#include <iostream>

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#include<conio.h>

int row;//行

int col;//列

int num[4][4];//初始化数组

int a=0,b=0,c=0,d=0;//初始化两个2

int e=0,f=0;//随机生成格子

int x[2]={2,4};//随机生成2或4

int i,j;//排序

int k=0;//计算移动，合并的次数

void init(){//初始化

for(row=0; row<4; row++){

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

num[row][col]=0;

}

while(a==c&&b==d){

srand((unsigned)time(NULL));

a = rand()%4;

b = rand()%4;

c = rand()%4;

d = rand()%4;

if(a!=c||b!=d)

break;

}

num[a][b]=2;

num[c][d]=2;

a=0;

b=0;

c=0;

d=0;

}

void print(){//输出

printf("\n");

for(row=0; row<4; row++){

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

printf("+----");

printf("+\n");

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

if (num[row][col] == 0)

printf("| ");

else

printf("|%4d", num[row][col]);

printf("|\n");

}

printf("+----+----+----+----+\n");

}

void generate(){//生成一个新的数

srand((unsigned)time(NULL));

e = rand()%4;

f = rand()%4;

while(num[e][f]!=0){

e = rand()%4;

f = rand()%4;

if(num[e][f]==0) break;

}

num[e][f]=x[rand()%2];

print();

}

void end(){//判断失败条件

int t=0;

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

for(col=0;col<3;col++)

if(num[row][col]!=num[row][col+1]&&num[row][col]!=0&&num[row][col+1]!=0)

t++;

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

for (row=0;row<3;row++)

if(num[row][col]!=num[row+1][col]&&num[row][col]!=0&&num[row+1][col]!=0)

t++;

if(t==24){

print();

printf("很遗憾，你失败了。");

exit(0);

}

}

int main(){

system("cls");//开始界面

printf("###############Welcome to 2048###############\n");

printf("# Please choose one command: #\n");

printf("# h: help #\n");

printf("# s: start #\n");

printf("# r: restart #\n");

printf("# q: quit #\n");

printf("##################Have fun###################\n");

while(1){

char ch=getch();

if(ch=='h'){//帮助界面

printf("\n");

printf("\*\*\*\*\*\*Help Information\*\*\*\*\*\*\n");

printf(" h : help info\n");

printf(" s : start game\n");

printf(" w/↑ : merge up\n");

printf(" s/↓ : merge down\n");

printf(" a/← : merge left\n");

printf(" d/→ : merge right\n");

printf(" r : restart\n");

printf(" q : quit\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

continue;

}

if(ch=='s'){//开始游戏

system("cls");

init();

print();

while(1){

ch=getch();

if(ch=='r'){//重新开始游戏

printf("\n是否重新开始游戏，请输入【y/n】");

while(1){

char m=getch();

if(m=='y'){

printf("\n重新开始游戏\n");

return main();}

if(m=='n'){

printf("\n请继续游戏\n");

break;

}

if(m!='y'&&m!='n'){

printf("无效指令\n");

continue;

}

}

}

if(ch=='q'){//退出游戏

printf("\n是否退出游戏，请输入【y/n】");

while(1){

char m=getch();

if(m=='y'){

printf("\n退出游戏\n");

exit(0);

}

if(m=='n'){

printf("\n请继续游戏\n");

break;

}

if(m!='y'&&m!='n'){

printf("无效指令\n");

continue;

}

}

}

if (ch=='a'){//左移

system("cls");

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

for(i=0;i<3;i++){

for(col=0;col<3-i;col++){

if(num[row][col]==0&&num[row][col+1]!=0){

num[row][col]=num[row][col+1];

num[row][col+1]=0;

k++;

}

}

}

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

for(col=0;col<3;col++){

if(num[row][col]==num[row][col+1]&&num[row][col]!=0){

num[row][col]=num[row][col]+num[row][col+1];

k++;

for(j=col+1;j<3;j++)

num[row][j]=num[row][j+1];

num[row][3]=0;

}

}

if(k==0){

end();

print();

}

else{

generate();

k=0;

}

}

if(ch=='d'){//右移

system("cls");

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

for(i=0;i<3;i++){

for(col=3;col>i;col--){

if(num[row][col]==0&&num[row][col-1]!=0){

num[row][col]=num[row][col-1];

num[row][col-1]=0;

k++;

}

}

}

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

for(col=3;col>0;col--){

if(num[row][col]==num[row][col-1]&&num[row][col]!=0){

num[row][col]=num[row][col]+num[row][col-1];

k++;

for(j=col-1;j>0;j--)

num[row][j]=num[row][j-1];

num[row][0]=0;

}

}

if(k==0){

end();

print();

}

else{

generate();

k=0;

}

}

if(ch=='w'){//上移

system("cls");

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

for(i=0;i<3;i++){

for(row=0;row<3-i;row++){

if(num[row][col]==0&&num[row+1][col]!=0){

num[row][col]=num[row+1][col];

num[row+1][col]=0;

k++;

}

}

}

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

for(row=0;row<3;row++){

if(num[row][col]==num[row+1][col]&&num[row][col]!=0){

num[row][col]=num[row][col]+num[row+1][col];

k++;

for(j=row+1;j<3;j++)

num[j][col]=num[j+1][col];

num[3][col]=0;

}

}

if(k==0){

end();

print();

}

else{

generate();

k=0;

}

}

if(ch=='s'){//下移

system("cls");

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

for(i=0;i<3;i++){

for(row=3;row>i;row--){

if(num[row][col]==0&&num[row-1][col]!=0){

num[row][col]=num[row-1][col];

num[row-1][col]=0;

k++;

}

}

}

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

for(row=3;row>0;row--){

if(num[row][col]==num[row-1][col]&&num[row][col]!=0){

num[row][col]=num[row][col]+num[row-1][col];

k++;

for(j=row-1;j>0;j--)

num[j][col]=num[j-1][col];

num[0][col]=0;

}

}

if(k==0){

end();

print();

}

else{

generate();

k=0;

}

}

if(ch!='w'&&ch!='s'&&ch!='a'&&ch!='d'&&ch!='q'&&ch!='r')//判断错误输入

printf("无效指令\n");

for(row=0; row<4; row++)//判断通关条件

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

if(num[row][col]==2048){

printf("恭喜你，通关了！");

exit(0);

}

}

}

if(ch=='q'){//退出游戏

printf("\n是否退出游戏，请输入【y/n】");

while(1){

ch=getch();

if(ch=='y'){

printf("\n退出游戏\n");

exit(0);

}

if(ch=='n'){

printf("\n请继续游戏\n");

break;

}

if(ch!='y'&&ch!='n'){

printf("无效指令\n");

continue;

}

}

}

if(ch!='h'&&ch!='s'&&ch!='q'){//判断错误输入

printf("无效指令\n");

continue;

}

}

}