//数独游戏c++

class CSudoku

{

int map[9][9];

int blanks;

int smod;

int solves;

int check(int,int,int\*);

void dfs();

public:

enum{ANY=0,ALL=1};

CSudoku(int);

CSudoku::CSudoku(int \*data);

void SudokuGenerator(int); //随机生成数独,n越大越难

void SudokuGenerator(int \*data);//人工指定数独

//virtual ~CSudoku();

void display();//显示数独

int resolve(int mod=ALL);//解数独

void analyze();

};

#include "stdio.h"

#include "stdlib.h"

#include "time.h"

#include "iostream"

#include "iomanip" //要用到格式控制符

using namespace std;

CSudoku::CSudoku(int n){

int j;

j=rand()%3;

blanks=n+j;

SudokuGenerator(blanks);

cout<<endl<<"随机数独: --->(Y轴)[x,y坐标均从0开始]"<<endl;

cout<<endl<<"填数请按Enter键,按指示操作"<<endl;

//cout<<(空格子数为"<<blanks<<")"<<endl;

display();

cout<<"press enter to continue! "<<endl;

getchar();

getchar();

while(1){

cout<<"%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

cout<<"%%请选择您的操作:%% "<<endl;

cout<<"%%============================%% "<<endl;

cout<<"%%%% "<<endl;

cout<<"%%1.显示当前数独%% "<<endl;

cout<<"%%2.分析求解%% "<<endl;

cout<<"%%3. 查看结果%% "<<endl;

cout<<"%%4.返回%% "<<endl;

cout<<"%%%% "<<endl;

cout<<"%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

int select;

cin>>select;

switch (select){

case 1:{

cout<<endl<<"当前随机数独: "<<endl;

display();

cout<<"press enter to continue! "<<endl;

getchar();

getchar();

break;

}

case 2:{

analyze();

break;

}

case 3:{

cout<<endl<<"显示结果: "<<endl;

resolve();

cout<<"press enter to continue! "<<endl<<endl;

getchar();

getchar();

break;

}

case 4: return;

default:{

cout<<"输入错误,请重新输入."<<endl;

break;

}

}

}

}

CSudoku::CSudoku(int \*data){

SudokuGenerator(data);

cout<<endl<<"已知数独为: "<<endl;

display();

cout<<"press enter to continue! "<<endl;

getchar();

getchar();

analyze();

}

void CSudoku::SudokuGenerator(int n){

int i,j;

int mark[10];

srand(time(0));

//每一行i产生一个随机位置（列j）并置为当前行值i+1,0=<i,j<=8。

do{

for(i=0;i<9;++i){

for(j=0;j<9;++j)

map[i][j]=0;

j=rand()%9;

map[i][j]=i+1;

}

//display();

}

while(!resolve(ANY));//生成完整的随机Sudoku表格

//挖窟窿

for(int k=0;k<n;++k){

int tmp,flag=0,sum=1;

do{

//cout<<"sum="<<sum<<endl;

if (sum++>81){

SudokuGenerator(n);

return;

}

if (flag==1)

map[i][j]=tmp;

do{

i=rand()%81;

j=i%9;

i=i/9;

}

while (map[i][j]==0);

tmp=map[i][j];

map[i][j]=0;

flag=1;

}

while(check(i,j,mark)>1);

}

}

void CSudoku::SudokuGenerator(int \*data){

int \*pm=(int\*)map;

for(int i=0;i<81;++i)

pm[i]=data[i];

}

void CSudoku::display(){

int count=0;

printf("┏━┯━┯━┳━┯━┯━┳━┯━┯━┓\n");//最开始的一行

for(int i=0;i<9;++i){

for(int j=0;j<9;++j){

if(j%3==0){

printf("┃");//先输出加粗制表符

printf(" ");

if(map[i][j]>0){

printf("%d",map[i][j]);

}

else{

printf(" ");

}

//printf(" ");//把光标移到下一个格子对应的起始位置

//printf(" ");

