**#pragma warning(disable: 4996)**

**#include<iostream>**

**#include<stdlib.h> /\*callocやrand,srandを使うのに必要\*/**

**#include<ctime>**

**#include<process.h>**

**#include<conio.h> //while (!\_kbhit()); を使うためのお呪い。**

**#include <math.h>**

**#include <cmath>**

**#include<process.h>**

**#include <iomanip>**

**#include <stdio.h>**

**#include <stdlib.h>**

**using namespace std;**

**const unsigned char th =36**

**void f(unsigned char g, unsigned char a);**

**void f1(unsigned char g, unsigned char a);**

**void sayu(unsigned char a);**

**void jyoge(unsigned char a);**

**void ten(unsigned char a);**

**void sayujyoge(unsigned char a);**

**void kokoro(unsigned char a);**

**void kyokusyokaiseki(unsigned char s, unsigned char t, unsigned char a);**

**void dainyu(unsigned char a);**

**char keizoku = 1;**

**unsigned char\* ht = (unsigned char\*)calloc(th, sizeof(unsigned char));**

**unsigned u = (unsigned)time(NULL);**

**void sudoku(void\* a);**

**void syokika(unsigned char a);**

**void nyuryokujyun(unsigned char g, unsigned char a);**

**unsigned char ks(unsigned char a);**

**unsigned char totalkaiseki(unsigned char a);**

**void kaitosakusei(unsigned char a);**

**void linehaijyo(unsigned char a);**

**void souhokakutei(unsigned char a);**

**void tont(unsigned char a);**

**void listhou(unsigned char a);**

**void gyou(unsigned char a);**

**void retu(unsigned char a);**

**void block(unsigned char a);**

**void sgyou(unsigned char a);**

**void sretu(unsigned char a);**

**void sblock(unsigned char a);**

**void tontgyou(unsigned char a);**

**void tontretu(unsigned char a);**

**void tontblock(unsigned char a);**

**void shphaijyo(unsigned char a);**

**unsigned char m[th][9][9], mx[th][9][9], wb[th][9][9][9], rlst[th][9][9][9], y[th][81], x[th][81];**

**unsigned char yy[th][81], xx[th][81], cm[th][9][9], ccm[th][9][9];**

**unsigned char hnt, owari = 0, krn[th], cn[th];**

**unsigned char S;**

**unsigned char main() {**

 **unsigned char ii[th];**

 **hnt = 20;**

 **clock\_t hj, ow;**

 **hj = clock();**

**tobi:;**

 **u = (unsigned)time(NULL);**

 **keizoku = 1;**

 **for (unsigned char i = 1; i < th; i += 1) {**

 **ii[i] = i;**

 **\_beginthread(sudoku, 0, &ii[i]); //新しいスレッドを起動して、そのスレッド上で関数f1を働かせなさいの命令**

 **}**

 **srand(u);**

 **while (1) {**

 **syokika(0);**

 **int sentaku = rand() % 4;**

 **if (sentaku == 0) sayu(0);**

 **if (sentaku == 1) jyoge(0);**

 **if (sentaku == 2) ten(0);**

 **if (sentaku == 3) sayujyoge(0);**

 **if (sentaku == 4) kokoro(0);**

 **if (keizoku == 0)break;**

 **cn[0] = 0;**

 **f1(0, 0);**

 **if (keizoku == 0)break;**

 **dainyu(0);**

 **if (keizoku == 0)break;**

 **cn[0] = 0;**

 **krn[0] = 81 - hnt;**

 **kaitosakusei(0);**

 **//f(hnt, 0);**

 **if (cn[0] == 1) {**

 **S = sentaku;**

 **ht[0] = 1;**

 **keizoku = 0;**

 **break;**

 **}**

 **}**

 **while (keizoku);**

 **unsigned char ik;**

 **for (unsigned char i = 0; i < th; i++)if (ht[i] == 1) { ik = i; break; }**

 **//数独を見つけたスレッドを特定した。**

 **FILE\* fp;**

 **/\*ファイル(save.csv)に書き込む\*/**

 **if ((fp = fopen("a.csv", "w")) != NULL) {**

 **for (unsigned char i = 0; i < 9; i++) {**

 **for (unsigned char j = 0; j < 9; j++) {**

 **fprintf(fp, "%d,\n", cm[ik][i][j]);**

 **}**

 **}**

 **for (unsigned char i = 0; i < 9; i++) {**

 **for (unsigned char j = 0; j < 9; j++) {**

 **/\*カンマで区切ることでCSVファイルとする\*/**

 **fprintf(fp, "%d,\n", m[ik][i][j]);**

 **}**

 **}**

 **}**

 **/\*忘れずに閉じる\*/**

 **fclose(fp);**

 **//for (int i = 0; i < 10; i++) {**

 **// if (i % 3 == 0) {**

 **// cout << " ";**

 **// for (int j = 0; j < 12; j++) {**

 **// cout << "- ";**

 **// }**

 **// cout << " ";**

 **// cout << endl;**

 **// }**

 **// if (i == 9)break;**

 **// for (int j = 0; j < 10; j++) {**

 **// if (j % 3 == 0) {**

 **// cout << "| ";**

 **// }**

 **// if (j < 9) {**

 **// if (cm[ik][i][j] == 0)cout << "\* "; else cout << +cm[ik][i][j] << " ";//問題**

 **// }**

 **// }**

 **// cout << endl;**

 **//}**

 **//cout << endl;**

 **//for (int i = 0; i < 10; i++) {**

 **// if (i % 3 == 0) {**

 **// cout << " ";**

 **// for (int j = 0; j < 12; j++) {**

 **// cout << "- ";**

 **// }**

 **// cout << " ";**

 **// cout << endl;**

 **// }**

 **// if (i == 9)break;**

 **// for (int j = 0; j < 10; j++) {**

 **// if (j % 3 == 0) {**

 **// cout << "| ";**

 **// }**

 **// if (j < 9) {**

 **// if (m[ik][i][j] == 0)cout << "\* "; else cout << +m[ik][i][j] << " ";//問題**

 **// }**

 **// }**

 **// cout << endl;**

 **//}**

 **if (ks(ik) == 1)cout << "〇" << endl;**

 **if (S == 0)cout << "左右対称型" << endl;**

 **if (S == 1)cout << "上下対称型" << endl;**

 **if (S == 2)cout << "点対称型" << endl;**

 **if (S == 3)cout << "線対称型かつ点対称型" << endl;**

 **ow = clock();**

 **cout << "計算時間は" << (double)(ow - hj) / CLOCKS\_PER\_SEC << "秒です。" << endl;**

 **//while (!\_kbhit()); //待機させるための命令**

 **return 0;**

**}**

**void kaitosakusei(unsigned char a) {**

 **unsigned char i, kkrn;**

 **if (totalkaiseki(a) == 1) {**

 **cn[a] = 2;**

 **return;**

 **}**

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

