置光标位置

}

//隐藏光标

void ViewInit(int width, int height)

{

HANDLE hStdOutput = GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_CURSOR\_INFO info;//声明一个结构体

GetConsoleCursorInfo(hStdOutput, &info);//检索光标

info.bVisible = 0;//使光标不可见

SetConsoleCursorInfo(hStdOutput, &info);//将修改值返回控制台

}

//打印格子

void GamePrint(int row, int col)

{

SetCurPos(5, 9);

printf("SCORE = %d", score);//设置光标位置并打印分数

SetCurPos(5, 10);

printf(" Highest=%d", best());

printf("\n您可以在您电脑中名为“records”的文件中\n所有的游戏记录\n");

//设置光标位置并打印最高纪录、查看文件提示

SetCurPos(48, 5);

int i = 0;

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

{

for (int k = 0; k < 4; k++)

{

printf(" ");

for (int j = 0; j < 4; j++)

{

printf("-");

}

}

printf("\n\t\t\t\t\t\t");//用于移动光标至指定位置

for (int k = 0; k < 4; k++)

{

printf("|");

if (arr[i][k])

{

printf("%4d", arr[i][k]);//将数字填入表格

}

else

{

for (int j = 0; j < 4; j++)

{

printf(" ");//没有数字的位置用空格替代

}

}

}

printf("|\n\t\t\t\t\t\t");//用于移动光标至指定位置

if (i == row - 1)

{

for (int k = 0; k < 4; k++)

{

printf(" ");

for (int j = 0; j < 4; j++)

{

printf("-");

}

}//打印不在“— |”循环范围内的第9行

printf("\n\t\t\t\t\t\t");//用于移动光标至指定位置

}

}//使用循环语句打印表格并将游戏数字置入表格空位

SetCurPos(48, 4);

printf("上下左右表示方向\n");

//设置光标位置以打印游戏操作要求

SetCurPos(48, 15);//设置光标位置

}

//生成随机数

void RandNum(int row, int col) //生成随机数

{

int x = 0;

int y = 0;

do

{

x = rand() % 4;

y = rand() % 4;

} while (arr[x][y]);

arr[x][y] = (rand() % 10 ? 2 : 4);

}

//向左移

int MoveLeft(int arr[ROW][COL], int\* a)

{

int m = 0;

for (int i = 0; i < ROW; i++)

{

int j = 1;

int k = 0;

for (; j < 4; j++)

{

if (arr[i][j] > 0)

{

if (arr[i][k] == 0)

{

arr[i][k] = arr[i][j];

arr[i][j] = 0;

m = 1;

}

else if (arr[i][k] == arr[i][j])

{

arr[i][k] = 2 \* arr[i][k];

score += arr[i][k];

arr[i][j] = 0;

k++;

m = 1;

\*a = 1;

}

else

{

arr[i][++k] = arr[i][j];

if (j != k)

{

arr[i][j] = 0;

m = 1;

}

}

}

}

}

return m;

}

//向右移

int MoveRight(int arr[ROW][COL], int\* a)

{

int m = 0;

for (int i = 0; i < ROW; i++)

{

int j = 2;

int k = 3;

for (; j >= 0; j--)

{

if (arr[i][j] > 0)

{

if (arr[i][k] == 0)

{

arr[i][k] = arr[i][j];

arr[i][j] = 0;

m = 1;

}

else if (arr[i][k] == arr[i][j])

{

arr[i][k] = 2 \* arr[i][k];

score += arr[i][k];

arr[i][j] = 0;

k--;

m = 1;

\*a = 1;

}

else

{

arr[i][--k] = arr[i][j];

if (j != k)

{

arr[i][j] = 0;

m = 1;

}

}

}

}

}

return m;

}

//向上移

int MoveUp(int arr[ROW][COL], int\* a)

{

int m = 0;

for (int j = 0; j < COL; j++)

{

int i = 1;

int k = 0;

for (; i < 4; i++)

{

if (arr[i][j] > 0)