}

else{

printf("│");//先输出正常制表符

printf(" ");

if(map[i][j]>0){

printf("%d",map[i][j]);

}

else{

printf(" ");

}

//printf(" ");//把光标移到下一个格子对应的起始位置

//printf(" ");

}

if(j==8){

printf("┃");//一行中最后一个粗黑色制表符

}

}

//一行输出完成

if(i!=8){

if((i+1)%3==0){

printf("\n");

printf("┣━┿━┿━╋━┿━┿━╋━┿━┿━┫\n");

}

else{

printf("\n");

printf("┠─┼─┼─╂─┼─┼─╂─┼─┼─┨\n");

}

}

if(i==8){

printf("\n");

printf("┗━┷━┷━┻━┷━┷━┻━┷━┷━┛\n"); //最后一行

}

}

}

int CSudoku::resolve(int mod){

smod=mod;

if(mod==ALL){

solves=0;

dfs();

return solves;

}

else if(mod==ANY){

try

{

dfs();

return 0;

}

catch(int){

return 1;

}

}

return 0;

}

int CSudoku::check(int y,int x,int \*mark){

int i,j,is,js,count=0;

for(i=1;i<=9;++i)

mark[i]=0;

for(i=0;i<9;++i)

mark[map[y][i]]=1;

for(i=0;i<9;++i)

mark[map[i][x]]=1;

is=y/3\*3;

js=x/3\*3;

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

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

mark[map[is+i][js+j]]=1;

}

for(i=1;i<=9;++i)

if(mark[i]==0)

count++;

return count;

}

void CSudoku::dfs(){

int i,j,im=-1,jm,min=10;

int mark[10];

// display();

//求自由度最小的格map[im][jm]

for(i=0;i<9;++i){

for(j=0;j<9;++j){

if(map[i][j])

//如果此格已填入数则看一下格。

continue;

int c=check(i,j,mark); //如果此格空，则求其自由度。

if(c==0)//已到结尾，没有空格了。

return;

if(c<min){

im=i;

jm=j;

min=c;

}

}

}

if(im==-1)//若im=-1，则格子都填满。

{

if(smod==ALL)//smod==ALL是求解过程。

{

//printf("No. %d:\n",++solves);

display();

return;

}

else if(smod==ANY) //smod==ANY是初始化过程。

{

throw(1);

}

}

check(im,jm,mark);

for(i=1;i<=9;++i){

if(mark[i]==0)

{

map[im][jm]=i;//从小到大让第一个可填的数填入自由度最小的格。

dfs();

}

}

map[im][jm]=0;

return;

}

void CSudoku::analyze(){

while(1){

cout<<"%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

cout<<"%%请问你想做什么?%% "<<endl;

cout<<"%%============================%% "<<endl;

cout<<"%%%% "<<endl;

cout<<"%%1.查找最小不确定度的格子%% "<<endl;

cout<<"%%2.指定格子的可填数%% "<<endl;

cout<<"%%3.给指定格子填数%% "<<endl;

cout<<"%%4.显示当前数独%% "<<endl;

cout<<"%%5.查看结果%% "<<endl;

cout<<"%%6.返回%% "<<endl;

cout<<"%%%% "<<endl;

cout<<"%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

int select;

cin>>select;

switch(select){

case 1://求不确定度最小的空格map[im][jm]

{

int i,j,im=-1,jm,min=10;

int mark[10];

for(i=0;i<9;++i)

{

for(j=0;j<9;++j)

{

if(map[i][j]) //如果此格已填入数则看一下格。

continue;

int c=check(i,j,mark); //如果此格空，则求其不确定度。

if(c==0)

{

cout<<"不能得到结果或已无空格子!自动返回! "<<endl;

return;

}

if(c<min)

{

im=i;

jm=j;

min=c;

}

}

}

cout<<endl<<"不确定度最小的格子为:["<<im<<","<<jm<<"]"<<"其可填的数的个数为:"<<min<<endl<<endl;

cout<<"press enter to continue! "<<endl;

getchar();

getchar();

break;