 **kkrn = krn[a];**

 **linehaijyo(a);**

 **if (cn[a] > 0) return;**

 **if (krn[a] == 0) {**

 **cn[a] = 1;**

 **return;**

 **}**

 **souhokakutei(a);**

 **tont(a);**

 **shphaijyo(a);**

 **if (krn[a] == 0) {**

 **cn[a] = 1;**

 **return;**

 **}**

 **listhou(a);**

 **if (cn[a] > 0)return;**

 **if (krn[a] == 0) {**

 **cn[a] = 1;**

 **return;**

 **}**

 **if (kkrn == krn[a]) {**

 **cn[a] = 0;**

 **return;**

 **}**

 **}**

**}**

**unsigned char totalkaiseki(unsigned char a) {**

 **unsigned char i, j;**

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

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

 **if (m[a][i][j] == 0) {**

 **kyokusyokaiseki(i, j, a);**

 **if (mx[a][i][j] == 0) {**

 **return(1);**

 **}**

 **}**

 **}**

 **}**

 **return(0);**

**}**

**void gyou(unsigned char a) {**

 **unsigned char i, j, k, w, jk, kk, s, t;**

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

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

 **w = 0;**

 **for (k = 0; k < 9; k++) {**

 **if (m[a][j][k] == 0) {**

 **if (wb[a][j][k][i] == 0) {**

 **jk = j;**

 **kk = k;**

 **w = w + 1;**

 **}**

 **}**

 **}**

 **if (w == 1) {**

 **if (krn[a] == 0) {**

 **cn[a] = 2;**

 **return;**

 **}**

 **m[a][jk][kk] = i + 1;**

 **krn[a] = krn[a] - 1;**

 **if (krn[a] == 0) return;**

 **for (k = 0; k < 9; k++) {**

 **if (m[a][jk][k] == 0) {**

 **wb[a][jk][k][m[a][jk][kk] - 1] = 1;**

 **}**

 **if (m[a][k][kk] == 0) {**

 **wb[a][k][kk][m[a][jk][kk] - 1] = 1;**

 **}**

 **s = 3 \* (jk / 3) + (k / 3);**

 **t = 3 \* (kk / 3) + (k % 3);**

 **if (m[a][s][t] == 0) {**

 **wb[a][s][t][m[a][jk][kk] - 1] = 1;**

 **}**

 **}**

 **for (k = 0; k < 9; k++) {**

 **if (m[a][jk][k] == 0)kyokusyokaiseki(jk, k, a);**

 **if (m[a][k][kk] == 0) kyokusyokaiseki(k, kk, a);**

 **s = 3 \* (jk / 3) + (k / 3);**

 **t = 3 \* (kk / 3) + (k % 3);**

 **if (m[a][s][t] == 0)kyokusyokaiseki(s, t, a);**

 **}**

 **}**

 **}**

 **}**

**}**

**void retu(unsigned char a) {**

 **unsigned char i, j, k, w, jk, kk, s, t;**

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

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

 **w = 0;**

 **for (k = 0; k < 9; k++) {**

 **if (m[a][k][j] == 0) {**

 **if (wb[a][k][j][i] == 0) {**

 **jk = j;**

 **kk = k;**

 **w = w + 1;**

 **}**

 **}**

 **}**

 **if (w == 1) {**

 **if (krn[a] == 0) {**

 **cn[a] = 2;**

 **return;**

 **}**

 **m[a][kk][jk] = i + 1;**

 **krn[a] = krn[a] - 1;**

 **if (krn[a] == 0) return;**

 **for (k = 0; k < 9; k++) {**

 **if (m[a][kk][k] == 0) {**

 **wb[a][kk][k][m[a][kk][jk] - 1] = 1;**

 **}**

 **if (m[a][k][jk] == 0) {**

 **wb[a][k][jk][m[a][kk][jk] - 1] = 1;**

 **}**

 **s = 3 \* (kk / 3) + (k / 3);**

 **t = 3 \* (jk / 3) + (k % 3);**

 **if (m[a][s][t] == 0) {**

 **wb[a][s][t][m[a][kk][jk] - 1] = 1;**

 **}**

 **}**

 **for (k = 0; k < 9; k++) {**

 **if (m[a][kk][k] == 0)kyokusyokaiseki(kk, k, a);**

 **if (m[a][k][jk] == 0)kyokusyokaiseki(k, jk, a);**

 **s = 3 \* (kk / 3) + (k / 3);**

 **t = 3 \* (jk / 3) + (k % 3);**

 **if (m[a][s][t] == 0)kyokusyokaiseki(s, t, a);**

 **}**

 **}**

 **}**

 **}**

**}**

**void block(unsigned char a) {**

 **unsigned char i, j, k, w, sk, tk, s, t, s1, t1;**

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

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

 **w = 0;**

 **for (k = 0; k < 9; k++) {**

 **s = 3 \* (j / 3) + (k / 3);**

 **t = 3 \* (j % 3) + (k % 3);**

 **if (m[a][s][t] == 0) {**

 **if (wb[a][s][t][i] == 0) {**

 **sk = s;**

 **tk = t;**

 **w = w + 1;**

 **}**

 **}**

 **}**

 **if (w == 1) {**

 **if (krn[a] == 0) {**

 **cn[a] = 2;**

 **return;**

 **}**

 **m[a][sk][tk] = i + 1;**

 **krn[a] = krn[a] - 1;**

 **if (krn[a] == 0) return;**

 **for (k = 0; k < 9; k++) {**

 **if (m[a][sk][k] == 0) {**

 **wb[a][sk][k][m[a][sk][tk] - 1] = 1;**

 **}**

 **if (m[a][k][tk] == 0) {**

 **wb[a][k][tk][m[a][sk][tk] - 1] = 1;**

 **}**

 **s1 = 3 \* (sk / 3) + (k / 3);**

 **t1 = 3 \* (tk / 3) + (k % 3);**

 **if (m[a][s1][t1] == 0) {**

 **wb[a][s1][t1][m[a][sk][tk] - 1] = 1;**

 **}**

 **}**

 **for (k = 0; k < 9; k++) {**

 **if (m[a][sk][k] == 0)kyokusyokaiseki(sk, k, a);**

 **if (m[a][k][tk] == 0)kyokusyokaiseki(k, tk, a);**

 **s1 = 3 \* (sk / 3) + (k / 3);**

 **t1 = 3 \* (tk / 3) + (k % 3);**

 **if (m[a][s1][t1] == 0)kyokusyokaiseki(s1, t1, a);**

 **}**

 **}**

 **}**

 **}**

**}**

**void linehaijyo(unsigned char a) {**

 **gyou(a);**

 **if (krn[a] == 0) {**

 **cn[a] = 1;**

 **return;**

 **}**

 **retu(a);**

 **if (krn[a] == 0) {**

 **cn[a] = 1;**

 **return;**

 **}**

 **block(a);**

**}**

**void sgyou(unsigned char a) {**

 **unsigned char i, j, k, w, onoff[9];**

 **unsigned char ckotae[9], l;**

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