{

if (arr[k][j] == arr[i][j])

{

score += arr[k++][j] \*= 2;

arr[i][j] = 0;

m = 1;

\*a = 1;

}

else if (arr[k][j] == 0)

{

arr[k][j] = arr[i][j];

arr[i][j] = 0;

m = 1;

}

else

{

arr[++k][j] = arr[i][j];

if (i != k)

{

arr[i][j] = 0;

m = 1;

}

}

}

}

}

return m;

}

//向下移

int MoveDown(int arr[ROW][COL], int\* a)

{

int m = 0;

for (int j = 0; j < COL; j++)

{

int i = 2;

int k = 3;

for (; i >= 0; i--)

{

if (arr[i][j] > 0)

{

if (arr[k][j] == 0)

{

arr[k][j] = arr[i][j];

arr[i][j] = 0;

m = 1;

}

else if (arr[k][j] == arr[i][j])

{

arr[k][j] = 2 \* arr[k][j];

score += arr[k][j];

arr[i][j] = 0;

k--;

m = 1;

\*a = 1;

}

else

{

arr[--k][j] = arr[i][j];

if (i != k)

{

arr[i][j] = 0;

m = 1;

}

}

}

}

}

return m;

}

//判断是否获胜

int IfWin(int row, int col)

{

int b = 0;

int copy\_board[ROW][COL], i, j;

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

{

for (j = 0; j < COL; j++)

{

copy\_board[i][j] = arr[i][j];

}

}

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

for (j = 0; j < col; j++)

if (arr[i][j] == 2048)

return 1;//获胜

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

for (j = 0; j < col; j++)

if (MoveLeft(copy\_board, &b) == 1 || MoveRight(copy\_board, &b) == 1 || MoveUp(copy\_board, &b) == 1 || MoveDown(copy\_board, &b) == 1)

return 0;//判断是否能移动

return -1;//失败

}

//游戏继续

int GameContinue()

{

int ch1 = 0;

int ch2 = 0;

int h = 0, b = 0;

int (\*f[4])(int arr[4][4], int\* a) = { MoveUp,MoveDown,MoveLeft,MoveRight };

while (1)

{

if (ch1 = \_getch())

{

ch2 = \_getch();

if (ch2 == 72)

{

h = (\*f[0])(arr, &b);

break;

}

if (ch2 == 80)

{

h = (\*f[1])(arr, &b);

break;

}

if (ch2 == 75)

{

h = (\*f[2])(arr, &b);

break;

}

if (ch2 == 77)

{

h = (\*f[3])(arr, &b);

break;

}

}

}

if (h == 0)

PlaySound("D:\\蜂鸣.wav", NULL, SND\_FILENAME | SND\_ASYNC);

else if (h == 1 && b == 0)

PlaySound("D:\\旋转.wav", NULL, SND\_FILENAME | SND\_ASYNC);

else

PlaySound("D:\\高级.wav", NULL, SND\_FILENAME | SND\_ASYNC);

return h;

}

void game()

{

srand((unsigned int)time(NULL));

ViewInit(20, 10);

RandNum(ROW, COL);

RandNum(ROW, COL);

GamePrint(ROW, COL);

while (1)

{

if (IfWin(ROW, COL) == 1)

{

printf("YOU WIN!\n");

PlaySound("D:\\胜利.wav", NULL, SND\_FILENAME | SND\_ASYNC);

reminder(score);

return;

}

else if (IfWin(ROW, COL) == -1)

{

printf("GAME OVER\n");

PlaySound("D:\\失败.wav", NULL, SND\_FILENAME | SND\_ASYNC);

reminder(score);

return;

}

else

{

if (GameContinue())

{

system("cls");

RandNum(ROW, COL);

GamePrint(ROW, COL);

}

else continue;

}

}

}

void game2()

{

ViewInit(20, 10);

GamePrint(ROW, COL);

while (1)

{

if (IfWin(ROW, COL) == 1)

{

printf("YOU WIN!\n");

