Cellular Automaton Graphics(7)

Morio Kikuchi

Abstract :

Developing a regular polyhedron on a plane, setting discrete coordinates on the development and applying a boundary condition of regular polyhedron to it, we realize a symmetrical graphics.

1. Catalan solid

Some of Catalan solids which are duality of semi-regular polyhedron seem to have a shape which is got by replacing a regular polygon which is an element of regular polyhedron with a pyramid which consists of triangles by the looks. However, the triangle is not regular triangle. The polyhedron using pyramid which consists of regular triangles is already argued in the following times:

- 2nd time : Figure 7, 8
- 4th time : Figure 9, 10

This time, we replace a square which is an element of hexahedron with a pyramid which consists of regular triangles. Figure 1 is the polyhedron.

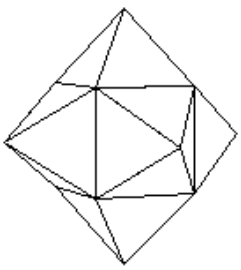

Figure 1

Vertexes is classified in two by the number of triangles around it.

- 6 triangles : 8 vertexes
- 4 triangles : 6 vertexes

The former are vertexes of original hexahedron and the latter are vertexes of pyramids.

2. Neighborhood

Figure 2 is the development of the polyhedron of Figure 1. In the figure, H1〜H6 represent pyramids and 0〜3, 0′〜3′ represent vertexes of two facing squares of original hexahedron. A side agrees with a side if they own two end points jointly.

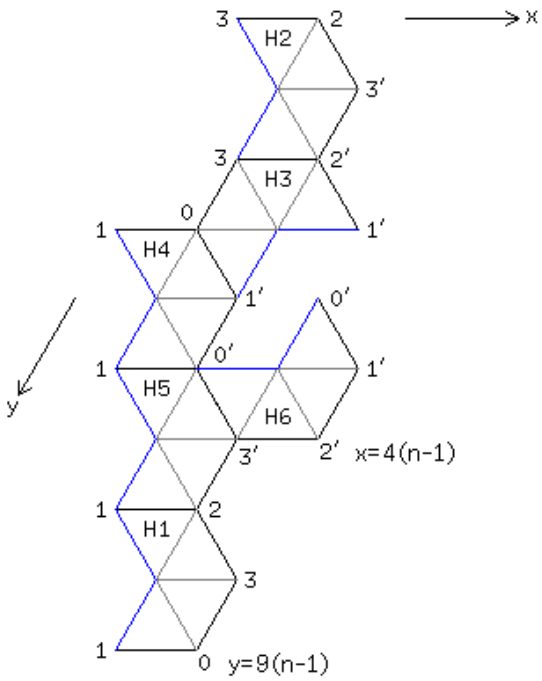

Figure 2

v?, s?, c? which appear in the following are signs in Figure 4 in 4th time. The neighborhood at vertex H1-v3 is

・60, 120, 180(H1-v3)+240, 300, 360(H4-v1)

The number is angle(deg.) of direction to which a pixel exists.
The neighborhood at vertex H4-v0 is

・300, 360(H4-v0)+300, 360(H1-v0)+300, 360(H5-v0)

The pair of blue lines in the figure is cut which was described in 4th time. The two blue lines in each pyramid agree. The neighborhood in the joint endpoint is, for example in H2

・240, 300, 360, 60, 120

However, in the joint endpoint, because there is no jump, program handles it appropriately by a general routine. Therefore, special coding on the joint endpoint is not required.
The neighborhood at side H1-s1 is

・120, 180, 240, 300(H1-s1)+240, 300(H2-s0)

The neighborhood at cut H2-c0 is

・300, 360, 60, 120(H2-c0)+300, 360(H2-c4)

3. Connection of 2 figures

We connect two polyhedrons in Figure 1. The development is Figure 3.

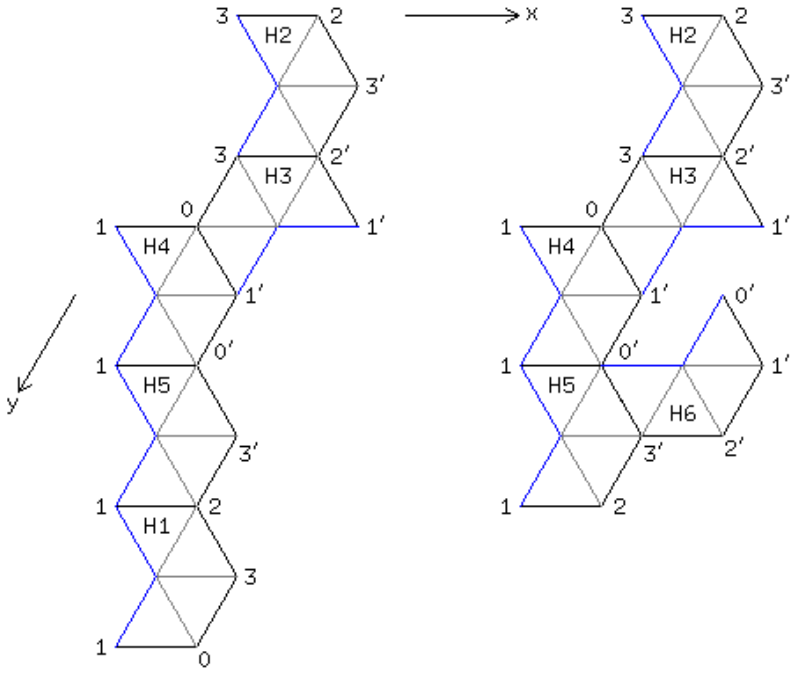

Figure 3

This time, on the connection, differing the past way, the agreement of vertexes, sides of each polyhedron falls into disorder. In the left figure, H6 is removed and in the right figure, H1 is removed. There is no visually clear hole which was used in the last and the last but one and a hole exists as a polygonal line. The polygonal line is represented like the following using vertex numbers:

・0′-1′-2′-3′　(left figure)
・1-0-3-2　(right figure)

The agreement of vertexes obeys the above term order like the following:

・0′ ⇔1, 1′ ⇔0, 2′ ⇔3, 3′ ⇔2

A side agrees with a side if they own two end points jointly like the following:

・0′-1′ ⇔1-0, 1′-2′ ⇔0-3, 2′-3′ ⇔3-2, 3′-0′ ⇔2-1

The neighborhood at vertex H5-v1 in the left figure is

・60, 120, 180, 240, 300(left：H5-v1)+300, 360, 60(right：H5-v0)

The neighborhood at vertex H5-v0 in the right figure is

・300, 360, 60(right：H5-v0)+60, 120, 180, 240, 300(left：H5-v1)

4. Connection of 3 figures

We connect three polyhedrons in Figure 1. The development is Figure 4.

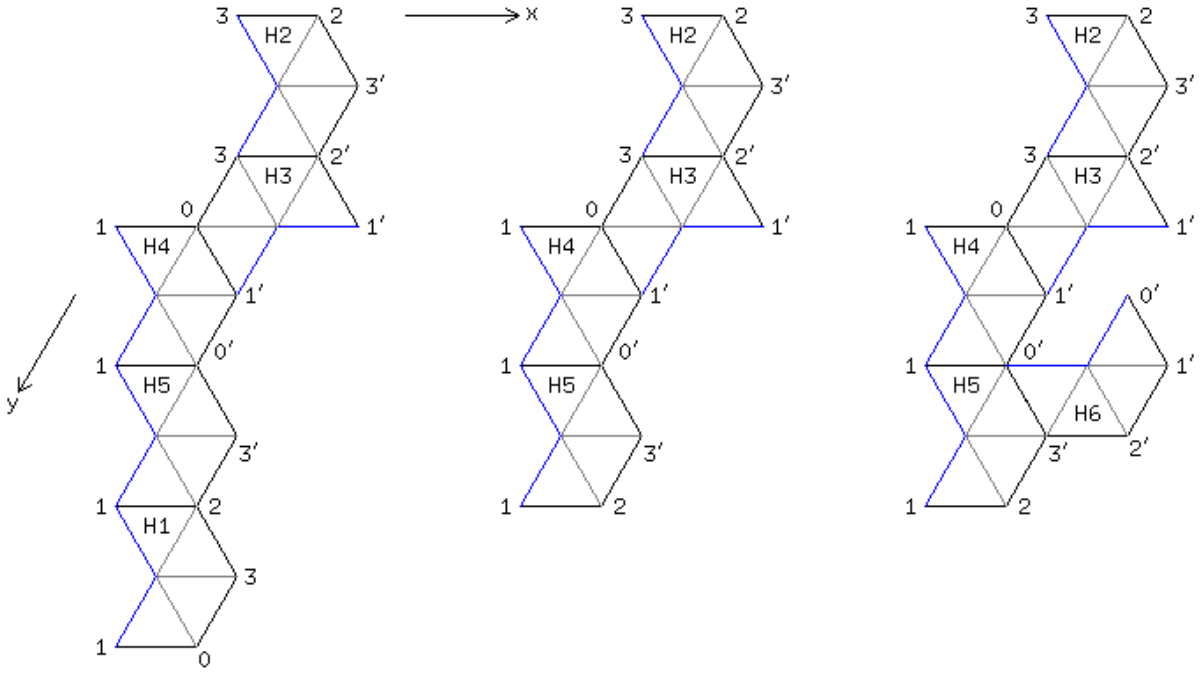

Figure 4

In the left figure, H6 is removed, in the centre figure, H1 and H6 are removed, and in the right figure, H1 is removed. The relation between the left figure and the centre figure is "3. Connection of 2 figures" and the relation between the centre figure and the right figure is also it .

5. Odd tunnel

This time, on the realization of odd tunnel, the setting is in disorder a little. The number of seed points is 4 or 8 and they are divided into H1 and H6 equally. If it is 4, we distinguish cross, parallel like the last Figure 7-(a), (b). Besides, close=1 means that all vertexes must be closed.

- number of seed points=8, close=1, 2 figures connection
- number of seed points=4, cross=0, no connection or 3 figures connection
- number of seed points=4, cross=1, 2 figures connection

6. Assignment

- Complete neighborhoods on vertexes and sides of Figure 2.
- Complete neighborhoods on vertexes and sides of Figure 3.
- Get neighborhoods on vertexes and sides of Figure 4.

The views are angle representation.

7. Modification of the past program

We modify cag_5.c in the 5th time.

```
#define CPMAX /*3*//*6*/6
#define CPHALF (CPMAX/2)
```

CPMAX/2 is bad if CPMAX=3. Modify it like the following:

```
#define CPMAX /*3*//*6*/6
#if CPMAX==6
#define CPHALF (CPMAX/2)
#else
#define CPHALF (CPMAX)
#endif
```

8. Concrete example

Figure 5 is a symmetrical graphics by Figure 4 and the following are data of program.

・SP(2×2):2(left H1)+2(right H6)
・cross=0
・$n = 6$
・coordinates of painting number 1：painting point a:($4(n-1)-1$, $9(n-1)-1$), painting point b:($3(n-1)$, $7(n-1)+1$)
・coordinates of painting number 1：painting point c:($14(n-1)-1$, $5(n-1)$), painting point d:($13(n-1)$, $6(n-1)-1$)
・coordinates of painting number 2：painting point a:$\Delta y = -1$, painting point b:$\Delta x = -1$
・coordinates of painting number 2：painting point c:$\Delta x = -1$;$\Delta y = -1$, painting point d:$\Delta x = 1$

If painting number 2 is finished at the first graphics, program pauses. Press Esc key.