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

 **for (k = j + 1; k < 9; k++) {**

 **if (mx[a][i][j] == 2 && mx[a][i][k] == 2) {**

 **for (l = 0; l < 9; l++) {**

 **onoff[l] = 0;**

 **}**

 **for (l = 0; l < 2; l++) {**

 **onoff[rlst[a][i][j][l] - 1] = 1;**

 **onoff[rlst[a][i][k][l] - 1] = 1;**

 **}**

 **w = 0;**

 **for (l = 0; l < 9; l++) {**

 **if (onoff[l] == 1) {**

 **ckotae[w] = l;**

 **w = w + 1;**

 **}**

 **}**

 **if (w == 2) {**

 **for (l = 0; l < 9; l++) {**

 **if (l != j && l != k) {**

 **if (m[a][i][l] == 0) {**

 **wb[a][i][l][ckotae[0]] = 1;**

 **wb[a][i][l][ckotae[1]] = 1;**

 **}**

 **}**

 **}**

 **for (l = 0; l < 9; l++) {**

 **if (l != j && l != k) {**

 **if (m[a][i][l] == 0) {**

 **kyokusyokaiseki(i, l, a);**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

**}**

**void sretu(unsigned char a) {**

 **unsigned char i, j, k, w, onoff[9];**

 **unsigned char ckotae[9], l;**

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

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

 **for (k = j + 1; k < 9; k++) {**

 **if (mx[a][j][i] == 2 && mx[a][k][i] == 2) {**

 **for (l = 0; l < 9; l++) {**

 **onoff[l] = 0;**

 **}**

 **for (l = 0; l < 2; l++) {**

 **onoff[rlst[a][j][i][l] - 1] = 1;**

 **onoff[rlst[a][k][i][l] - 1] = 1;**

 **}**

 **w = 0;**

 **for (l = 0; l < 9; l++) {**

 **if (onoff[l] == 1) {**

 **ckotae[w] = l;**

 **w = w + 1;**

 **}**

 **}**

 **if (w == 2) {**

 **for (l = 0; l < 9; l++) {**

 **if (l != j && l != k) {**

 **if (m[a][l][i] == 0) {**

 **wb[a][l][i][ckotae[0]] = 1;**

 **wb[a][l][i][ckotae[1]] = 1;**

 **}**

 **}**

 **}**

 **for (l = 0; l < 9; l++) {**

 **if (l != j && l != k) {**

 **if (m[a][l][i] == 0) {**

 **kyokusyokaiseki(l, i, a);**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

**}**

**void sblock(unsigned char a) {**

 **unsigned char i, j, k, w, onoff[9], s1, s2, t1, t2, s3, t3;**

 **unsigned char ckotae[9], l;**

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

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

 **s1 = 3 \* (i / 3) + (j / 3);**

 **t1 = 3 \* (i % 3) + (j % 3);**

 **for (k = j + 1; k < 9; k++) {**

 **s2 = 3 \* (i / 3) + (k / 3);**

 **t2 = 3 \* (i % 3) + (k % 3);**

 **if (mx[a][s1][t1] == 2 && mx[a][s2][t2] == 2) {**

 **for (l = 0; l < 9; l++) {**

 **onoff[l] = 0;**

 **}**

 **for (l = 0; l < 2; l++) {**

 **onoff[rlst[a][s1][t1][l] - 1] = 1;**

 **onoff[rlst[a][s2][t2][l] - 1] = 1;**

 **}**

 **w = 0;**

 **for (l = 0; l < 9; l++) {**

 **if (onoff[l] == 1) {**

 **ckotae[w] = l;**

 **w = w + 1;**

 **}**

 **}**

 **if (w == 2) {**

 **for (l = 0; l < 9; l++) {**

 **if (l != j && l != k) {**

 **s3 = 3 \* (i / 3) + (l / 3);**

 **t3 = 3 \* (i % 3) + (l % 3);**

 **if (m[a][s3][t3] == 0) {**

 **wb[a][s3][t3][ckotae[0]] = 1;**

 **wb[a][s3][t3][ckotae[1]] = 1;**

 **}**

 **}**

 **}**

 **for (l = 0; l < 9; l++) {**

 **if (l != j && l != k) {**

 **s3 = 3 \* (i / 3) + (l / 3);**

 **t3 = 3 \* (i % 3) + (l % 3);**

 **if (m[a][s3][t3] == 0) {**

 **kyokusyokaiseki(s3, t3, a);**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

**}**

**void souhokakutei(unsigned char a) {**

 **sgyou(a);**

 **sretu(a);**

 **sblock(a);**

**}**

**void tontgyou(unsigned char a) {**

 **unsigned char i, j, k, w, onoff[9];**

 **unsigned char ckotae[9], l, n;**

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

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

 **for (k = j + 1; k < 9; k++) {**

 **for (l = k + 1; l < 9; l++) {**

 **if ((m[a][i][j] == 2 || m[a][i][j] == 3) && (m[a][i][k] == 2 || m[a][i][k] == 3) && (m[a][i][l] == 2 || m[a][i][l] == 3)) {**

 **for (n = 0; n < 9; n++) {**

 **onoff[n] = 0;**

 **}**

 **for (n = 0; n < m[a][i][j]; n++) {**

 **onoff[rlst[a][i][j][n] - 1] = 1;**

 **}**

 **for (n = 0; n < m[a][i][k]; n++) {**

 **onoff[rlst[a][i][k][n] - 1] = 1;**

 **}**

 **for (n = 0; n < m[a][i][l]; n++) {**

 **onoff[rlst[a][i][l][n] - 1] = 1;**

 **}**

 **w = 0;**

 **for (n = 0; n < 9; n++) {**

 **if (onoff[n] == 1) {**

 **ckotae[w] = n;**

 **w = w + 1;**

 **}**

 **}**

 **if (w == 3) {**

 **for (n = 0; n < 9; n++) {**

 **if (n != j && n != k && n != l) {**

 **if (m[a][i][n] == 0) {**

 **wb[a][i][n][ckotae[0]] = 1;**

 **wb[a][i][n][ckotae[1]] = 1;**

 **wb[a][i][n][ckotae[2]] = 1;**

 **}**

 **}**

 **}**

 **for (n = 0; n < 9; n++) {**

 **if (n != j && n != k && n != l) {**

 **if (m[a][i][n] == 0) {**

 **kyokusyokaiseki(i, n, a);**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

**}**

**void tontretu(unsigned char a) {**

 **unsigned char i, j, k, w, onoff[9];**

 **unsigned char ckotae[9], l, n;**

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

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

 **for (k = j + 1; k < 9; k++) {**

 **for (l = k + 1; l < 9; l++) {**

 **if ((m[a][j][i] == 2 || m[a][j][i] == 3) && (m[a][k][i] == 2 || m[a][k][i] == 3) && (m[a][l][i] == 2 || m[a][l][i] == 3)) {**