PlaySound("D:\\胜利.wav", NULL, SND\_FILENAME | SND\_ASYNC);

reminder(score);

return;

}

else if (IfWin(ROW, COL) == -1)

{

printf("GAME OVER\n");

PlaySound("D:\\失败.wav", NULL, SND\_FILENAME | SND\_ASYNC);

reminder(score);

return;

}

else

{

if (GameContinue())

{

system("cls");

RandNum(ROW, COL);

GamePrint(ROW, COL);

}

else continue;

}

}

}

void GAME()

{

char c = '1', a;

int m, i;

while (1)

{

while (c == '1')

{

for (i = 0; i < 4; i++)

for (m = 0; m < 4; m++)

arr[m][i] = 0;

score = 0;

system("cls");

game();

if (score < 2000)

{

printf("请输入1以继续，输入b退出：\n");

do {

c = \_getch();

} while (!((c == '1') || (c == 'b')));

if (c == 'b')

exit(0);

}

else

{

printf("您是否选择用2000分换取复活（消除第一行数字并在左下角获得福利）\n若您选择复活，请回复2;\n若重新开始游戏，请回复1：\n（小声告诉你，回复3有惊喜）\n");

do {

c = \_getch();

} while (!(c == '2' || c == '3' || c == '1'));

}

continue;

}

while (c == '2')

{

arr[0][0] = 0;

arr[0][1] = 0;

arr[0][2] = 0;

arr[0][3] = 0;

arr[3][0] = 1024;

score -= 2000;

game2();

if (score < 2000)

{

printf("请输入1以继续，输入b以退出游戏：\n");

do {

c = \_getch();

} while (!((c == '1') || (c == 'b')));

if (c == 'b')

exit(0);

}

else

{

printf("您是否选择分数减2000，从而消除第一行数字并在左下角获得福利,得到复活？\n若您选择复活，请回复2;\n若重新开始游戏，请回复1，输入b以退出游戏：\n（小声告诉你，回复3有惊喜）\n");

c = \_getch();

}

continue;

}

while (c == '3')

{

printf("这里有一个神奇的的内部通道，输入邀请码，感受不一样的2048。\n↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓\n＋客服小姐姐QQ：876918519在线索要！！！\n\n要是你懒得加请回复0，问到了就输入1（输入b以退出游戏）：\n");

do

{

a = \_getch();

} while (!(a == '1' || a == 'b' || a == '0'));

if (a == 'b') exit(0);

if (a == '0')

{

printf("请输入1以继续，输入b以退出游戏：\n");

do

{

c = \_getch();

} while (!(c == '1' || c == 'b'));

if (c == 'b') exit(0);

}

else if (a == '1')

{

char code[50];

printf("\n好样的，我猜你要到了邀请码，那么接下来请输入邀请码：");

scanf\_s("%s", code);

if (strcmp(code, "iama2048genius") == 0)

{

int i, j;

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

{

for (j = 0; j < COL; j++)

arr[i][j] = 128;

}

score = 100000;

game2();

c = '4';

printf("请输入1以继续，输入b以退出游戏：\n");

do {

c = \_getch();

} while (!((c == '1') || (c == 'b')));

if (c == 'b')

exit(0);

}

else

{

printf("别瞎猜了你猜不对的，请老老实实找小姐姐要邀请码\n(要是不好意思也可以找我们的客服小哥QQ：2830904724要)\n\n再给你一次重头选择的机会\n\n");

printf("是否选择清空分数，从而消除第一行数字并在左下角获得福利,并得到复活？（复读）\n若您选择复活，请回复2;\n若重新开始游戏，请回复1(输入b以退出游戏）：\n3再考虑一下？？？真的不会要钱\n");

do {

c = \_getch();

} while (!(c == '2' || c == '1' || c == 'b' || c == '3'));

if (c == 'b')

exit(0);

}

}

continue;

}

}

printf("\n\t\t您已退出游戏。\n");

}

int main()

{

system("color E8");

GAME();

system("pause");

return 0;

}