・choice of CW, CCW：the same as the first
・painting algorithm：logical angle method
・painting timing：immediate painting
・push to stack：the same as the first

Array which is used for painting is initialized as follows:

・target pixel：15
・wall pixel：0

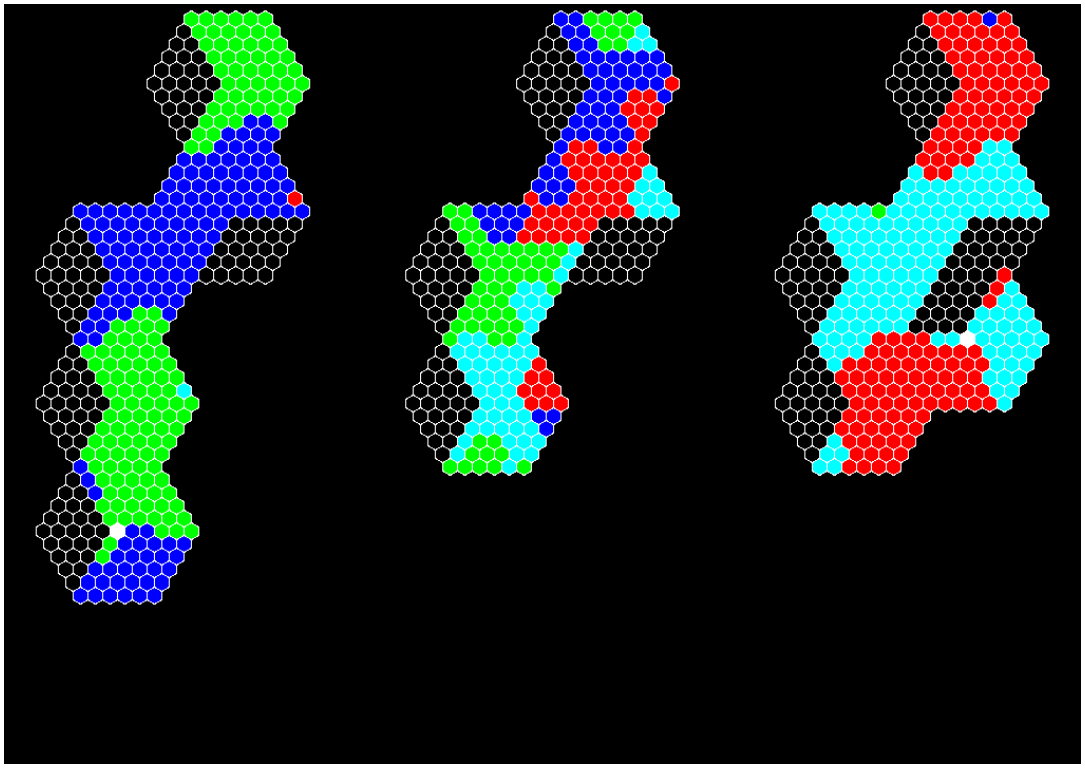

Figure 5

****

セルラーオートマトングラフィクス (7)

菊池盛雄

アブストラクト：
　正多面体を平面上に展開し、この展開図形に離散座標を設定し、正多面体の境界条件を適用して対称なグラフィクスを実現します。

1. カタランの立体

　半正多面体の双対であるカタランの立体の一部は外見的には正多面体の要素である正多角形を三角形から成る角錐で置き換えた形になっています。この三角形は正三角形ではありません。正三角形から成る角錐を用いた多面体は以下の回において既に議論しました。

・第 2 回：図 7、8
・第 4 回：図 9、10

今回は正六面体の要素である正方形を正三角形から成る角錐で置き換えます。図 1 はその多面体です。

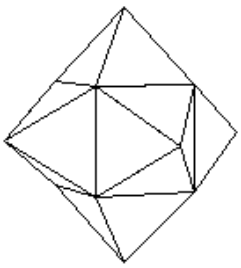

図 1

頂点はその回りに何個の正三角形があるかで二つに分類されます。

・6 個の正三角形：8 個の頂点
・4 個の正三角形：6 個の頂点

前者は元々の正六面体の頂点であり、後者は角錐の頂点です。

2. 近傍

　図 2 は図 1 の多面体の展開図です。図において、H1～H6 は角錐を表し、0～3 と 0′～3′ は元々の正六面体の正対する正方形の頂点を表します。辺は両端点が同じである辺と一致します。

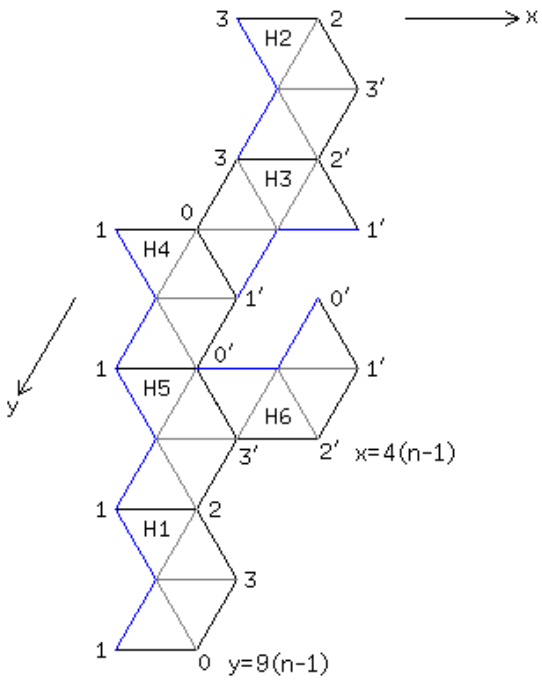

図 2

以下で現れる v?、s?、c?は第 4 回の図 4 における記号です。頂点 H1-v3 における近傍は

・60, 120, 180(H1-v3)+240, 300, 360(H4-v1)

数字はピクセルの存在する方向の角度です。
頂点 H4-v0 における近傍は

・300, 360(H4-v0)+300, 360(H1-v0)+300, 360(H5-v0)

図中の青い直線は第 4 回で述べたカットです。各角錐におけるカットは一致します。その共有端点における近傍は、たとえば H2 では

・240, 300, 360, 60, 120

ただし、共有端点においてはジャンプがないのでプログラムが一般的なルーチンで適正に処理します。したがって、共有端点に関する特別な記述は不要です。
辺 H1-s1 における近傍は

・120, 180, 240, 300(H1-s1)+240, 300(H2-s0)

カット H2-c0 における近傍は

・300, 360, 60, 120(H2-c0)+300, 360(H2-c4)

3. 2 連結

図 1 の多面体を 2 連結します。展開図は図 3 のようになります。

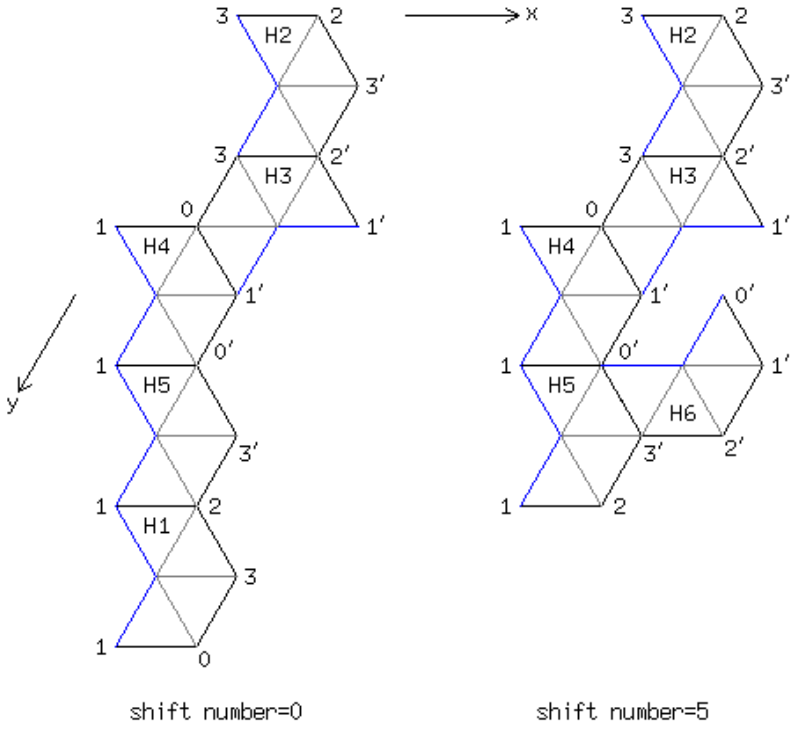

図 3

今回の連結は今までのそれとは異なり、各々の多面体の頂点、辺の一致が乱れます。左図では H6 が除去され、右図では H1 が除去されます。前回までの視覚的に明確な穴は存在せず、穴は折れ線として存在します。この折れ線は頂点番号を用いると以下のように表されます

・0′-1′-2′-3′　(左図)
・1-0-3-2　(右図)

頂点の一致は以下のように項の順序に従います。

・0′ ⇔1, 1′ ⇔0, 2′ ⇔3, 3′ ⇔2

辺は以下のように両端点が同じである辺と一致します。

・0′-1′ ⇔1-0, 1′-2′ ⇔0-3, 2′-3′ ⇔3-2, 3′-0′ ⇔2-1

左図の頂点 H5-v1 における近傍は

・60, 120, 180, 240, 300(左図：H5-v1)+300, 360, 60(右図：H5-v0)

右図の頂点 H5-v0 における近傍は

・300, 360, 60(右図：H5-v0)+60, 120, 180, 240, 300(左図：H5-v1)

4. 3 連結

図 1 の多面体を 3 連結します。展開図は図 4 のようになります。

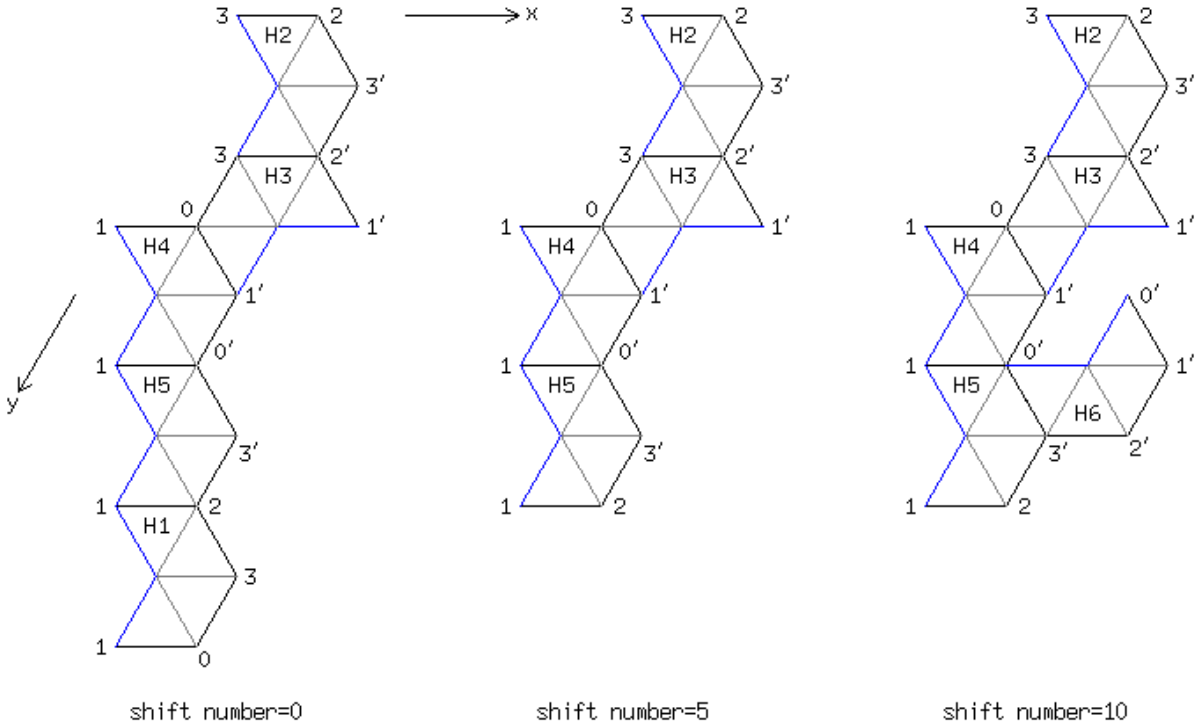

図 4

左図では H6 が、中図では H1 と H6 が、右図では H1 が除去されています。左図と中図の関係は”3. 2 連結”であり、中図と右図の関係も”3. 2 連結”です。

5. 奇トンネル

今回においては奇トンネルを実現する設定は多少雑然としています。シードポイントは 4 個または 8 個で、H1 と H6 に二分割します。シードポイントが 4 個の場合は、前回の図 7-(a), (b) のように cross、parallel を区別します。また、close が 1 ならば全頂点を必ず閉じます。

・シードポイント=8 個、close=1、2 連結
・シードポイント=4 個、cross=0、無連結または 3 連結
・シードポイント=4 個、cross=1、2 連結

6. 課題

・図 2 の頂点、辺、カットにおける近傍 (角度表現) を完成させてください。
・図 3 の頂点、辺、カットにおける近傍 (角度表現) を完成させてください。
・図 4 の頂点、辺、カットにおける近傍 (角度表現) を求めてください。

7. 過去分の修正

第 5 回の cag_5.c を修正します。

```
#define CPMAX /*3*//*6*/6
#define CPHALF (CPMAX/2)
```

CPMAX/2 は CPMAX=3 においては不正です。以下のように修正してください。

```
#define CPMAX /*3*//*6*/6
#if CPMAX==6
#define CPHALF (CPMAX/2)
```

```
#else
#define CPHALF (CPMAX)
#endif
```

8. 具体例

図 5 は図 4 による対称グラフィクスであり、以下はプログラムのデータです。

・SP(2×2):2(left H1)+2(right H6)
・cross=0
・$n = 6$
・塗番号 1 の座標：塗点 a:$(4(n-1)-1,\ 9(n-1)-1)$、塗点 b:$(3(n-1),\ 7(n-1)+1)$
・塗番号 1 の座標：塗点 c:$(14(n-1)-1,\ 5(n-1))$、塗点 d:$(13(n-1),\ 6(n-1)-1)$
・塗番号 2 の座標：塗点 a:$\Delta y = -1$、塗点 b:$\Delta x = -1$
・塗番号 2 の座標：塗点 c:$\Delta x = -1; \Delta y = -1$、塗点 d:$\Delta x = 1$