 **for (n = 0; n < 9; n++) {**

 **onoff[n] = 0;**

 **}**

 **for (n = 0; n < m[a][j][i]; n++) {**

 **onoff[rlst[a][j][i][n] - 1] = 1;**

 **}**

 **for (n = 0; n < m[a][k][i]; n++) {**

 **onoff[rlst[a][k][i][n] - 1] = 1;**

 **}**

 **for (n = 0; n < m[a][l][i]; n++) {**

 **onoff[rlst[a][l][i][n] - 1] = 1;**

 **}**

 **w = 0;**

 **for (n = 0; n < 9; n++) {**

 **if (onoff[n] == 1) {**

 **ckotae[w] = n;**

 **w = w + 1;**

 **}**

 **}**

 **if (w == 3) {**

 **for (n = 0; n < 9; n++) {**

 **if (n != j && n != k && n != l) {**

 **if (m[a][n][i] == 0) {**

 **wb[a][n][i][ckotae[0]] = 1;**

 **wb[a][n][i][ckotae[1]] = 1;**

 **wb[a][n][i][ckotae[2]] = 1;**

 **}**

 **}**

 **}**

 **for (n = 0; n < 9; n++) {**

 **if (n != j && n != k && n != l) {**

 **if (m[a][n][i] == 0) {**

 **kyokusyokaiseki(n, i, a);**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

**}**

**void tontblock(unsigned char a) {**

 **unsigned char i, j, k, w, onoff[9];**

 **unsigned char ckotae[9], l, n;**

 **unsigned char s1, s2, s3, s4, t1, t2, t3, t4;**

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

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

 **s1 = 3 \* (i / 3) + (j / 3);**

 **t1 = 3 \* (i % 3) + (j % 3);**

 **for (k = j + 1; k < 9; k++) {**

 **s2 = 3 \* (i / 3) + (k / 3);**

 **t2 = 3 \* (i % 3) + (k % 3);**

 **for (l = k + 1; l < 9; l++) {**

 **s3 = 3 \* (i / 3) + (l / 3);**

 **t3 = 3 \* (i % 3) + (l % 3);**

 **if ((m[a][s1][t1] == 2 || m[a][s1][t1] == 3) && (m[a][s2][t2] == 2 || m[a][s2][t2] == 3) && (m[a][s3][t3] == 2 || m[a][s3][t3] == 3)) {**