}

case 2:{

int x,y;

int mark[10];

cout<<endl<<"请输入格子位置(如[2,4],则输入:2 4,中间用空格.): [x,y]=";

cin>>x>>y;

getchar();

while (map[x][y]!=0){

cout<<endl<<"警告:此格已经有数!请重新输入."<<endl<<endl;

cout<<"请重新输入格子位置(如[2,4],则输入:2 4,中间用空格.): [x,y]=";

cin>>x>>y;

getchar();

}

int i,j,is,js,count=0;

for(i=1;i<=9;++i)

mark[i]=0;

for(i=0;i<9;++i)

mark[map[x][i]]=1;

for(i=0;i<9;++i)

mark[map[i][y]]=1;

is=x/3\*3;

js=y/3\*3;

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

{

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

mark[map[is+i][js+j]]=1;

}

cout<<endl<<"此格可填数为:";

for(i=1;i<=9;++i)

if(mark[i]==0) {

count++;

cout<<setw(4)<<i;

}

cout<<endl;

cout<<"press enter to continue! "<<endl;

getchar();

break;

}

case 3:{

int x,y;

cout<<endl<<"请输入您要填格子的位置(如[2,4],则输入:2 4,中间用空格.): [x,y]=";

cin>>x>>y;

cout<<"请输入要填入的数: ";

cin>>map[x][y];

cout<<"您填入的格为: map["<<x<<","<<y<<"]="<<map[x][y]<<endl;

cout<<"press enter to continue! "<<endl;

getchar();

getchar();

break;

}

case 4:{

cout<<endl<<"当前随机数独: "<<endl;

display();

cout<<"press enter to continue! "<<endl;

getchar();

getchar();

break;

}

case 5:{

cout<<endl<<"显示结果: "<<endl;

resolve();//没有输入参数，则默认为smod==ALL,见程序开始函数声明。

cout<<"press enter to continue! "<<endl<<endl;

getchar();

getchar();

break;

}

case 6:

return;

default:{

cout<<"输入错误,请重新输入."<<endl;

break;

}

}

}

}

//main函数

void main(){

int blanks;

while(1){

bool exit\_f=true;

cout<<endl;

cout<<"%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

cout<<"%%SUDOKU游戏%% "<<endl;

cout<<"%%白鹤制作%% "<<endl;

cout<<"%%=====================%% "<<endl;

cout<<"%%1.新游戏%% "<<endl;

cout<<"%%2.退出%% "<<endl;

cout<<"%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

int select;

cin>>select;

switch (select){

case 1: //开始新游戏

{

while(exit\_f)

{

cout<<"%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

cout<<"%%请选择游戏难度%% "<<endl;

cout<<"%%=====================%% "<<endl;

cout<<"%%%% "<<endl;

cout<<"%%1.简单%% "<<endl;

cout<<"%%2.中等%% "<<endl;

cout<<"%%3.困难%% "<<endl;

cout<<"%%4.返回%% "<<endl;

cout<<"%%%% "<<endl;

cout<<"%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

/\*

cout<<endl<<"请选择游戏难度: "<<endl;

cout<<"======================= "<<endl;

cout<<"1.简单"<<endl;

cout<<"2.中等"<<endl;

cout<<"3.困难"<<endl;

cout<<"4. 解一个已知数独"<<endl;

cout<<"5. 退出"<<endl; cout<<"======================= "<<endl<<endl;\*/

int level;

cin>>level;

switch(level){

case 1:{

blanks=33;

CSudoku s(blanks);

break;

}

case 2:{

blanks=36;

CSudoku s(blanks);

break;

}

case 3:{

blanks=39;

CSudoku s(blanks);

break;

}

case 4:{

exit\_f=false;

break;

}

default: {

cout<<"输入错误,请重新输入."<<endl<<endl;

break;

}

}

}

break;

}

case 2: return;

default: {

cout<<"输入有误,请重新选择!"<<endl;

break;

}

}

}

}