最初のグラフィクスで塗番号 2 が終了するとプログラムが一時停止します。Esc キーを押してください。

・CW、CCW の選択：初回と同じ
・塗りつぶしアルゴリズム：論理角度法
・塗り方：即時塗りつぶし
・スタックへの座標のプッシュ：初回と同じ

塗りつぶしに用いられる配列は以下のように初期化します。

・ターゲットピクセル：15
・壁ピクセル：0

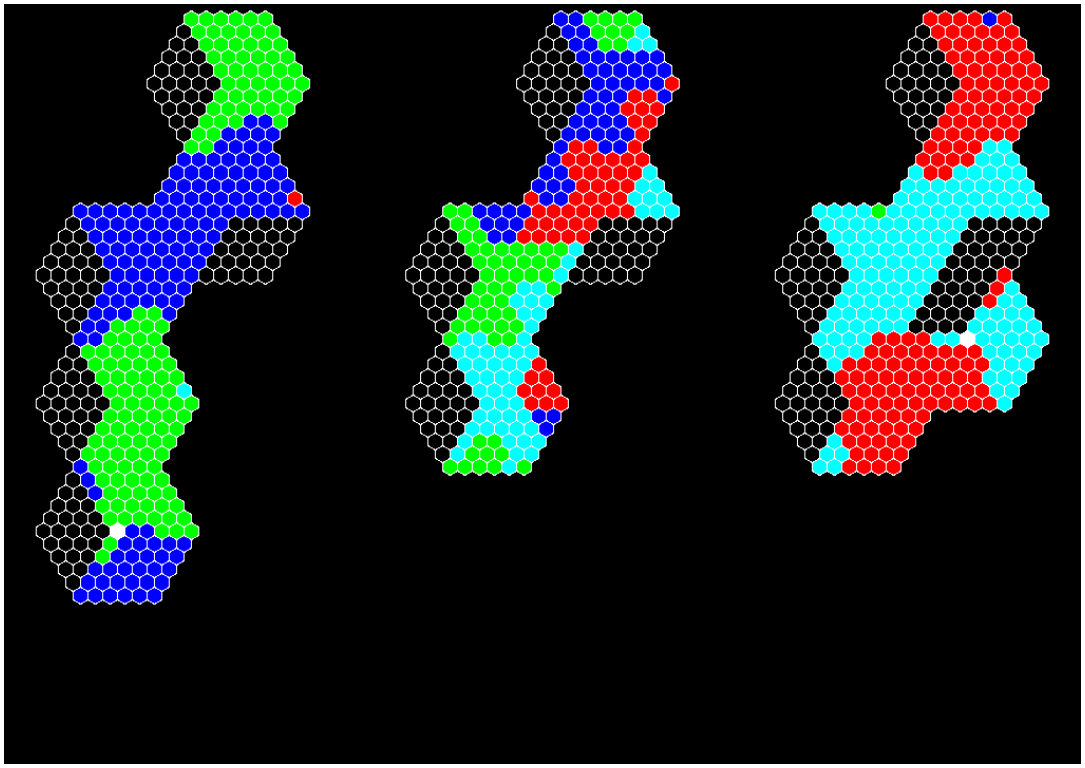

図 5

****

List 1:cag_7.c

```
/* t2.31                                                                    */
/* 2019 Morio Kikuchi                                                       */

#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

#define VGACOLORS (16+8)
#define GKS GetKeyState
#define ASIZE_MS ((1024*768)/CPMAX)  /* memory stack */
#define CPMAX /*4*//*8*/4
#define RESO /*6*//*8*//*10*/6                    /* n */

#if CPMAX==8
#define DIV 1
#define CROSS -1
#define FNUM 2
#define CLOSE 1
#else

#define DIV /*0*//*1*/1
#define CROSS /*0*//*1*/0
#if DIV==1 && CROSS==0
#define FNUM /*1*//*3*/3
#elif DIV==1 && CROSS==1
#define FNUM /*2*/2
#else
#define FNUM /*1*//*2*//*3*/3
#endif

#define CLOSE 0
#endif

#if DIV==0
#define CPHALF CPMAX
#else
#define CPHALF (CPMAX/2)
#endif

#define X0 (170)
```

```
#define Y0 (10)
#define dyMAX 603
#define GRPH_0_MAX 10000

#define ICEIL(a,b) (((a)+((b)-1))/(b))

char refill,pauseflag,fieldflag,GRPH;
char charcode,charflag;
int X,Y,X_,Y_,Nx,Ny,h1[3],h6[3];
int enX[6+6],enY[6+6],enX_[6+6],enY_[6+6],enSN[6+6];
int algo,combination,drn,ig,PIXSIZE,dy_hex,jmpflag,sn_,sn;
int xt,yt,ssize,std_x,std_y,last_x,last_y,tmp0,tmp1;
long asize=ASIZE_MS;
long rcount[CPMAX],cnt;

/*unsigned */char **pixel;
long fp_mem[CPMAX];

char function,usflag;
unsigned char yorn;
int XRESO,YRESO,WB,DX_FRAME,DY_FRAME,DY_CAPTION;
FILE *fp;

typedef struct {int xx,yy,xx_,yy_,sn;} ss;
ss s;ss rtn[CPMAX][ASIZE_MS];
typedef struct {
unsigned char red,green,blue;} srgb;
srgb irgb[VGACOLORS];
typedef struct {
unsigned long back_;int back,fore;} bf;
bf bfset[]={{WHITENESS,15,0},{BLACKNESS,0,15}};

HINSTANCE hinstance;
HWND hwnd;
HDC hdcdisplay,hdctmp1;
HBITMAP hbitmap1;
HPEN hpen;
HBRUSH hbrush;

void closegraph_(void),initpalette(void),BitBlt_full(void),setup(void),
     cleardevice_(char,int,int,int,int),rectangle_(int,int,int,int),
     delay_(long),beep(long),kbhit_(void),restore_3(void),initgraph_return(void),
     use_subroop(void),keydowns_f2(void),bitblt(char,int,int,int,int,int,int),
     arrayreset(void),fwrite_mem(int),fread_mem(int),putpixel_(int,int,int),
     field(int,int,int),check_rcount(void),putdelta(int,int,int,int,int,int);
unsigned char subroop(void);
```

```
int initgraph_(void),setup_(void),fourfloor_fiveceil(double),random_(int),
    getpixel_(int,int,int,int),cag_r(void);
long ftell_mem(int);

COLORREF PALETTE(int color);
LRESULT CALLBACK wndproc_by_kbhit_(HWND,UINT,WPARAM,LPARAM);
int wndproc_filer(HWND,UINT,WPARAM,LPARAM);



int main(int argc,unsigned char **argv)
{
long mytime;

if(argc>1 && strcmp(argv[1],"0")==0){
GRPH=0;
if(argc==2) argc=1;else argc=2;
}
else GRPH=1;
WB=1;
refill=1;

if(initgraph_()==1) return 1;

cleardevice_(1,0,0,XRESO,YRESO);
BitBlt_full();

cnt=-1;
field(1,1,10);
field(1,1,5);
field(1,1,0);
cnt=0;

/*xt=4*(RESO-1);
yt=9*(RESO-1);*/
printf(" xt=%d yt=%d\n",xt/(RESO-1),yt/(RESO-1));
if(setup_()==1) return 1;
arrayreset();

if(argc>1) {time(&mytime);srand((unsigned int)mytime);}
else
srand(1);

combination=1;
drn=4;

while(1){
```

```
/*if(cnt==281) GRPH=1;*/
            field(0,1,10);
if(FNUM==3) field(0,0,5);
else        field(0,1,5);
if(FNUM==1) field(1,1,0);
else        field(1,0,0);
cag_r();
if(refill==0) break;
check_rcount();
printf(" \n");
if(refill==0) break;

if(GRPH){
beep(50);

delay_(6000);
if(pauseflag==1) {pauseflag=0;use_subroop();}
}/**if(GRPH)**/
if(refill==0) break;
/*if(cnt==3) GRPH=1;*/
}/**while(1)**/

closegraph_();

return 0;
}/** main **/



void ls_image(char flag,char *file,int x,int y,int dx,int dy)
{
unsigned long xsize,ysize,size;
unsigned long width,height,imagesize;
unsigned long bits,bytesPerPixel,lineSizeDW,lineSize;
HDC hdce,hdc;
HBITMAP hbitmape;
BITMAPFILEHEADER bfh;
BITMAPINFOHEADER bih;
BYTE *gdata;
FILE *fpo,*fpi;

if(flag<=3){                        /* save */
if(flag==0){
}
else if(flag==1){
}
else if(flag==2){
```

```
}
else if(flag==3){
}
else return;

if((fpo=fopen(file,"wb"))==NULL) {printf("Can't open a file.\n");return;}

width=dx;
height=dy;

bits=/*16*/24/*32*/;
bytesPerPixel=bits/8;
lineSizeDW=bytesPerPixel*width;
lineSizeDW=ICEIL(lineSizeDW,sizeof(long));
lineSize=lineSizeDW*sizeof(long);
imagesize=lineSize*height;

bfh.bfType=0x4d42;                    /* "BM" */
bfh.bfSize=54+imagesize;
bfh.bfReserved1=0;
bfh.bfOffBits=54;
bfh.bfReserved2=0;

bih.biSize=40;
bih.biWidth=width;
bih.biHeight=height;
bih.biPlanes=1;
bih.biBitCount=bits;
bih.biCompression=0;
bih.biSizeImage=imagesize;
bih.biXPelsPerMeter=0;
bih.biYPelsPerMeter=0;
bih.biClrUsed=0;
bih.biClrImportant=0;

if(flag<=1)
/*hdce=CreateCompatibleDC(hdctmp2)*/;
else if(flag==2)
hdce=CreateCompatibleDC(hdctmp1);
else{
hdc=CreateDC("DISPLAY",NULL,NULL,NULL);
hdce=CreateCompatibleDC(hdc);
}

hbitmape=CreateDIBSection(hdce,(LPBITMAPINFO)&bih,DIB_RGB_COLORS,&gdata,NULL,0);
SelectObject(hdce,hbitmape);
```

```
if(flag<=1)
/*BitBlt(hdce,0,0,dx,dy,hdctmp2,x,y,SRCCOPY)*/;
else if(flag==2)
BitBlt(hdce,0,0,dx,dy,hdctmp1,x,y,SRCCOPY);
else
BitBlt(hdce,0,0,dx,dy,hdc,x,y,SRCCOPY);

size=bih.biSizeImage;

fwrite(&bfh,14,1,fpo);
fwrite(&bih,40,1,fpo);
fwrite(gdata,size,1,fpo);

fclose(fpo);

if(flag==3) DeleteDC(hdc);
DeleteDC(hdce);
DeleteObject(hbitmape);
}
else{                                /* load */
if((fpi=fopen(file,"rb"))==NULL) {printf("Can't open the file.\n");return;}

fread(&bfh,14,1,fpi);
if(bfh.bfType!=0x4d42) {fclose(fpi);printf("Not BM.\n");return;}
fread(&bih,40,1,fpi);

fseek(fpi,bfh.bfOffBits,0);
size=bih.biSizeImage;
gdata=(BYTE *)malloc(size);
fread(gdata,size,1,fpi);

/*StretchDIBits(hdctmp2,x,y,bih.biWidth,bih.biHeight,0,0,bih.biWidth,bih.biHeight,
              gdata,(LPBITMAPINFO)&bih,DIB_RGB_COLORS,SRCCOPY);*/

fclose(fpi);
free(gdata);
}
}/** ls_image **/


void fprintf_(long v1,long v2,long v3,long v4,long v5)
{
FILE *fp;

fp=fopen("cpage.bin","ab");
```

```
fprintf(fp," %ld %ld %ld %ld %ld\n",v1,v2,v3,v4,v5);

fclose(fp);
}/** fprintf_ **/


void use_subroop(void)
{
char function_old,charflag_old;

usflag=1;

function_old=function;function=2;
charflag_old=charflag;

yorn=subroop();

function=function_old;
charflag=charflag_old;
}/** use_subroop **/


unsigned char subroop(void)
{
charflag=1;

while(1){
kbhit_();
if(charflag==0) return charcode;
}
}/** subroop **/


void keydowns_f2(void)
{
int dy;

if(GKS(VK_ESCAPE)<0 || GKS(VK_PAUSE)<0) charflag=0;
else if(GKS('S')<0){
dy=Y0+(yt+1/*1.7*/)*(sqrt(3)/2)*PIXSIZE+10;
ls_image(2,"ss.bmp",0,0,XRESO,/*YRESO*/dy);
/*printf(" %d\n",dy);*/
beep(300);
}
}/** keydowns_f2 **/
```

```
void restore_in_PAINT(void)
{
ValidateRect(hwnd,NULL);

bitblt(1,0,0,XRESO,YRESO,0,0);
}/** restore_in_PAINT **/


void setup(void)
{
XRESO=1024-4;YRESO=768-24*2;
}/** setup **/


int get_dx(int nx,int ny)
{
int dx;

dx=ff_fc(X0+nx*1.0*PIXSIZE-ny*0.5*PIXSIZE);

return dx;
}/** get_dx **/


int get_dy(int nx,int ny)
{
int dy;

dy=ff_fc(Y0+ny*(sqrt(3)/2)*PIXSIZE);

return dy;
}/** get_dy **/


int setup_(void)
{
int i,dy;

PIXSIZE=15;
dy=Y0+(yt+1/*1.7*/)*(sqrt(3)/2)*PIXSIZE+10;

if(dy>dyMAX){
while(1){
PIXSIZE--;
```

```
dy=Y0+(yt+1/*1.7*/)*(sqrt(3)/2)*PIXSIZE+10;
if(dy<=dyMAX) break;
}
}

if(PIXSIZE<4) PIXSIZE=4;

dy=get_dy(1,1)-get_dy(0,0);
dy_hex=ff_fc(dy/3.);

pixel=(/*unsigned */char **)malloc(sizeof(/*unsigned */char *)*((xt+1)+1));
if(pixel==NULL){
DeleteDC(hdctmp1);
DeleteObject(hbitmap1);
initgraph_return();return 1;}

i=0;
while(1){
pixel[i]=(/*unsigned */char *)malloc(sizeof(/*unsigned */char)*((yt+1)+1));

if(pixel[i]==NULL){