 **for (n = 0; n < 9; n++) {**

 **onoff[n] = 0;**

 **}**

 **for (n = 0; n < m[a][s1][t1]; n++) {**

 **onoff[rlst[a][s1][t1][n] - 1] = 1;**

 **}**

 **for (n = 0; n < m[a][s2][t2]; n++) {**

 **onoff[rlst[a][s2][t2][n] - 1] = 1;**

 **}**

 **for (n = 0; n < m[a][s3][t3]; n++) {**

 **onoff[rlst[a][s3][t3][n] - 1] = 1;**

 **}**

 **w = 0;**

 **for (n = 0; n < 9; n++) {**

 **if (onoff[n] == 1) {**

 **ckotae[w] = n;**

 **w = w + 1;**

 **}**

 **}**

 **if (w == 3) {**

 **for (n = 0; n < 9; n++) {**

 **if (n != j && n != k && n != l) {**

 **s4 = 3 \* (i / 3) + (n / 3);**

 **t4 = 3 \* (i % 3) + (n % 3);**

 **if (m[a][s4][t4] == 0) {**

 **wb[a][s4][t4][ckotae[0]] = 1;**

 **wb[a][s4][t4][ckotae[1]] = 1;**

 **wb[a][s4][t4][ckotae[2]] = 1;**

 **}**

 **}**

 **}**

 **for (n = 0; n < 9; n++) {**

 **if (n != j && n != k && n != l) {**

 **s4 = 3 \* (i / 3) + (n / 3);**

 **t4 = 3 \* (i % 3) + (n % 3);**

 **if (m[a][s4][t4] == 0) {**

 **kyokusyokaiseki(s4, t4, a);**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

**}**

**void tont(unsigned char a) {**

 **tontgyou(a);**

 **tontretu(a);**

 **tontblock(a);**

**}**

**void shphaijyo(unsigned char a) {**

 **unsigned char i, j, wx, wy, k;**

 **unsigned char yk[9][9], xk[9][9];**

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

 **wy = 0;**

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

 **wx = 0;**

 **for (k = 0; k < 9; k++) {**

 **if (m[a][j][k] == 0) {**

 **if (wb[a][j][k][i] == 0) {**

 **yk[wy][wx] = j;**

 **xk[wy][wx] = k;**

 **wx = wx + 1;**

 **}**

 **}**

 **}**

 **if (wx == 2) {**

 **wy = wy + 1;**

 **}**

 **}**

 **if (wy == 2) {**

 **if (xk[0][0] == xk[1][0] && xk[0][1] == xk[1][1]) {**

 **for (k = 0; k < 9; k++) {**

 **if (k != xk[0][0] && k != xk[0][1]) {**

 **if (m[a][yk[0][0]][k] == 0) {**

 **wb[a][yk[0][0]][k][i] = 1;**

 **}**

 **if (m[a][yk[1][1]][k] == 0) {**

 **wb[a][yk[1][1]][k][i] = 1;**

 **}**

 **}**

 **}**

 **for (k = 0; k < 9; k++) {**

 **if (k != xk[0][0] && k != xk[0][1]) {**

 **if (m[a][yk[0][0]][k] == 0) {**

 **kyokusyokaiseki(yk[0][0], k, a);**

 **}**

 **if (m[a][yk[1][1]][k] == 0) {**

 **kyokusyokaiseki(yk[1][1], k, a);**

 **}**

 **}**

 **}**

 **for (k = 0; k < 9; k++) {**

 **if (k != yk[0][0] && k != yk[1][1]) {**

 **if (m[a][k][xk[0][0]] == 0) {**

 **wb[a][k][xk[0][0]][i] = 1;**

 **}**

 **if (m[a][k][xk[1][1]] == 0) {**

 **wb[a][k][xk[1][1]][i] = 1;**

 **}**

 **}**

 **}**

 **for (k = 0; k < 9; k++) {**

 **if (k != yk[0][0] && k != yk[1][1]) {**

 **if (m[a][k][xk[0][0]] == 0) {**

 **kyokusyokaiseki(k, xk[0][0], a);**

 **}**

 **if (m[a][k][xk[1][1]] == 0) {**

 **kyokusyokaiseki(k, xk[1][1], a);**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

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

 **wx = 0;**

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

 **wy = 0;**

 **for (k = 0; k < 9; k++) {**

 **if (m[a][k][j] == 0) {**

 **if (wb[a][k][j][i] == 0) {**

 **yk[wx][wy] = k;**

 **xk[wx][wy] = j;**

 **wy = wy + 1;**

 **}**

 **}**

 **}**

 **if (wy == 2) {**

 **wx = wx + 1;**

 **}**

 **}**

 **if (wx == 2) {**

 **if (yk[0][0] == yk[1][0] && yk[0][1] == yk[1][1]) {**

 **for (k = 0; k < 9; k++) {**

 **if (k != yk[0][0] && k != yk[0][1]) {**

 **if (m[a][k][xk[0][0]] == 0) {**

 **wb[a][k][xk[0][0]][i] = 1;**

 **}**

 **if (m[a][k][xk[1][1]] == 0) {**

 **wb[a][k][xk[1][1]][i] = 1;**

 **}**

 **}**

 **}**

 **for (k = 0; k < 9; k++) {**

 **if (k != yk[0][0] && k != yk[1][1]) {**

 **if (m[a][k][xk[0][0]] == 0) {**

 **kyokusyokaiseki(k, xk[0][0], a);**

 **}**

 **if (m[a][k][xk[1][1]] == 0) {**

 **kyokusyokaiseki(k, xk[1][1], a);**

 **}**

 **}**

 **}**

 **for (k = 0; k < 9; k++) {**

 **if (k != xk[0][0] && k != xk[1][1]) {**

 **if (m[a][yk[0][0]][k] == 0) {**

 **wb[a][yk[0][0]][k][i] = 1;**

 **}**

 **if (m[a][yk[1][1]][k] == 0) {**

 **wb[a][yk[1][1]][k][i] = 1;**

 **}**

 **}**

 **}**

 **for (k = 0; k < 9; k++) {**

 **if (k != xk[0][0] && k != xk[1][1]) {**

 **if (m[a][yk[0][0]][k] == 0) {**

 **kyokusyokaiseki(yk[0][0], k, a);**

 **}**

 **if (m[a][yk[1][1]][k] == 0) {**

 **kyokusyokaiseki(yk[1][1], k, a);**

 **}**

 **}**

 **}**

 **}**

 **}**

 **}**

**}**

**void listhou(unsigned char a) {**

 **unsigned char i, j, k, s, t;**

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

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

 **if (m[a][i][j] == 0) {**

 **if (mx[a][i][j] == 0) {**

 **cn[a] = 2;**

 **return;**

 **}**

 **if (mx[a][i][j] == 1) {**

 **m[a][i][j] = rlst[a][i][j][0];**

 **krn[a] = krn[a] - 1;**

 **if (krn[a] == 0)return;**

 **for (k = 0; k < 9; k++) {**

 **if (m[a][i][k] == 0) {**

 **wb[a][i][k][m[a][i][j] - 1] = 1;**

 **}**

 **if (m[a][k][j] == 0) {**

 **wb[a][k][j][m[a][i][j] - 1] = 1;**

 **}**

 **s = 3 \* (i / 3) + (k / 3);**

 **t = 3 \* (j / 3) + (k % 3);**

 **if (m[a][s][t] == 0) {**

 **wb[a][s][t][m[a][i][j] - 1] = 1;**

 **}**

 **}**

 **for (k = 0; k < 9; k++) {**

 **if (m[a][i][k] == 0) kyokusyokaiseki(i, k, a);**

 **if (m[a][k][j] == 0) kyokusyokaiseki(k, j, a);**

 **s = 3 \* (i / 3) + (k / 3);**

 **t = 3 \* (j / 3) + (k % 3);**

 **if (m[a][s][t] == 0) kyokusyokaiseki(s, t, a);**

 **}**

 **}**

 **}**

 **}**

 **}**

**}**

**void sayu(unsigned char a) {**

 **unsigned char ty, gz;**

 **if (hnt % 2 == 0) {**

 **gz = rand() % 5;**

 **if (gz == 0)ty = 0;**

 **if (gz > 0 && gz < 4)ty = 2;**

 **if (gz == 4)ty = 4;**

 **}**

 **else {**

 **gz = rand() % 7;**

 **if (gz < 4)ty = 1;**

 **if (gz > 3 && gz < 6)ty = 3;**

 **if (gz == 6)ty = 5;**

 **}**

 **//ty = 2;**

 **unsigned char s = rand() % 11;**

 **unsigned char rnk;**

 **if (s == 0) rnk = 47;**

 **if (s == 1) rnk = 7;**

 **if (s == 2) rnk = 11;**

 **if (s == 3) rnk = 13;**

 **if (s == 4) rnk = 17;**

 **if (s == 5) rnk = 19;**

 **if (s == 6) rnk = 23;**

 **if (s == 7) rnk = 29;**

 **if (s == 8) rnk = 31;**

 **if (s == 9) rnk = 37;**

 **if (s == 10) rnk = 41;**

 **//rnk = 4;**

 **unsigned char ss = rand() % 9;**

 **unsigned char sss = rand() % 36;**

 **for (unsigned char i = 0; i < (hnt - ty) / 2; i++) {**

 **xx[a][i] = ((i \* rnk + sss) % 36) / 9;**

 **xx[a][hnt - 1 - i] = 8 - xx[a][i];**

 **yy[a][i] = ((i \* rnk + sss) % 36) % 9;**

 **yy[a][hnt - 1 - i] = yy[a][i];**

 **}**

 **unsigned char tyrnk;**

 **while (1) {**

 **tyrnk = rand() % 6;**

 **if (tyrnk > 2 && tyrnk % 3 != 0)break;**

 **}**

 **for (unsigned char i = 0; i < ty; i++) {**

 **xx[a][i + (hnt - ty) / 2] = 4;**

 **yy[a][i + (hnt - ty) / 2] = (i \* tyrnk + ss) % 9;**

 **}**

**}**

**void jyoge(unsigned char a) {**

 **unsigned char ty, gz;**

 **if (hnt % 2 == 0) {**

 **gz = rand() % 5;**

 **if (gz == 0)ty = 0;**

 **if (gz > 0 && gz < 4)ty = 2;**

 **if (gz == 4)ty = 4;**

 **}**

 **else {**

 **gz = rand() % 7;**

 **if (gz < 4)ty = 1;**

 **if (gz > 3 && gz < 6)ty = 3;**

 **if (gz == 6)ty = 5;**

 **}**

 **//ty = 2;**

 **unsigned char s = rand() % 11;**

 **unsigned char rnk;**

 **if (s == 0) rnk = 47;**

 **if (s == 1) rnk = 7;**

 **if (s == 2) rnk = 11;**

 **if (s == 3) rnk = 13;**

 **if (s == 4) rnk = 17;**

 **if (s == 5) rnk = 19;**

 **if (s == 6) rnk = 23;**

 **if (s == 7) rnk = 29;**

 **if (s == 8) rnk = 31;**

 **if (s == 9) rnk = 37;**

 **if (s == 10) rnk = 41;**

 **//rnk = 4;**

 **unsigned char ss = rand() % 9;**

 **unsigned char sss = rand() % 36;**

 **for (unsigned char i = 0; i < (hnt - ty) / 2; i++) {**

 **yy[a][i] = ((i \* rnk + sss) % 36) / 9;**

 **yy[a][hnt - 1 - i] = 8 - yy[a][i];**

 **xx[a][i] = ((i \* rnk + sss) % 36) % 9;**

 **xx[a][hnt - 1 - i] = xx[a][i];**

 **}**

 **unsigned char tyrnk;**

 **while (1) {**

 **tyrnk = rand() % 6;**

 **if (tyrnk > 2 && tyrnk % 3 != 0)break;**

 **}**

 **for (unsigned char i = 0; i < ty; i++) {**

 **yy[a][i + (hnt - ty) / 2] = 4;**

 **xx[a][i + (hnt - ty) / 2] = (i \* tyrnk + ss) % 9;**

 **}**

**}**

**void ten(unsigned char a) {**

 **unsigned char s, rnk, sss;**

 **s = rand() % 11;**

 **if (s == 0) rnk = 47;**

 **if (s == 1) rnk = 7;**

 **if (s == 2) rnk = 11;**

 **if (s == 3) rnk = 13;**

 **if (s == 4) rnk = 17;**

 **if (s == 5) rnk = 19;**

 **if (s == 6) rnk = 23;**

 **if (s == 7) rnk = 29;**

 **if (s == 8) rnk = 31;**

 **if (s == 9) rnk = 59;**

 **if (s == 10) rnk = 61;**

 **while (1) {**

 **s = rand() % (hnt / 9 + 2);**

 **if ((hnt - s) % 2 == 0) break;**

 **}**

 **sss = rand() \* 40;**

 **if (hnt % 2 == 0) {**

 **for (unsigned char i = 0; i < hnt / 2; i++) {**

 **yy[a][i] = ((i \* rnk + sss) % 40) / 9;**

 **yy[a][hnt - 1 - i] = 8 - yy[a][i];**

 **xx[a][i] = ((i \* rnk + sss) % 40) % 9;**

 **xx[a][hnt - 1 - i] = 8 - xx[a][i];**

 **}**

 **}**

 **else {**

 **for (unsigned char i = 0; i < hnt / 2; i++) {**

 **yy[a][i] = ((i \* rnk + sss) % 40) / 9;**

 **yy[a][hnt - 1 - i] = 8 - yy[a][i];**

 **xx[a][i] = ((i \* rnk + sss) % 40) % 9;**

 **xx[a][hnt - 1 - i] = 8 - xx[a][i];**

 **}**

 **yy[a][(hnt - 1) / 2] = 4;**

 **xx[a][(hnt - 1) / 2] = 4;**

 **}**

**}**

**void sayujyoge(unsigned char a) {**

 **unsigned char sss, b, rnk, s, mns;**

 **unsigned char kh[16];**

 **if (hnt % 2 == 0) {**

 **if (hnt % 4 == 0) {**

 **s = rand() % 5;**

 **if (s == 0) rnk = 3;**

 **if (s == 1) rnk = 5;**

 **if (s == 2) rnk = 7;**

 **if (s == 3) rnk = 11;**

 **if (s == 4) rnk = 13;**

 **sss = rand() % 16;**

 **b = hnt / 4 - 1;**

 **for (unsigned char i = 0; i < b + 1; i++) {**

 **kh[i] = (sss + rnk \* i) % 16;**

 **}**

 **for (unsigned char i = 0; i < b + 1; i++) {**

 **yy[a][i] = kh[i] / 4;**

 **xx[a][i] = kh[i] % 4;**

 **yy[a][2 \* (b + 1) - i - 1] = yy[a][i];**

 **xx[a][2 \* (b + 1) - i - 1] = 8 - xx[a][i];**

 **yy[a][3 \* (b + 1) - i - 1] = 8 - yy[a][i];**

 **xx[a][3 \* (b + 1) - i - 1] = xx[a][i];**

 **yy[a][4 \* (b + 1) - i - 1] = 8 - yy[a][i];**

 **xx[a][4 \* (b + 1) - i - 1] = 8 - xx[a][i];**

 **}**

 **return;**

 **}**

 **s = rand() % 3;**

 **if (s < 2) mns = 1; else mns = 3;**

 **s = rand() % 4;**

 **for (unsigned char i = 0; i < mns + 1; i++) {**

 **xx[a][i] = 4;**

 **xx[a][2 \* mns - 1 - i] = 4;**

 **yy[a][i] = (s + 3 \* i) % 4;**

 **yy[a][2 \* mns - 1 - i] = 8 - yy[a][i];**

 **}**

 **b = (hnt - 2 \* mns) / 4 - 1;**

 **s = rand() % 5;**

 **if (s == 0) rnk = 3;**

 **if (s == 1) rnk = 5;**

 **if (s == 2) rnk = 7;**

 **if (s == 3) rnk = 11;**

 **if (s == 4) rnk = 13;**

 **sss = rand() % 16;**

 **for (unsigned char i = 0; i < b + 1; i++) {**

 **kh[i] = (sss + rnk \* i) % 16;**

 **}**

 **for (unsigned char i = 0; i < b + 1; i++) {**

 **yy[a][2 \* mns + i] = kh[i] / 4;**

 **xx[a][2 \* mns + i] = kh[i] % 4;**

 **yy[a][2 \* mns + 2 \* (b + 1) - i - 1] = yy[a][2 \* mns + i];**

 **xx[a][2 \* mns + 2 \* (b + 1) - i - 1] = 8 - xx[a][2 \* mns + i];**

 **yy[a][2 \* mns + 3 \* (b + 1) - i - 1] = 8 - yy[a][2 \* mns + i];**

 **xx[a][2 \* mns + 3 \* (b + 1) - i - 1] = xx[a][2 \* mns + i];**

 **yy[a][2 \* mns + 4 \* (b + 1) - i - 1] = 8 - yy[a][2 \* mns + i];**

 **xx[a][2 \* mns + 4 \* (b + 1) - i - 1] = 8 - xx[a][2 \* mns + i];**

 **}**

 **return;**

 **}**

 **if (hnt % 2 == 1) {**

 **xx[a][0] = 4;**

 **yy[a][0] = 4;**

 **if (((hnt - 1) % 4) == 0) {**

 **s = rand() % 