while(1){
i--;
if(i<0) break;
free(pixel[i]);
}
free(pixel);
DeleteDC(hdctmp1);
DeleteObject(hbitmap1);
initgraph_return();return 1;}

i++;
if(i==(xt+1)+1) break;
}

return 0;
}/** setup_ **/


int initgraph_(void)
{
int i,width,height;
WNDCLASS wndclass;

setup();
```

```
wndclass.hInstance    =hinstance;
wndclass.lpszClassName="CAGCLASS";
wndclass.lpszMenuName =NULL;
wndclass.lpfnWndProc  =wndproc_by_kbhit_;
wndclass.style        =0;
wndclass.hIcon        =LoadIcon(hinstance,"MYICON");
wndclass.hCursor      =LoadCursor(NULL,IDC_ARROW);
wndclass.cbClsExtra   =0;
wndclass.cbWndExtra   =0;
if(WB==0)
wndclass.hbrBackground=GetStockObject(WHITE_BRUSH);
else
wndclass.hbrBackground=GetStockObject(BLACK_BRUSH);

if(RegisterClass(&wndclass)==0) return 1;

hwnd=CreateWindow("CAGCLASS"," CAG",
                  /*WS_POPUP,*/
                  WS_OVERLAPPED | WS_CAPTION | WS_SYSMENU | WS_MINIMIZEBOX,
                  0,0,XRESO+DX_FRAME,YRESO+DY_CAPTION+DY_FRAME,
                  NULL,NULL,hinstance,NULL);
if(hwnd==NULL) {MessageBox(NULL,"Memory space is not left.","CAG",MB_OK);return 1;}

SetWindowPos(hwnd,HWND_TOP,0,0,0,0,SWP_NOMOVE | SWP_NOSIZE);
ShowWindow(hwnd,SW_SHOWDEFAULT);

hdcdisplay=GetDC(hwnd);

hbitmap1=CreateCompatibleBitmap(hdcdisplay,XRESO,YRESO);
hdctmp1=CreateCompatibleDC(hdcdisplay);  /* text, dialog, menu */
SelectObject(hdctmp1,hbitmap1);
SetBkMode(hdcdisplay,TRANSPARENT);
SetBkMode(hdctmp1,TRANSPARENT);

initpalette();

SetBkColor(hdcdisplay,PALETTE(bfset[WB].back));
SetBkColor(hdctmp1,PALETTE(bfset[WB].back));

return 0;
}/** initgraph_ **/


void initgraph_return(void)
{
/*EndPaint(hwnd,&paintstruct);*/
```

```
ReleaseDC(hwnd,hdcdisplay);
DestroyWindow(hwnd);
/*UnregisterClass("CAGCLASS",hinstance);*/

MessageBox(NULL,"Memory space is not left.","CAG",MB_OK);
}/** initgraph_return **/


void closegraph_(void)
{
int i;

i=0;
while(1){
free(pixel[i]);
i++;
if(i==(xt+1)+1) break;
}
free(pixel);

DeleteDC(hdctmp1);
DeleteObject(hbitmap1);

/*EndPaint(hwnd,&paintstruct);*/
ReleaseDC(hwnd,hdcdisplay);
DestroyWindow(hwnd);
/*UnregisterClass("CAGCLASS",hinstance);*/
}/** closegraph_ **/


void initpalette(void)
{
int i;

irgb[0].red=0;irgb[0].green=0;irgb[0].blue=0;

irgb[9].red=0;irgb[9].green=0;irgb[9].blue=255;  /* blue */
irgb[10].red=0;irgb[10].green=255;irgb[10].blue=0;  /* green */
irgb[11].red=0;irgb[11].green=255;irgb[11].blue=255;  /* cyan */
irgb[12].red=255;irgb[12].green=0;irgb[12].blue=0;  /* red */
irgb[13].red=255;irgb[13].green=0;irgb[13].blue=255;  /* magenta */
irgb[14].red=255;irgb[14].green=255;irgb[14].blue=0;  /* yellow */

irgb[15].red=255;irgb[15].green=255;irgb[15].blue=255;

for(i=1;i<7;i++){                   /* 1 -> 6 */
```

```
if(irgb[i+8].red==255)
irgb[i].red=127+64;else irgb[i].red=0;
if(irgb[i+8].green==255)
irgb[i].green=127+64;else irgb[i].green=0;
if(irgb[i+8].blue==255)
irgb[i].blue=127+64;else irgb[i].blue=0;
}

for(i=7;i<9;i++){                    /* 7, 8 */
irgb[i].red=127+32*(8-i);
irgb[i].green=irgb[i].red;
irgb[i].blue=irgb[i].red;
}

for(i=16;i<16+8;i++){                    /* 16+8 colors */
irgb[i].red=255*(8.-(i-16))/9;
irgb[i].green=irgb[i].red;
irgb[i].blue=irgb[i].red;
}
}/** initpalette **/


void BitBlt_full(void)
{
bitblt(1,0,0,XRESO,YRESO,0,0);
}/** BitBlt_full **/


void bitblt(char flag,int x,int y,int xsize,int ysize,int x_,int y_)
{
BitBlt(hdcdisplay,x_,y_,xsize,ysize,
       hdctmp1,x,y,SRCCOPY);
}/** bitblt **/


void cleardevice_(char flag,int x,int y,int xsize,int ysize)
{
PatBlt(hdctmp1,x,y,xsize,ysize,bfset[WB].back_);
}/** cleardevice_ **/


COLORREF PALETTE(int color)
{
return RGB(irgb[color].red,irgb[color].green,irgb[color].blue);
}/** PALETTE **/
```

```
void kbhit_(void)
{
MSG msg;

if(PeekMessage(&msg,NULL,0,0,PM_REMOVE)){
TranslateMessage(&msg);
DispatchMessage(&msg);
}
}/** kbhit_ */


LRESULT CALLBACK  wndproc_by_kbhit_(HWND hwnd,UINT umsg,WPARAM wparam,LPARAM lparam)
{
if(wndproc_filer(hwnd,umsg,wparam,lparam)!=0) return 1;

return DefWindowProc(hwnd,umsg,wparam,lparam);
}/** wndproc_by_kbhit_ **/


int wndproc_filer(HWND hwnd,UINT umsg,WPARAM wparam,LPARAM lparam)
{
if(umsg==WM_KEYDOWN){
/************************** menu keydowns -> ****************************/
/************************** <- menu keydowns ****************************/

/************************** dialog keydowns -> ****************************/
/************************** <- dialog keydowns ****************************/

if(function==2){
keydowns_f2();
return 1;
}

if(usflag==1) usflag=0;

     if(GKS(VK_ESCAPE)<0 || GKS(VK_PAUSE)<0) refill=0;
else if(GKS(VK_SHIFT)<0) pauseflag=1;

return 1;
}/**else if(umsg)**/
else if(umsg==WM_SYSKEYDOWN){
}/**else if(umsg)**/
else if(umsg==WM_CLOSE){
if(function==2) charflag=0;
else refill=0;
```

```
return 1;
}/**else if(umsg)**/
else if(umsg==WM_PAINT){
restore_in_PAINT();

return 1;
}/**else if(umsg)**/
else{}

return 0;
}/** wndproc_filer **/


void delay_(long millisecond)
{
long oldtime,nowtime,dtime;
double i=CLOCKS_PER_SEC,j;

j=millisecond;
millisecond=j*(i/1000.);
oldtime=clock();

while(1){
kbhit_();
if(pauseflag==1 && refill==0) {pauseflag=0;refill=1;break;}
if(refill==0) break;

nowtime=clock();dtime=nowtime-oldtime;
if(dtime>=millisecond) break;
if(dtime<0) break;
}
}/** delay_ **/


void beep(long millisecond)
{
Beep(888,millisecond);
}/** beep **/


int fourfloor_fiveceil(double val_d)
{
int val_i,val;

val_i=floor(val_d);
```

```
val=(val_d-val_i<0.5)?val_i:val_i+1;

return val;
}/** fourfloor_fiveceil **/


int ff_fc(double val_d)
{
return fourfloor_fiveceil(val_d);
}/** ff_fc **/


void arrayreset(void)
{
int i,j;

i=0;
while(1){

j=0;
while(1){
pixel[i][j]=0/*1*/;
j++;
if(j==(yt+1)+1) break;
}

i++;
if(i==(xt+1)+1) break;
}
}/** arrayreset **/


int putpixel(int nx,int ny,int pcolor)
{
int i,dx,dy,color,n=RESO;
POINT vertex[7];

if(nx<0 || ny<0) return 1;
if(GRPH==0) goto end;

dx=get_dx(nx,ny);
dy=get_dy(nx,ny);

vertex[0].x=dx;                vertex[0].y=dy;
vertex[2].x=get_dx(nx+1,ny+0);vertex[2].y=get_dy(nx+1,ny+0);
vertex[4].x=get_dx(nx+1,ny+1);vertex[4].y=get_dy(nx+1,ny+1);
```

```
vertex[1].x=vertex[4].x      ;vertex[1].y=vertex[0].y-dy_hex;
vertex[3].x=vertex[2].x      ;vertex[3].y=vertex[4].y-dy_hex;
vertex[5].x=vertex[0].x      ;vertex[5].y=vertex[4].y-dy_hex;

vertex[6].x=vertex[0].x;
vertex[6].y=vertex[0].y;

if(pcolor==15)
hpen=CreatePen(PS_SOLID,1,PALETTE(9));
else
hpen=CreatePen(PS_SOLID,1,PALETTE(15));

if(fieldflag==1){
/*if(nx==2*(n-1) && ny==1*(n-1)) color=4;
else if(nx==3*(n-1) && ny==6*(n-1)) color=4;
else if(nx==3*(n-1) && ny==3*(n-1)) color=12;
else if(nx==2*(n-1) && ny==4*(n-1)) color=12;
else if(nx==(5+1)*(n-1) && ny==0*(n-1)) color=4;
else if(nx==(5+3)*(n-1) && ny==7*(n-1)) color=4;
else if(nx==(5+1)*(n-1) && ny==3*(n-1)) color=12;

else if(nx==2*(n-1) && ny==5*(n-1)) color=14;
else if(nx==2*(n-1) && ny==2*(n-1)) color=6;
else if(nx==(5+0)*(n-1) && ny==3*(n-1)) color=14;
else if(nx==(5+1)*(n-1) && ny==5*(n-1)) color=14;
else if(nx==(5+2)*(n-1) && ny==7*(n-1)) color=14;
else if(nx==(5+0)*(n-1) && ny==0*(n-1)) color=6;
else if(nx==(5+1)*(n-1) && ny==2*(n-1)) color=6;

else */color=pcolor;
}
else{
color=pcolor;
}
hbrush=CreateSolidBrush(PALETTE(color));

SelectObject(hdcdisplay,hpen);
SelectObject(hdcdisplay,hbrush);

Polyline(hdcdisplay,vertex,6+1);
Polygon(hdcdisplay,vertex,6);

SelectObject(hdctmp1,hpen);
SelectObject(hdctmp1,hbrush);
```

```
Polyline(hdctmp1,vertex,6+1);
Polygon(hdctmp1,vertex,6);

DeleteObject(hbrush);
DeleteObject(hpen);

end:
if(nx<=xt && ny<=yt)
pixel[nx][ny]=pcolor;

return 0;
}/** putpixel **/


void field_hex(int v,int x,int y)
{
int i,j,n,XDP,YDP,begin;

n=RESO;

XDP=x+(n-1);