5;**

 **if (s == 0) rnk = 3;**

 **if (s == 1) rnk = 5;**

 **if (s == 2) rnk = 7;**

 **if (s == 3) rnk = 11;**

 **if (s == 4) rnk = 13;**

 **sss = rand() % 16;**

 **b = (hnt - 1) / 4 - 1;**

 **for (unsigned char i = 0; i < b + 1; i++) {**

 **kh[i] = (sss + rnk \* i) % 16;**

 **}**

 **for (unsigned char i = 0; i < b + 1; i++) {**

 **yy[a][1 + i] = kh[i] / 4;**

 **xx[a][1 + i] = kh[i] % 4;**

 **yy[a][1 + 2 \* (b + 1) - i - 1] = yy[a][1 + i];**

 **xx[a][1 + 2 \* (b + 1) - i - 1] = 8 - xx[a][1 + i];**

 **yy[a][1 + 3 \* (b + 1) - i - 1] = 8 - yy[a][1 + i];**

 **xx[a][1 + 3 \* (b + 1) - i - 1] = xx[a][1 + i];**

 **yy[a][1 + 4 \* (b + 1) - i - 1] = 8 - yy[a][1 + i];**

 **xx[a][1 + 4 \* (b + 1) - i - 1] = 8 - xx[a][1 + i];**

 **}**

 **return;**

 **}**

 **s = rand() % 3;**

 **if (s < 2) mns = 1; else mns = 3;**

 **s = rand() % 4;**

 **mns = 3;**

 **for (unsigned char i = 0; i < mns; i++) {**

 **xx[a][1 + i] = 4;**

 **xx[a][1 + 2 \* mns - 1 - i] = 4;**

 **yy[a][1 + i] = (s + 3 \* i) % 4;**

 **yy[a][1 + 2 \* mns - 1 - i] = 8 - yy[a][1 + i];**

 **}**

 **b = (hnt - 1 - 2 \* mns) / 4 - 1;**

 **s = rand() % 4;**

 **if (s == 0) rnk = 3;**

 **if (s == 1) rnk = 5;**

 **if (s == 2) rnk = 7;**

 **if (s == 3) rnk = 11;**

 **sss = rand() % 16;**

 **for (unsigned char i = 0; i < b + 1; i++) {**

 **kh[i] = (sss + rnk \* i) % 16;**

 **}**

 **for (unsigned char i = 0; i < b + 1; i++) {**

 **yy[a][1 + 2 \* mns + i] = kh[i] / 4;**

 **xx[a][1 + 2 \* mns + i] = kh[i] % 4;**

 **yy[a][1 + 2 \* mns + 2 \* (b + 1) - i - 1] = yy[a][1 + 2 \* mns + i];**

 **xx[a][1 + 2 \* mns + 2 \* (b + 1) - i - 1] = 8 - xx[a][1 + 2 \* mns + i];**

 **yy[a][1 + 2 \* mns + 3 \* (b + 1) - i - 1] = 8 - yy[a][1 + 2 \* mns + i];**

 **xx[a][1 + 2 \* mns + 3 \* (b + 1) - i - 1] = xx[a][1 + 2 \* mns + i];**

 **yy[a][1 + 2 \* mns + 4 \* (b + 1) - i - 1] = 8 - yy[a][1 + 2 \* mns + i];**

 **xx[a][1 + 2 \* mns + 4 \* (b + 1) - i - 1] = 8 - xx[a][1 + 2 \* mns + i];**

 **}**

 **}**

**}**

**void kokoro(unsigned char a) {**

 **unsigned char b[8][8];**

 **for (unsigned char i = 0; i < 5; i++) {**

 **yy[a][i] = 8 - i;**

 **xx[a][i] = 4 - i;**

 **}**

 **for (unsigned char i = 5; i < 9; i++) {**

 **yy[a][i] = yy[a][i - 4];**

 **xx[a][i] = 8 - xx[a][i - 4];**

 **}**

 **yy[a][9] = 3;**

 **xx[a][9] = 0;**

 **yy[a][25] = 3;**

 **xx[a][25] = 8;**

 **for (unsigned char i = 11; i < 13; i++) {**

 **yy[a][i] = 13 - i;**

 **xx[a][i] = i - 11;**

 **yy[a][i + 2] = yy[a][i];**

 **xx[a][i + 2] = 8 - xx[a][i];**

 **}**

 **for (unsigned char i = 15; i < 125; i++) {**

 **yy[a][i] = i - 14;**

 **xx[a][i] = i - 13;**

 **yy[a][i + 2] = yy[a][i];**

 **xx[a][i + 2] = 8 - xx[a][i];**

 **}**

 **yy[a][19] = 3;**

 **xx[a][19] = 4;**

 **for (unsigned char i = 0; i < 20; i++) {**

 **b[yy[a][i]][xx[a][i]] = 10;**

 **}**

 **unsigned char sa, h;**

 **sa = hnt - 20;**

 **for (unsigned char i = 1; i < 8; i++) {**

 **h = 0;**

 **for (unsigned char j = 1; j < 9; j++) {**

 **if (b[j - 1][i] == 10) h = h + 1;**

 **if (h == 1 && b[j][i] != 10) b[j][i] = 3;**

 **}**

 **}**

 **unsigned char k, i, j;**

 **if (hnt % 2 == 0) {**

 **for (unsigned char k = 20; k < 20 + sa / 2; k++) {**

 **while (1) {**

 **i = rand() % 9;**

 **j = rand() % 4;**

 **if (b[i][j] == 3) {**

 **b[i][j] = 4;**

 **b[i][8 - j] = 4;**

 **yy[a][k] = i;**

 **xx[a][k] = j;**

 **yy[a][k + sa / 2] = i;**

 **xx[a][k + sa / 2] = 8 - j;**

 **break;**

 **}**

 **}**

 **}**

 **return;**

 **}**

 **else {**

 **while (1) {**

 **j = 4 + rand() % 4;**

 **if (b[j][4] == 3) {**

 **b[j][4] = 4;**

 **yy[a][20] = j;**

 **xx[a][20] = 4;**

 **break;**

 **}**

 **}**

 **if (hnt > 21) {**

 **for (unsigned char i = 21; i < 21 + (hnt - 21) / 2; i++) {**

 **unsigned char s, t;**

 **while (1) {**

 **s = rand() % 9;**

 **t = rand() % 4;**

 **if (b[s][t] == 3) {**

 **b[s][t] = 4;**

 **b[s][8 - t] = 4;**

 **yy[a][i] = s;**

 **xx[a][i] = t;**

 **yy[a][i + (hnt - 21) / 2] = s;**

 **xx[a][i + (hnt - 21) / 2] = 8 - t;**

 **break;**

 **}**

 **}**

 **}**

 **}**

 **}**

**}**

**void sudoku(void\* aa) {**

 **unsigned char a = \*(unsigned char\*)aa;**

 **srand(u - 19 \* (a + 1));**

 **while (1) {**

 **syokika(a);**

 **int sentaku = rand() % 4;**

 **if (sentaku == 0) sayu(a);**

 **if (sentaku == 1) jyoge(a);**

 **if (sentaku == 2) ten(a);**

 **if (sentaku == 3) sayujyoge(a);**

 **if (sentaku == 4) kokoro(a);**

 **if (keizoku == 0)return;**

 **cn[a] = 0;**

 **f1(0, a);**

 **if (keizoku == 0)return;**

 **dainyu(a);**

 **if (keizoku == 0)return;**

 **cn[a] = 0;**

 **krn[a] = 81 - hnt;**

 **kaitosakusei(a);**

 **//f(hnt, a);**

 **if (cn[a] == 1) {**

 **S = sentaku;**

 **ht[a] = 1;**

 **keizoku = 0;**

 **return;**

 **}**

 **}**

**}**

**void dainyu(unsigned char a) {**

 **for (unsigned char i = 0; i < hnt; i++) {**

 **cm[a][yy[a][i]][xx[a][i]] = m[a][yy[a][i]][xx[a][i]];**

 **}**

**}**

**void syokika(unsigned char a) {**

 **for (unsigned char i = 0; i < 9; i++) {**

 **for (unsigned char j = 0; j < 9; j++) {**

 **m[a][i][j] = 0;**

 **cm[a][i][j] = 0;**

 **mx[a][i][j] = 9;**

 **for (unsigned char k = 0; k < 9; k++) {**

 **wb[a][i][j][k] = 0;**

 **}**

 **}**

 **}**

**}**

**void nyuryokujyun(unsigned char g, unsigned char a) {**

 **unsigned char ik, jk, mn = 100;**

 **for (unsigned char i = 0; i < 9; i++) {**

 **for (unsigned char j = 0; j < 9; j++) {**

 **if (m[a][i][j] == 0) {**

 **if (mx[a][i][j] <= mn) {**

 **mx[a][i][j] = mn;**

 **ik = i;**

 **jk = j;**

 **}**

 **}**

 **}**

 **}**

 **y[a][g] = ik;**

 **x[a][g] = jk;**

 **kyokusyokaiseki(ik, jk, a);**

**}**

**void f(unsigned char g, unsigned char a) {**

 **unsigned char i, j, s, t, p, q, ii, iii, k;**

 **unsigned char gy[9], r[9], b[9];**

 **nyuryokujyun(g, a);**

 **s = y[a][g];**

 **t = x[a][g];**

 **if (mx[a][s][t] == 0)return;**

 **if (cn[a] > 1)return;**

 **if (keizoku == 0)return;**

 **for (i = 0; i < mx[a][s][t]; i++) {**

 **m[a][s][t] = rlst[a][s][t][i];**

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