YDP=y+(n-1);

begin=XDP-(n-1);
for(j=0;j<n;j++)
for(i=0;i<2*n-1-j;i++){
putpixel_(begin+i,YDP-j,tmp0);
if(begin+i>xt) xt=begin+i;
if(YDP-j>yt) yt=YDP-j;
}

begin=XDP+(n-1);
for(j=0;j<n;j++)
for(i=0;i<2*n-1-j;i++){
putpixel_(begin-i,YDP+j,tmp0);
if(begin-i>xt) xt=begin-i;
if(YDP+j>yt) yt=YDP+j;
}

if(v==0){
putdelta(n,0,x,y,0,0);
putdelta(n,1,x,y+(n-1),0,1);
for(i=0;i<=n-2;i++)
putpixel_(x+(n-1)-(1+i),y+(n-1),0);
}
else if(v==4){
```

```
putdelta(n,0,x+(n-1),y+(n-1),1,1);
putdelta(n,1,x+(n-1),y+(n-1),1,1);
for(i=0;i<=n-2;i++)
putpixel_(x+(n-1)+(1+i),y+(n-1)+(1+i),0);
}
else if(v==1){
putdelta(n,0,x,y,1,0);
putdelta(n,1,x,y,0,0);
}
}/** field_hex **/


void putdelta(int n,int lr,int x,int y,int dx,int dy)
{
int i,j;

if(lr==0){
for(j=0;j<n-1;j++)
for(i=0;i<j;i++)
putpixel_(x+dx+i,y+dy+j,0/*6*/);
}
else{
for(j=0;j<n-1;j++)
for(i=0;i<j;i++)
putpixel_(x+dx+(n-2)-i,y+dy+(n-2)-j,0/*6*/);
}
}/** putdelta **/


void check_rcount(void)
{
int i;
long val[2];

if(GRPH>0){
for(i=0;i<CPMAX;i++)
printf(" %ld %ld\n",cnt,rcount[i]);
}
else{
if(/*Odd>0 || CPMAX==5*/1){
val[0]=rcount[0];
for(i=1;i<CPMAX;i++){
if(rcount[i]!=val[0]) {beep(1000);refill=0;break;}
}

if(refill==0)
```

```
printf(" %ld %ld %d:%ld\n",cnt,val[0],i,rcount[i]);
else
printf(" %ld %ld\n",cnt,val[0]);
}/**if(_6dRow, CPMAX)**/
else{
val[0]=rcount[0];
for(i=1;i<CPHALF;i++){
if(rcount[i]!=val[0]) {beep(1000);refill=0;break;}
}

if(refill){
val[1]=rcount[CPHALF];
for(i=CPHALF+1;i<CPMAX;i++){
if(rcount[i]!=val[1]) {beep(1000);refill=0;break;}
}

if(refill)
printf(" %ld %ld %ld\n",cnt,val[0],val[1]);
else
printf(" %ld 1st:%ld 2nd:%ld %d:%ld\n",cnt,val[0],val[1],i,rcount[i]);
}/**if(refill)**/
else{
printf(" %ld 1st:%ld %d:%ld\n",cnt,val[0],i,rcount[i]);
}/**else(refill)**/
}/**else(_6dRow, CPMAX)**/
}

if(GRPH==0 && cnt==GRPH_0_MAX) {beep(100);refill=0;}
}/** check_rcount **/


void field(int h1_,int h6_,int begin)
{
char vflag=CLOSE;
int n,pcolor=8;

fieldflag=1;

n=RESO;
sn=begin;
tmp0=15;
h1[sn/5]=h1_;
h6[sn/5]=h6_;

field_hex(0,(sn+0)*(n-1),0*(n-1));
field_hex(4,(sn+1)*(n-1),2*(n-1));
```

```
field_hex(0,(sn+0)*(n-1),3*(n-1));
field_hex(0,(sn+1)*(n-1),5*(n-1));
if(h6[sn/5]==1){
field_hex(1,(sn+2)*(n-1),4*(n-1));  /* h6 */
if(vflag==1 && cnt>-1){
pp_(3,4,pcolor);
pp_(4,5,pcolor);
pp_(4,6,pcolor);
}
}
if(h1[sn/5]==1){
field_hex(0,(sn+2)*(n-1),7*(n-1));  /* h1 */
if(vflag==1 && cnt>-1){
pp_(4,8,pcolor);
pp_(4,9,pcolor);
pp_(3,9,pcolor);
}
}

if(cnt>-1){
pixel[(sn+3)*(n-1)][8*(n-1)]=-1;
pixel[(sn+3)*(n-1)][5*(n-1)]=-1;
}

if(vflag==1 && cnt>-1){
/* 1st */
if(/*sn==0*/1){
pp_(1,3,pcolor);
pp_(1,0,pcolor);pp_(3,7,pcolor);
}
if(/*sn==5*/1){
pp_(3,3,pcolor);pp_(2,4,pcolor);
pp_(2,1,pcolor);pp_(3,6,pcolor);
}

/* 2nd */
if(/*sn==0*/1){
pp_(0,3,pcolor);pp_(1,5,pcolor);pp_(2,7,pcolor);
pp_(0,0,pcolor);pp_(1,2,pcolor);
}
if(/*sn==5*/1){
pp_(2,5,pcolor);
pp_(2,2,pcolor);
}
}
```

```
fieldflag=0;
}/** field **/


void pp_(int nx,int ny,int pcolor)
{
nx+=sn;
nx*=RESO-1;ny*=RESO-1;
putpixel(nx,ny,pcolor);
}/** pp_ **/


void rot(int pos,int dth)
{
pos=pos%6;
dth=dth%6;

pos+=dth;

if(pos>5) pos-=6;
else if(pos<0) pos+=6;

if(pos==0) {tmp0=1;tmp1=0;}  /* ca1 */
else if(pos==1) {tmp0=1;tmp1=1;}  /* ca5 */
else if(pos==2) {tmp0=0;tmp1=1;}  /* ca2 */
else if(pos==3) {tmp0=-1;tmp1=0;}  /* ca3 */
else if(pos==4) {tmp0=-1;tmp1=-1;}  /* ca7 */
else if(pos==5) {tmp0=0;tmp1=-1;}  /* ca4 */
}/** rot **/


int nh(int x,int y,int nx,int ny,int pos)
{
pos=pos%6;

if(pos==0){
if(nx==x+1 && ny==y) return 1;else return 0;
}
else if(pos==1){
if(nx==x+1 && ny==y+1) return 1;else return 0;
}
else if(pos==2){
if(nx==x && ny==y+1) return 1;else return 0;
}
else if(pos==3){
if(nx==x-1 && ny==y) return 1;else return 0;
```

```
}
else if(pos==4){
if(nx==x-1 && ny==y-1) return 1;else return 0;
}
else if(pos==5){
if(nx==x && ny==y-1) return 1;else return 0;
}
}/** nh **/


int v(int h,int v,int nx,int ny)
{
int n;

n=RESO;

/** 11 **/
if(h==1 && v==3){
if(nx==4*(n-1) && ny==9*(n-1)) return 1;else return 0;
}
else if((h==4 && v==1)||(h==3 && v==5)){
if(nx==1*(n-1) && ny==3*(n-1)) return 1;else return 0;
}
else if(h==1 && v==0){
if(nx==2*(n-1) && ny==7*(n-1)) return 1;else return 0;
}
else if(h==1 && v==4){
if(nx==3*(n-1) && ny==9*(n-1)) return 1;else return 0;
}
else if(h==4 && v==0){
if(nx==0*(n-1) && ny==3*(n-1)) return 1;else return 0;
}
else if(h==4 && v==4){
if(nx==1*(n-1) && ny==5*(n-1)) return 1;else return 0;
}
else if(h==1 && v==1){
if(nx==3*(n-1) && ny==7*(n-1)) return 1;else return 0;
}
else if(h==2 && v==1){
if(nx==1*(n-1) && ny==0*(n-1)) return 1;else return 0;
}
else if(h==1 && v==2){
if(nx==4*(n-1) && ny==8*(n-1)) return 1;else return 0;
}
else if(h==2 && v==0){
if(nx==0*(n-1) && ny==0*(n-1)) return 1;else return 0;
```

```
}
else if(h==2 && v==4){
if(nx==1*(n-1) && ny==2*(n-1)) return 1;else return 0;
}

/** 9 **/
else if(h==6 && v==1){
if(nx==3*(n-1) && ny==4*(n-1)) return 1;else return 0;
}
else if(h==6 && v==5){
if(nx==2*(n-1) && ny==5*(n-1)) return 1;else return 0;
}
else if(h==6 && v==2){
if(nx==4*(n-1) && ny==5*(n-1)) return 1;else return 0;
}
else if(h==3 && v==2){
if(nx==3*(n-1) && ny==3*(n-1)) return 1;else return 0;
}
else if(h==3 && v==4){
if(nx==2*(n-1) && ny==4*(n-1)) return 1;else return 0;
}
else if(h==6 && v==3){
if(nx==4*(n-1) && ny==6*(n-1)) return 1;else return 0;
}
else if((h==3 && v==1)||(h==2 && v==3)){
if(nx==2*(n-1) && ny==2*(n-1)) return 1;else return 0;
}
else if((h==6 && v==4)||(h==5 && v==2)){
if(nx==3*(n-1) && ny==6*(n-1)) return 1;else return 0;
}
else if(h==2 && v==2){
if(nx==2*(n-1) && ny==1*(n-1)) return 1;else return 0;
}

return 0;
}/** v **/


int cut(int h,int cut,int nx,int ny)
{
int n;

n=RESO;

if(h==1 && cut==4){
if(nx==3*(n-1) && ny>=8*(n-1)+1 && ny<=9*(n-1)-1) return 1;else return 0;
```

```
}
else if(h==1 && cut==0){
if(nx>=2*(n-1)+1 && nx<=3*(n-1)-1 && ny==nx+5*(n-1)) return 1;else return 0;
}
else if(h==2 && cut==4){
if(nx==1*(n-1) && ny>=1*(n-1)+1 && ny<=2*(n-1)-1) return 1;else return 0;
}
else if(h==2 && cut==0){
if(nx>=0*(n-1)+1 && nx<=1*(n-1)-1 && ny==nx+0*(n-1)) return 1;else return 0;
}
else if(h==4 && cut==4){
if(nx==1*(n-1) && ny>=4*(n-1)+1 && ny<=5*(n-1)-1) return 1;else return 0;
}
else if(h==4 && cut==0){
if(nx>=0*(n-1)+1 && nx<=1*(n-1)-1 && ny==nx+3*(n-1)) return 1;else return 0;
}
else if(h==5 && cut==4){
if(nx==2*(n-1) && ny>=6*(n-1)+1 && ny<=7*(n-1)-1) return 1;else return 0;
}
else if(h==5 && cut==0){
if(nx>=1*(n-1)+1 && nx<=2*(n-1)-1 && ny==nx+4*(n-1)) return 1;else return 0;
}
else if(h==3 && cut==2){
if(nx>=2*(n-1)+1 && nx<=3*(n-1)-1 && ny==3*(n-1)) return 1;else return 0;
}
else if(h==3 && cut==4){
if(nx==2*(n-1) && ny>=3*(n-1)+1 && ny<=4*(n-1)-1) return 1;else return 0;
}
else if(h==6 && cut==5){
if(nx>=2*(n-1)+1 && nx<=3*(n-1)-1 && ny==5*(n-1)) return 1;else return 0;
}
else if(h==6 && cut==1){
if(nx==3*(n-1) && ny>=4*(n-1)+1 && ny<=5*(n-1)-1) return 1;else return 0;
}

return 0;
}/** cut **/


int side(int h,int side,int nx,int ny)
{
int n;

n=RESO;

if(h==1 && side==2){
```

```
if(nx==4*(n-1) && ny>=8*(n-1)+1 && ny<=9*(n-1)-1) return 1;else return 0;
}
else if(h==3 && side==5){
if(nx==1*(n-1) && ny>=2*(n-1)+1 && ny<=3*(n-1)-1) return 1;else return 0;
}
else if(h==6 && side==2){
if(nx==4*(n-1) && ny>=5*(n-1)+1 && ny<=6*(n-1)-1) return 1;else return 0;
}
else if(h==3 && side==1){
if(nx>=2*(n-1)+1 && nx<=3*(n-1)-1 && ny==nx+0*(n-1)) return 1;else return 0;
}
else if(h==5 && side==2){
if(nx==3*(n-1) && ny>=6*(n-1)+1 && ny<=7*(n-1)-1) return 1;else return 0;
}
else if(h==2 && side==1){
if(nx>=1*(n-1)+1 && nx<=2*(n-1)-1 && ny==nx-1*(n-1)) return 1;else return 0;
}
else if(h==1 && side==3){
if(nx>=3*(n-1)+1 && nx<=4*(n-1)-1 && ny==9*(n-1)) return 1;else return 0;
}
else if(h==4 && side==0){
if(nx>=0*(n-1)+1 && nx<=1*(n-1)-1 && ny==3*(n-1)) return 1;else return 0;
}
else if(h==6 && side==3){
if(nx>=3*(n-1)+1 && nx<=4*(n-1)-1 && ny==6*(n-1)) return 1;else return 0;
}
else if(h==2 && side==2){
if(nx==2*(n-1) && ny>=1*(n-1)+1 && ny<=2*(n-1)-1) return 1;else return 0;
}
else if(h==1 && side==1){
if(nx>=3*(n-1)+1 && nx<=4*(n-1)-1 && ny==nx+4*(n-1)) return 1;else return 0;
}
else if(h==2 && side==0){
if(nx>=0*(n-1)+1 && nx<=1*(n-1)-1 && ny==0*(n-1)) return 1;else return 0;
}
else if(h==6 && side==1){
if(nx>=3*(n-1)+1 && nx<=4*(n-1)-1 && ny==nx+1*(n-1)) return 1;else return 0;
}
else if(h==4 && side==2){
if(nx==2*(n-1) && ny>=4*(n-1)+1 && ny<=5*(n-1)-1) return 1;else return 0;
}

else if(h==5 && side==1){
if(nx>=2*(n-1)+1 && nx<=3*(n-1)-1 && ny==nx+3*(n-1)) return 1;else return 0;
}
else if(h==5 && side==3){
```

```
if(nx>=2*(n-1)+1 && nx<=3*(n-1)-1 && ny==7*(n-1)) return 1;else return 0;
}

return 0;
}/** side **/


void putpixel_(int nx,int ny,int pcolor)
{
int n,dlt,flag,snold;

if(cnt<0) return;
n=RESO;

putpixel(nx,ny,pcolor);
if(fieldflag) return;
rcount[ig]++;

/*return;*/  /* here */
nx-=sn*(n-1);

/* v0 */
if(h1[sn/5]==1 && v(1,3,nx,ny)==1){
pp_(4,9,pcolor);pp_(1,3,pcolor);
}
else if(h1[sn/5]==1 && v(4,1,nx,ny)==1){
pp_(4,9,pcolor);pp_(1,3,pcolor);
}
else if(h1[sn/5]==0 && v(4,1,nx,ny)==1){
pp_(1,3,pcolor);

snold=sn;
if(sn==5) sn=0;
else if(sn==10) sn=5;

pp_(3,3,pcolor);
pp_(1+1,3+1,pcolor);
sn=snold;
}
/* v1 */
else if(h1[sn/5]==1 && v(1,0,nx,ny)==1){
pp_(2,7,pcolor);pp_(3,9,pcolor);
pp_(0,3,pcolor);pp_(1,5,pcolor);
}
else if(h1[sn/5]==0 && v(1,0,nx,ny)==1){
pp_(2,7,pcolor);
```

```
pp_(0,3,pcolor);pp_(1,5,pcolor);

snold=sn;
if(sn==5) sn=0;
else if(sn==10) sn=5;

pp_(2,5,pcolor);
sn=snold;
}
else if(h1[sn/5]==1 && v(1,4,nx,ny)==1){
pp_(2,7,pcolor);pp_(3,9,pcolor);
pp_(0,3,pcolor);pp_(1,5,pcolor);
}
else if(h1[sn/5]==1 && v(4,0,nx,ny)==1){
pp_(2,7,pcolor);pp_(3,9,pcolor);
pp_(0,3,pcolor);pp_(1,5,pcolor);
}
else if(h1[sn/5]==0 && v(4,0,nx,ny)==1){
pp_(2,7,pcolor);
pp_(0,3,pcolor);pp_(1,5,pcolor);

snold=sn;
if(sn==5) sn=0;
else if(sn==10) sn=5;

pp_(2,5,pcolor);
sn=snold;
}
else if(h1[sn/5]==1 && v(4,4,nx,ny)==1){
pp_(2,7,pcolor);pp_(3,9,pcolor);
pp_(0,3,pcolor);pp_(1,5,pcolor);
}
else if(h1[sn/5]==0 && v(4,4,nx,ny)==1){
pp_(2,7,pcolor);
pp_(0,3,pcolor);pp_(1,5,pcolor);

snold=sn;
if(sn==5) sn=0;
else if(sn==10) sn=5;

pp_(2,5,pcolor);
sn=snold;
}
/* v2 */
else if(h1[sn/5]==1 && v(1,1,nx,ny)==1){
pp_(3,7,pcolor);pp_(1,0,pcolor);
```

```
}
else if(h1[sn/5]==0 && v(1,1,nx,ny)==1){
pp_(3,7,pcolor);pp_(1,0,pcolor);

snold=sn;
if(sn==5) sn=0;
else if(sn==10) sn=5;

pp_(2,1,pcolor);
pp_(3,6,pcolor);
sn=snold;
}
else if(h1[sn/5]==1 && v(2,1,nx,ny)==1){
pp_(3,7,pcolor);pp_(1,0,pcolor);
}
else if(h1[sn/5]==0 && v(2,1,nx,ny)==1){
pp_(3,7,pcolor);pp_(1,0,pcolor);

snold=sn;
if(sn==5) sn=0;
else if(sn==10) sn=5;

pp_(2,1,pcolor);
pp_(3,6,pcolor);
sn=snold;
}
/* v3 */
else if(h1[sn/5]==1 && v(1,2,nx,ny)==1){
pp_(4,8,pcolor);pp_(0,0,pcolor);pp_(1,2,pcolor);
}
else if(h1[sn/5]==1 && v(2,0,nx,ny)==1){
pp_(4,8,pcolor);pp_(0,0,pcolor);pp_(1,2,pcolor);
}
else if(h1[sn/5]==0 && v(2,0,nx,ny)==1){
pp_(0,0,pcolor);pp_(1,2,pcolor);

snold=sn;
if(sn==5) sn=0;
else if(sn==10) sn=5;

pp_(2,2,pcolor);
sn=snold;
}
else if(h1[sn/5]==1 && v(2,4,nx,ny)==1){
pp_(4,8,pcolor);pp_(0,0,pcolor);pp_(1,2,pcolor);
}
```

```
else if(h1[sn/5]==0 && v(2,4,nx,ny)==1){
pp_(0,0,pcolor);pp_(1,2,pcolor);

snold=sn;
if(sn==5) sn=0;
else if(sn==10) sn=5;

pp_(2,2,pcolor);
sn=snold;
}

/* v0_ */
else if(h6[sn/5]==1 && v(6,1,nx,ny)==1){
pp_(3,4,pcolor);pp_(2,5,pcolor);
}
else if(h6[sn/5]==1 && v(6,5,nx,ny)==1){
pp_(3,4,pcolor);pp_(2,5,pcolor);
}
else if(h6[sn/5]==0 && v(6,5,nx,ny)==1){
pp_(2,5,pcolor);

snold=sn;
if(sn==0) sn=5;
else if(sn==5) sn=10;

pp_(2,7,pcolor);
pp_(0,3,pcolor);pp_(1,5,pcolor);
sn=snold;
}
/* v1_ */
else if(h6[sn/5]==1 && v(6,2,nx,ny)==1){
pp_(4,5,pcolor);pp_(3,3,pcolor);pp_(2,4,pcolor);
}
else if(h6[sn/5]==1 && v(3,2,nx,ny)==1){
pp_(4,5,pcolor);pp_(3,3,pcolor);pp_(2,4,pcolor);
}
else if(h6[sn/5]==0 && v(3,2,nx,ny)==1){
pp_(3,3,pcolor);pp_(2,4,pcolor);

snold=sn;
if(sn==0) sn=5;
else if(sn==5) sn=10;

pp_(1,3,pcolor);
sn=snold;
}
```

```
else if(h6[sn/5]==1 && v(3,4,nx,ny)==1){
pp_(4,5,pcolor);pp_(3,3,pcolor);pp_(2,4,pcolor);
}
else if(h6[sn/5]==0 && v(3,4,nx,ny)==1){
pp_(3,3,pcolor);pp_(2,4,pcolor);

snold=sn;
if(sn==0) sn=5;
else if(sn==5) sn=10;

pp_(1,3,pcolor);
sn=snold;
}
/* v2_ */
else if(h6[sn/5]==1 && v(6,3,nx,ny)==1){
pp_(4,6,pcolor);pp_(2,2,pcolor);
}
else if(h6[sn/5]==1 && v(3,1,nx,ny)==1){
pp_(4,6,pcolor);pp_(2,2,pcolor);
}
else if(h6[sn/5]==0 && v(3,1,nx,ny)==1){
pp_(2,2,pcolor);

snold=sn;
if(sn==0) sn=5;
else if(sn==5) sn=10;

pp_(0,0,pcolor);
pp_(1,2,pcolor);
sn=snold;
}
/* v3_ */
else if(h6[sn/5]==1 && v(6,4,nx,ny)==1){
pp_(3,6,pcolor);pp_(2,1,pcolor);
}
else if(h6[sn/5]==0 && v(6,4,nx,ny)==1){
pp_(3,6,pcolor);pp_(2,1,pcolor);

snold=sn;
if(sn==0) sn=5;
else if(sn==5) sn=10;

pp_(1,0,pcolor);
pp_(3,7,pcolor);
sn=snold;
}
```

```
else if(h6[sn/5]==1 && v(2,2,nx,ny)==1){
pp_(3,6,pcolor);pp_(2,1,pcolor);
}
else if(h6[sn/5]==0 && v(2,2,nx,ny)==1){
pp_(3,6,pcolor);pp_(2,1,pcolor);

snold=sn;
if(sn==0) sn=5;
else if(sn==5) sn=10;

pp_(1,0,pcolor);
pp_(3,7,pcolor);
sn=snold;
}
else if(1){
/*999*/
if(0) ;
#if 1
else if(side(1,2,nx,ny)) flag=101;
else if(side(3,5,nx,ny)==1 && h1[sn/5]==1) flag=102;  /* s3 */
else if(side(3,5,nx,ny)==1 && h1[sn/5]==0) flag=-102;  /* s3 */
else if(side(6,2,nx,ny)) flag=103;
else if(side(3,1,nx,ny)==1 && h6[sn/5]==1) flag=104;  /* s1_ */
else if(side(3,1,nx,ny)==1 && h6[sn/5]==0) flag=-104;  /* s1_ */
else if(side(5,2,nx,ny)) flag=105;
else if(side(2,1,nx,ny)) flag=106;
else if(side(1,3,nx,ny)) flag=107;
else if(side(4,0,nx,ny)==1 && h1[sn/5]==1) flag=108;  /* s0 */
else if(side(4,0,nx,ny)==1 && h1[sn/5]==0) flag=-108;  /* s0 */
else if(side(6,3,nx,ny)) flag=109;
else if(side(2,2,nx,ny)==1 && h6[sn/5]==1) flag=110;  /* s2_ */
else if(side(2,2,nx,ny)==1 && h6[sn/5]==0) flag=-110;  /* s2_ */
else if(side(1,1,nx,ny)) flag=111;
else if(side(2,0,nx,ny)==1 && h1[sn/5]==1) flag=112;  /* s2 */
else if(side(2,0,nx,ny)==1 && h1[sn/5]==0) flag=-112;  /* s2 */
else if(side(6,1,nx,ny)) flag=113;
else if(side(4,2,nx,ny)==1 && h6[sn/5]==1) flag=114;  /* s0_ */
else if(side(4,2,nx,ny)==1 && h6[sn/5]==0) flag=-114;  /* s0_ */

else if(side(5,1,nx,ny)==1 && h6[sn/5]==0) flag=-115;  /* s3_ */
else if(side(5,3,nx,ny)==1 && h1[sn/5]==0) flag=-116;  /* s1 */
#endif

else if(cut(1,4,nx,ny)) flag=1001;
else if(cut(1,0,nx,ny)) flag=1002;
else if(cut(2,4,nx,ny)) flag=1003;
```

```
else if(cut(2,0,nx,ny)) flag=1004;
else if(cut(4,4,nx,ny)) flag=1005;
else if(cut(4,0,nx,ny)) flag=1006;
else if(cut(5,4,nx,ny)) flag=1007;
else if(cut(5,0,nx,ny)) flag=1008;
else if(cut(3,2,nx,ny)) flag=1009;
else if(cut(3,4,nx,ny)) flag=1010;
else if(cut(6,5,nx,ny)) flag=1011;
else if(cut(6,1,nx,ny)) flag=1012;

else flag=0;

if(flag!=/*>*/0){
if(flag==101){
nx=1*(n-1);ny=ny-6*(n-1);
}
else if(flag==102){
nx=4*(n-1);ny=ny+6*(n-1);
}
else if(flag==103){
dlt=ny-5*(n-1);nx=3*(n-1)-dlt;ny=3*(n-1)-dlt;
}
else if(flag==104){
dlt=3*(n-1)-nx;nx=4*(n-1);ny=5*(n-1)+dlt;
}
else if(flag==105){
dlt=ny-6*(n-1);nx=2*(n-1)-dlt;ny=1*(n-1)-dlt;
}
else if(flag==106){
dlt=2*(n-1)-nx;nx=3*(n-1);ny=6*(n-1)+dlt;
}
else if(flag==107){
nx-=3*(n-1);ny=3*(n-1);
}
else if(flag==108){
nx+=3*(n-1);ny=9*(n-1);
}
else if(flag==109){
dlt=nx-3*(n-1);nx=2*(n-1);ny=1*(n-1)+dlt;
}
else if(flag==110){
dlt=ny-1*(n-1);nx=3*(n-1)+dlt;ny=6*(n-1);
}
else if(flag==111){
dlt=4*(n-1)-nx;nx=0*(n-1)+dlt;ny=0*(n-1);
}
```

```
else if(flag==112){
dlt=1*(n-1)-nx;nx=3*(n-1)+dlt;ny=7*(n-1)+dlt;