 **gy[j] = 0;**

 **r[j] = 0;**

 **b[j] = 0;**

 **}**

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

 **if (m[a][s][j] == 0) {**

 **if (wb[a][s][j][m[a][s][t] - 1] == 0) {**

 **wb[a][s][j][m[a][s][t] - 1] = 1;**

 **kyokusyokaiseki(s, j, a);**

 **r[j] = 1;**

 **}**

 **}**

 **}**

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

 **if (m[a][j][t] == 0) {**

 **if (wb[a][j][t][m[a][s][t] - 1] == 0) {**

 **wb[a][j][t][m[a][s][t] - 1] = 1;**

 **kyokusyokaiseki(j, t, a);**

 **gy[j] = 1;**

 **}**

 **}**

 **}**

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

 **p = 3 \* (s / 3) + (j / 3);**

 **q = 3 \* (t / 3) + (j % 3);**

 **if (p != s && q != t) {**

 **if (m[a][p][q] == 0) {**

 **if (wb[a][p][q][m[a][s][t] - 1] == 0) {**

 **wb[a][p][q][m[a][s][t] - 1] = 1;**

 **kyokusyokaiseki(p, q, a);**

 **b[j] = 1;**

 **}**

 **}**

 **}**

 **}**

 **if (keizoku == 0)return;**

 **if (g + 1 < 81) {**

 **f(g + 1, a);**

 **if (cn[a] > 1)return;**

 **if (keizoku == 0)return;**

 **}**

 **else {**

 **cn[a]++;**

 **if (cn[a] == 1) {**

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

 **for (k = 0; k < 9; k++) {**

 **ccm[a][j][k] = m[a][j][k];**

 **}**

 **}**

 **}**

 **if (cn[a] > 1)return;**

 **if (keizoku == 0)return;**

 **}**

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

 **if (r[j] == 1) {**

 **wb[a][s][j][m[a][s][t] - 1] = 0;**

 **}**

 **if (gy[j] == 1) {**

 **wb[a][j][t][m[a][s][t] - 1] = 0;**

 **}**

 **p = 3 \* (s / 3) + (j / 3);**

 **q = 3 \* (t / 3) + (j % 3);**

 **if (b[j] == 1) {**

 **wb[a][p][q][m[a][s][t] - 1] = 0;**

 **}**

 **}**

 **}**

 **m[a][s][t] = 0;**

 **return;**

**}**

**void f1(unsigned char g, unsigned char a) {**

 **unsigned char i, j, s, t, p, q, ii, iii, k;**

 **unsigned char gy[9], r[9], b[9];**

 **if (g < hnt) {**

 **s = yy[a][g];**

 **t = xx[a][g];**

 **kyokusyokaiseki(s, t, a);**

 **}**

 **else {**

 **nyuryokujyun(g, a);**

 **s = y[a][g];**

 **t = x[a][g];**

 **}**

 **if (mx[a][s][t] == 0)return;**

 **ii = rand() % mx[a][s][t];**

 **if (cn[a] == 1)return;**

 **if (keizoku == 0)return;**

 **for (i = 0; i < mx[a][s][t]; i++) {**

 **iii = (i + ii) % mx[a][s][t];**

 **m[a][s][t] = rlst[a][s][t][iii];**

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

 **gy[j] = 0;**

 **r[j] = 0;**

 **b[j] = 0;**

 **}**

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

 **if (m[a][s][j] == 0) {**

 **if (wb[a][s][j][m[a][s][t] - 1] == 0) {**

 **wb[a][s][j][m[a][s][t] - 1] = 1;**

 **kyokusyokaiseki(s, j, a);**

 **r[j] = 1;**

 **}**

 **}**

 **}**

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

 **if (m[a][j][t] == 0) {**

 **if (wb[a][j][t][m[a][s][t] - 1] == 0) {**

 **wb[a][j][t][m[a][s][t] - 1] = 1;**

 **kyokusyokaiseki(j, t, a);**

 **gy[j] = 1;**

 **}**

 **}**

 **}**

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

 **p = 3 \* (s / 3) + (j / 3);**

 **q = 3 \* (t / 3) + (j % 3);**

 **if (p != s && q != t) {**

 **if (m[a][p][q] == 0) {**

 **if (wb[a][p][q][m[a][s][t] - 1] == 0) {**

 **wb[a][p][q][m[a][s][t] - 1] = 1;**

 **kyokusyokaiseki(p, q, a);**

 **b[j] = 1;**

 **}**

 **}**

 **}**

 **}**

 **if (keizoku == 0)return;**

 **if (g + 1 < hnt) {**

 **f1(g + 1, a);**

 **if (cn[a] == 1)return;**

 **if (keizoku == 0)return;**

 **}**

 **else {**

 **cn[a]++;**

 **if (cn[a] == 1)return;**

 **if (keizoku == 0)return;**

 **}**

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

 **if (r[j] == 1) {**

 **wb[a][s][j][m[a][s][t] - 1] = 0;**

 **}**

 **if (gy[j] == 1) {**

 **wb[a][j][t][m[a][s][t] - 1] = 0;**

 **}**

 **p = 3 \* (s / 3) + (j / 3);**

 **q = 3 \* (t / 3) + (j % 3);**

 **if (b[j] == 1) {**

 **wb[a][p][q][m[a][s][t] - 1] = 0;**

 **}**

 **}**

 **}**

 **m[a][s][t] = 0;**

 **return;**

**}**

**void kyokusyokaiseki(unsigned char s, unsigned char t, unsigned char a) {**

 **unsigned char i, w = 0;**

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

 **if (wb[a][s][t][i] == 0) {**

 **rlst[a][s][t][w] = i + 1;**

 **w++;**

 **}**

 **}**

 **mx[a][s][t] = w;**

**}**

**unsigned char ks(unsigned char a) {**

 **unsigned char p[9], s, t;**

 **for (unsigned char i = 0; i < 9; i++) {**

 **p[i] = 0;**

 **for (unsigned char j = 0; j < 9; j++) {**

 **for (unsigned char k = 0; k < 9; k++) {**

 **p[m[a][j][k] - 1] = 1;**

 **}**

 **}**

 **for (unsigned char j = 0; j < 9; j++) {**

 **if (p[j] == 0)return(0);**

 **}**

 **}**

 **for (unsigned char i = 0; i < 9; i++) {**

 **p[i] = 0;**

 **for (unsigned char j = 0; j < 9; j++) {**

 **for (unsigned char k = 0; k < 9; k++) {**

 **p[m[a][k][j] - 1] = 1;**

 **}**

 **}**

 **for (unsigned char j = 0; j < 9; j++) {**

 **if (p[j] == 0)return(0);**

 **}**

 **}**

 **for (unsigned char i = 0; i < 9; i++) {**

 **p[i] = 0;**

 **for (unsigned char j = 0; j < 9; j++) {**

 **for (unsigned char k = 0; k < 9; k++) {**

 **s = 3 \* (i / 3) + (j / 3);**

 **t = 3 \* (i % 3) + (j % 3);**

 **p[m[a][s][t] - 1] = 1;**

 **}**

 **}**

 **for (unsigned char j = 0; j < 9; j++) {**

 **if (p[j] == 0)return(0);**

 **}**

 **}**

 **return(1);**

**}**