}
else if(flag==113){
dlt=4*(n-1)-nx;nx=2*(n-1);ny=4*(n-1)+dlt;
}
else if(flag==114){
dlt=5*(n-1)-ny;nx=3*(n-1)+dlt;ny=4*(n-1)+dlt;
}


else if(flag==-110){  /* s2_ */
snold=sn;
if(sn==0) sn=5;
else if(sn==5) sn=10;

dlt=ny-1*(n-1);nx=1*(n-1)-dlt;ny=0*(n-1);
}
else if(flag==-112){  /* s2 */
snold=sn;
if(sn==5) sn=0;
else if(sn==10) sn=5;

dlt=1*(n-1)-nx;nx=2*(n-1);ny=1*(n-1)+dlt;
}
else if(flag==-108){  /* s0 */
snold=sn;
if(sn==5) sn=0;
else if(sn==10) sn=5;

dlt=1*(n-1)-nx;nx=2*(n-1);ny=4*(n-1)+dlt;
}
else if(flag==-114){  /* s0_ */
snold=sn;
if(sn==0) sn=5;
else if(sn==5) sn=10;

dlt=ny-4*(n-1);nx=1*(n-1)-dlt;ny=3*(n-1);
}
else if(flag==-102){  /* s3 */
snold=sn;
if(sn==5) sn=0;
else if(sn==10) sn=5;

dlt=ny-2*(n-1);nx=2*(n-1)+dlt;ny=ny;
}
```

```
else if(flag==-104){  /* s1_ */
snold=sn;
if(sn==0) sn=5;
else if(sn==5) sn=10;

nx=1*(n-1);ny=ny;
}
else if(flag==-115){  /* s3_ */
snold=sn;
if(sn==0) sn=5;
else if(sn==5) sn=10;

nx=nx;ny=7*(n-1);
}
else if(flag==-116){  /* s1 */
snold=sn;
if(sn==5) sn=0;
else if(sn==10) sn=5;

dlt=3*(n-1)-nx;nx=nx;ny=6*(n-1)-dlt;
}



else if(flag==1001){
dlt=ny-8*(n-1);nx-=dlt;ny=8*(n-1);
dlt=3*(n-1)-nx;nx=nx;ny-=dlt;
}
else if(flag==1002){
nx=nx;ny=8*(n-1);
dlt=3*(n-1)-nx;nx=3*(n-1);ny+=dlt;
}
else if(flag==1003){
dlt=ny-1*(n-1);nx-=dlt;ny=1*(n-1);
dlt=1*(n-1)-nx;nx=nx;ny-=dlt;
}
else if(flag==1004){
nx=nx;ny=1*(n-1);
dlt=1*(n-1)-nx;nx=1*(n-1);ny+=dlt;
}
else if(flag==1005){
dlt=ny-4*(n-1);nx-=dlt;ny=4*(n-1);
dlt=1*(n-1)-nx;nx=nx;ny-=dlt;
}
else if(flag==1006){
nx=nx;ny=4*(n-1);
dlt=1*(n-1)-nx;nx=1*(n-1);ny+=dlt;
```

```
}
else if(flag==1007){
dlt=ny-6*(n-1);nx-=dlt;ny=6*(n-1);
dlt=2*(n-1)-nx;nx=nx;ny-=dlt;
}
else if(flag==1008){
nx=nx;ny=6*(n-1);
dlt=2*(n-1)-nx;nx=2*(n-1);ny+=dlt;
}
else if(flag==1009){
dlt=nx-2*(n-1);nx=nx;ny+=dlt;
nx=2*(n-1);ny=ny;
}
else if(flag==1010){
dlt=ny-3*(n-1);nx+=dlt;ny=ny;
nx=nx;ny=3*(n-1);
}

else if(flag==1011){
dlt=3*(n-1)-nx;nx=nx;ny-=dlt;
nx=3*(n-1);ny=ny;
}
else if(flag==1012){
dlt=5*(n-1)-ny;nx-=dlt;ny=ny;
nx=nx;ny=5*(n-1);
}

nx+=sn*(n-1);
putpixel(nx,ny,pcolor);

if(flag>=-116 && flag<=-102){
sn=snold;
}
}
}
}/** putpixel_ **/


int in(int nx,int ny)
{
/*if(nx<0 || ny<0) */return 0;

return pixel[nx][ny];
}/** in **/
```

```
int getpixel_(int x,int y,int nx,int ny)
{
int n,dlt,flag,dsn;

/*if(nx<0 || ny<0) return 0;*/  /* here */

n=RESO;

x-=sn*(n-1);
nx-=sn*(n-1);

/*999*/
/*FNUM=1;field(0,0,5);
h1[sn/5]=1;h6[sn/5]=1;*/
if(v(1,0,x,y)) flag=1;
else if(v(1,1,x,y)) flag=2;
else if(v(1,2,x,y)) flag=3;
else if(v(1,3,x,y)) flag=4;
else if(v(1,4,x,y)) flag=5;
else if(v(2,0,x,y)) flag=6;
else if(v(2,1,x,y)) flag=7;
else if(v(2,2,x,y)) flag=8;
else if(v(2,3,x,y)) flag=9;
else if(v(2,4,x,y)) flag=10;
else if(v(3,2,x,y)) flag=11;
else if(v(3,4,x,y)) flag=12;
else if(v(3,5,x,y)) flag=13;
else if(v(4,0,x,y)) flag=14;
else if(v(4,4,x,y)) flag=15;
else if(v(5,2,x,y)) flag=16;
else if(v(6,1,x,y)) flag=17;
else if(v(6,2,x,y)) flag=18;
else if(v(6,3,x,y)) flag=19;

#if 1
else if(side(1,2,x,y)) flag=101;
else if(side(3,5,x,y)==1 && h1[sn/5]==1) flag=102;  /* s3 */
else if(side(3,5,x,y)==1 && h1[sn/5]==0) flag=-102;  /* s3 */
else if(side(6,2,x,y)) flag=103;
else if(side(3,1,x,y)==1 && h6[sn/5]==1) flag=104;  /* s1_ */
else if(side(3,1,x,y)==1 && h6[sn/5]==0) flag=-104;  /* s1_ */
else if(side(5,2,x,y)) flag=105;
else if(side(2,1,x,y)) flag=106;
else if(side(1,3,x,y)) flag=107;
else if(side(4,0,x,y)==1 && h1[sn/5]==1) flag=108;  /* s0 */
else if(side(4,0,x,y)==1 && h1[sn/5]==0) flag=-108;  /* s0 */
```

```
else if(side(6,3,x,y)) flag=109;
else if(side(2,2,x,y)==1 && h6[sn/5]==1) flag=110;  /* s2_ */
else if(side(2,2,x,y)==1 && h6[sn/5]==0) flag=-110;  /* s2_ */
else if(side(1,1,x,y)) flag=111;
else if(side(2,0,x,y)==1 && h1[sn/5]==1) flag=112;  /* s2 */
else if(side(2,0,x,y)==1 && h1[sn/5]==0) flag=-112;  /* s2 */
else if(side(6,1,x,y)) flag=113;
else if(side(4,2,x,y)==1 && h6[sn/5]==1) flag=114;  /* s0_ */
else if(side(4,2,x,y)==1 && h6[sn/5]==0) flag=-114;  /* s0_ */

else if(side(5,1,x,y)==1 && h6[sn/5]==0) flag=-115;  /* s3_ */
else if(side(5,3,x,y)==1 && h1[sn/5]==0) flag=-116;  /* s1 */
#endif

else if(cut(1,4,x,y)) flag=1001;
else if(cut(1,0,x,y)) flag=1002;
else if(cut(2,4,x,y)) flag=1003;
else if(cut(2,0,x,y)) flag=1004;
else if(cut(4,4,x,y)) flag=1005;
else if(cut(4,0,x,y)) flag=1006;
else if(cut(5,4,x,y)) flag=1007;
else if(cut(5,0,x,y)) flag=1008;
else if(cut(3,2,x,y)) flag=1009;
else if(cut(3,4,x,y)) flag=1010;
else if(cut(6,5,x,y)) flag=1011;
else if(cut(6,1,x,y)) flag=1012;

else flag=0;
/*h1[sn/5]=0;h6[sn/5]=0;*/

X=nx;Y=ny;
jmpflag=0;
dsn=0;
/*goto end;*/  /* here */

/* v(2,1, v(2,2, */
if(flag!=0){
/*if(flag>0){*/
if(flag==1){  /* v(1,0, */
     if(nh(x,y,nx,ny,2)) {x=/*3*/0*(n-1);y=/*9*/3*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=0*(n-1);y=3*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=1*(n-1);y=5*(n-1);X=x+1;Y=y;jmpflag=1;}
}
else if(flag==2){  /* v(1,1, */
     if(nh(x,y,nx,ny,0)) {x=1*(n-1);y=0*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(h1[sn/5]==0 && nh(x,y,nx,ny,1)==1) {x=1*(n-1);y=0*(n-1);X=x-1;Y=y;jmpflag=1;}
```

```
}
else if(flag==3){  /* v(1,2, */
     if(nh(x,y,nx,ny,5)) {x=0*(n-1);y=0*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=1*(n-1);y=2*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=1*(n-1);y=2*(n-1);X=x+1;Y=y+1;jmpflag=1;}
}
else if(flag==4){  /* v(1,3, */
     if(nh(x,y,nx,ny,0)) {x=1*(n-1);y=3*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=1*(n-1);y=3*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,2)) {x=1*(n-1);y=3*(n-1);X=x;Y=y+1;jmpflag=1;}
}
else if(flag==5){  /* v(1,4, */
     if(nh(x,y,nx,ny,1)) {x=0*(n-1);y=3*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,2)) {x=1*(n-1);y=5*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=1*(n-1);y=5*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=2*(n-1);y=7*(n-1);X=x+1;Y=y;jmpflag=1;}
}
else if(flag==6){  /* v(2,0, */
     if(nh(x,y,nx,ny,2)) {x=1*(n-1);y=2*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=1*(n-1);y=2*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=/*4*/1*(n-1);y=/*8*/2*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,5)) {x=4*(n-1);y=8*(n-1);X=x-1;Y=y;jmpflag=1;}
}
else if(flag==7){  /* v(2,1, */
     if(nh(x,y,nx,ny,4)) {x=3*(n-1);y=7*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,5)) {x=3*(n-1);y=7*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=3*(n-1);y=7*(n-1);X=x-1;Y=y-1;jmpflag=1;}
}
else if(flag==8){  /* v(2,2, */
     if(nh(x,y,nx,ny,5)) {x=3*(n-1);y=6*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=3*(n-1);y=6*(n-1);X=x-1;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=3*(n-1);y=6*(n-1);X=x;Y=y-1;jmpflag=1;}
}
else if(flag==9){  /* v(2,3, */
     if(nh(x,y,nx,ny,0)) {x=4*(n-1);y=6*(n-1);X=x-1;Y=y-1;jmpflag=1;}
}
else if(flag==10){  /* v(2,4, */
     if(nh(x,y,nx,ny,3)) {x=4*(n-1);y=8*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=/*4*/0*(n-1);y=/*8*/0*(n-1);X=x+1;Y=y;jmpflag=1;}
}
else if(flag==11){  /* v(3,2, */
     if(nh(x,y,nx,ny,5)) {x=4*(n-1);y=5*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=2*(n-1);y=4*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=2*(n-1);y=4*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,2)) {x=2*(n-1);y=4*(n-1);X=x-1;Y=y-1;jmpflag=1;}
}
```

```
else if(flag==12){  /* v(3,4, */
     if(nh(x,y,nx,ny,0)) {x=3*(n-1);y=3*(n-1);X=x-1;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=4*(n-1);y=5*(n-1);X=x-1;Y=y;jmpflag=1;}
}
else if(flag==13){  /* v(3,5, */
     if(nh(x,y,nx,ny,4)) {x=4*(n-1);y=9*(n-1);X=x-1;Y=y-1;jmpflag=1;}
}
else if(flag==14){  /* v(4,0, */
     if(nh(x,y,nx,ny,2)) {x=1*(n-1);y=5*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=1*(n-1);y=5*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=2*(n-1);y=7*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,5)) {x=2*(n-1);y=7*(n-1);X=x+1;Y=y+1;jmpflag=1;}
}
else if(flag==15){  /* v(4,4, */
     if(nh(x,y,nx,ny,2)) {x=2*(n-1);y=7*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=2*(n-1);y=7*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=/*3*/0*(n-1);y=/*9*/3*(n-1);X=x+1;Y=y;jmpflag=1;}
}
else if(flag==16){  /* v(5,2, */
     if(nh(x,y,nx,ny,1)) {x=2*(n-1);y=1*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(h6[sn/5]==0 && nh(x,y,nx,ny,0)==1) {x=2*(n-1);y=1*(n-1);X=x;Y=y+1;jmpflag=1;}
}
else if(flag==17){  /* v(6,1, */
     if(nh(x,y,nx,ny,3)) {x=2*(n-1);y=5*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=2*(n-1);y=5*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,5)) {x=2*(n-1);y=5*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=2*(n-1);y=5*(n-1);X=x-1;Y=y-1;jmpflag=1;}
}
else if(flag==18){  /* v(6,2, */
     if(nh(x,y,nx,ny,5)) {x=2*(n-1);y=4*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=2*(n-1);y=4*(n-1);X=x-1;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=2*(n-1);y=4*(n-1);X=x;Y=y-1;jmpflag=1;}
}
else if(flag==19){  /* v(6,3, */
     if(nh(x,y,nx,ny,0)) {x=2*(n-1);y=2*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=2*(n-1);y=2*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,2)) {x=2*(n-1);y=2*(n-1);X=x-1;Y=y-1;jmpflag=1;}
}

else if(flag==101){  /* side(1,2, */
     if(nh(x,y,nx,ny,0)) {rot(0,0);jmpflag=101;}
else if(nh(x,y,nx,ny,1)) {rot(1,0);jmpflag=101;}
if(jmpflag>100) {x=1*(n-1);y=y-6*(n-1);}
}
else if(flag==102){  /* side(3,5, */
     if(nh(x,y,nx,ny,3)) {rot(3,0);jmpflag=101;}
```

```
else if(nh(x,y,nx,ny,4)) {rot(4,0);jmpflag=101;}
if(jmpflag>100) {x=4*(n-1);y=y+6*(n-1);}
}
else if(flag==103){  /* side(6,2, */
     if(nh(x,y,nx,ny,0)) {rot(0,2);jmpflag=101;}
else if(nh(x,y,nx,ny,1)) {rot(1,2);jmpflag=101;}
if(jmpflag>100) {dlt=y-5*(n-1);x=3*(n-1)-dlt;y=3*(n-1)-dlt;}
}
else if(flag==104){  /* side(3,1, */
     if(nh(x,y,nx,ny,5)) {rot(5,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,0)) {rot(0,-2);jmpflag=101;}
if(jmpflag>100) {dlt=3*(n-1)-x;x=4*(n-1);y=5*(n-1)+dlt;}
}
else if(flag==105){  /* side(5,2, */
     if(nh(x,y,nx,ny,0)) {rot(0,2);jmpflag=101;}
else if(nh(x,y,nx,ny,1)) {rot(1,2);jmpflag=101;}
if(jmpflag>100) {dlt=y-6*(n-1);x=2*(n-1)-dlt;y=1*(n-1)-dlt;}
}
else if(flag==106){  /* side(2,1, */
     if(nh(x,y,nx,ny,5)) {rot(5,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,0)) {rot(0,-2);jmpflag=101;}
if(jmpflag>100) {dlt=2*(n-1)-x;x=3*(n-1);y=6*(n-1)+dlt;}
}
else if(flag==107){  /* side(1,3, */
     if(nh(x,y,nx,ny,1)) {rot(1,0);jmpflag=101;}
else if(nh(x,y,nx,ny,2)) {rot(2,0);jmpflag=101;}
if(jmpflag>100) {x-=3*(n-1);y=3*(n-1);}
}
else if(flag==108){  /* side(4,0, */
     if(nh(x,y,nx,ny,4)) {rot(4,0);jmpflag=101;}
else if(nh(x,y,nx,ny,5)) {rot(5,0);jmpflag=101;}
if(jmpflag>100) {x+=3*(n-1);y=9*(n-1);}
}
else if(flag==109){  /* side(6,3, */
     if(nh(x,y,nx,ny,1)) {rot(1,2);jmpflag=101;}
else if(nh(x,y,nx,ny,2)) {rot(2,2);jmpflag=101;}
if(jmpflag>100) {dlt=x-3*(n-1);x=2*(n-1);y=1*(n-1)+dlt;}
}
else if(flag==110){  /* side(2,2, */
     if(nh(x,y,nx,ny,0)) {rot(0,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,1)) {rot(1,-2);jmpflag=101;}
if(jmpflag>100) {dlt=y-1*(n-1);x=3*(n-1)+dlt;y=6*(n-1);}
}
else if(flag==111){  /* side(1,1, */
     if(nh(x,y,nx,ny,5)) {rot(5,2);jmpflag=101;}
else if(nh(x,y,nx,ny,0)) {rot(0,2);jmpflag=101;}
```

```
if(jmpflag>100) {dlt=4*(n-1)-x;x=0*(n-1)+dlt;y=0*(n-1);}
}
else if(flag==112){  /* side(2,0, */
     if(nh(x,y,nx,ny,4)) {rot(4,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,5)) {rot(5,-2);jmpflag=101;}
if(jmpflag>100) {dlt=1*(n-1)-x;x=3*(n-1)+dlt;y=7*(n-1)+dlt;}
}
else if(flag==113){  /* side(6,1, */
     if(nh(x,y,nx,ny,5)) {rot(5,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,0)) {rot(0,-2);jmpflag=101;}
if(jmpflag>100) {dlt=4*(n-1)-x;x=2*(n-1);y=4*(n-1)+dlt;}
}
else if(flag==114){  /* side(4,2, */
     if(nh(x,y,nx,ny,0)) {rot(0,2);jmpflag=101;}
else if(nh(x,y,nx,ny,1)) {rot(1,2);jmpflag=101;}
if(jmpflag>100) {dlt=5*(n-1)-y;x=3*(n-1)+dlt;y=4*(n-1)+dlt;}
}


else if(flag==-110){  /* side(2,2, */
     if(nh(x,y,nx,ny,0)==1) {rot(0,1);jmpflag=101;}
else if(nh(x,y,nx,ny,1)==1) {rot(1,1);jmpflag=101;}
if(jmpflag>100){
if(sn==0) dsn=5;
else if(sn==5) dsn=5;
dlt=y-1*(n-1);x=1*(n-1)-dlt;y=0*(n-1);
}
}
else if(flag==-112){  /* side(2,0, */
     if(nh(x,y,nx,ny,4)==1) {rot(4,-1);jmpflag=101;}
else if(nh(x,y,nx,ny,5)==1) {rot(5,-1);jmpflag=101;}
if(jmpflag>100){
if(sn==5) dsn=-5;
else if(sn==10) dsn=-5;
dlt=1*(n-1)-x;x=2*(n-1);y=1*(n-1)+dlt;
}
}
else if(flag==-108){  /* side(4,0, */
     if(nh(x,y,nx,ny,4)==1) {rot(4,-1);jmpflag=101;}
else if(nh(x,y,nx,ny,5)==1) {rot(5,-1);jmpflag=101;}
if(jmpflag>100){
if(sn==5) dsn=-5;
else if(sn==10) dsn=-5;
dlt=1*(n-1)-x;x=2*(n-1);y=4*(n-1)+dlt;
}
}
```

```
else if(flag==-114){  /* side(4,2, */
     if(nh(x,y,nx,ny,0)==1) {rot(0,1);jmpflag=101;}
else if(nh(x,y,nx,ny,1)==1) {rot(1,1);jmpflag=101;}
if(jmpflag>100){
if(sn==0) dsn=5;
else if(sn==5) dsn=5;
dlt=y-4*(n-1);x=1*(n-1)-dlt;y=3*(n-1);
}
}
else if(flag==-102){  /* side(3,5, */
     if(nh(x,y,nx,ny,3)==1) {rot(3,-1);jmpflag=101;}
else if(nh(x,y,nx,ny,4)==1) {rot(4,-1);jmpflag=101;}
if(jmpflag>100){
if(sn==5) dsn=-5;
else if(sn==10) dsn=-5;
dlt=y-2*(n-1);x=2*(n-1)+dlt;y=y;
}
}
else if(flag==-104){  /* side(3,1, */
     if(nh(x,y,nx,ny,5)==1) {rot(5,1);jmpflag=101;}
else if(nh(x,y,nx,ny,0)==1) {rot(0,1);jmpflag=101;}
if(jmpflag>100){
if(sn==0) dsn=5;
else if(sn==5) dsn=5;
x=1*(n-1);y=y;
}
}
else if(flag==-115){  /* side(5,1, */
     if(nh(x,y,nx,ny,5)==1) {rot(5,-1);jmpflag=101;}
else if(nh(x,y,nx,ny,0)==1) {rot(0,-1);jmpflag=101;}
if(jmpflag>100){
if(sn==0 && (Nx>=(sn+2)*(n-1) && Ny<=6*(n-1))) dsn=5;
else if(sn==5 && (Nx<(sn+2)*(n-1) || Ny>6*(n-1))) dsn=-5;
else if(sn==5 && (Nx>=(sn+2)*(n-1) && Ny<=6*(n-1))) dsn=5;
else if(sn==10 && (Nx<(sn+2)*(n-1) || Ny>6*(n-1))) dsn=-5;
/*if(sn==0) dsn=5;
else if(sn==5) dsn=5;*/
x=x;y=7*(n-1);
}
}
else if(flag==-116){  /* side(5,3, */
     if(nh(x,y,nx,ny,1)==1) {rot(1,1);jmpflag=101;}
else if(nh(x,y,nx,ny,2)==1) {rot(2,1);jmpflag=101;}
if(jmpflag>100){
if(sn==0 && (Nx>=(sn+2)*(n-1) && Ny<=6*(n-1))) dsn=5;
else if(sn==5 && (Nx<(sn+2)*(n-1) || Ny>6*(n-1))) dsn=-5;
```

```
else if(sn==5 && (Nx>=(sn+2)*(n-1) && Ny<=6*(n-1))) dsn=5;
else if(sn==10 && (Nx<(sn+2)*(n-1) || Ny>6*(n-1))) dsn=-5;
/*if(sn==5) dsn=-5;
else if(sn==10) dsn=-5;*/
dlt=3*(n-1)-x;x=x;y=6*(n-1)-dlt;
}
}


else if(flag==1001){  /* cut(1,4, */
     if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,2);jmpflag=1001;}
else if(nh(x,y,nx,ny,4)==1 && in(nx,ny)==0) {rot(4,2);jmpflag=1001;}
if(jmpflag>1000){
dlt=y-8*(n-1);x-=dlt;y=8*(n-1);
dlt=3*(n-1)-x;x=x;y-=dlt;
}
}
else if(flag==1002){  /* cut(1,0, */
     if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,-2);jmpflag=1001;}
else if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,-2);jmpflag=1001;}
if(jmpflag>1000){
x=x;y=8*(n-1);
dlt=3*(n-1)-x;x=3*(n-1);y+=dlt;
}
}
else if(flag==1003){  /* cut(2,4, */
     if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,2);jmpflag=1001;}
else if(nh(x,y,nx,ny,4)==1 && in(nx,ny)==0) {rot(4,2);jmpflag=1001;}
if(jmpflag>1000){
dlt=y-1*(n-1);x-=dlt;y=1*(n-1);
dlt=1*(n-1)-x;x=x;y-=dlt;
}
}
else if(flag==1004){  /* cut(2,0, */
     if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,-2);jmpflag=1001;}
else if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,-2);jmpflag=1001;}
if(jmpflag>1000){
x=x;y=1*(n-1);
dlt=1*(n-1)-x;x=1*(n-1);y+=dlt;
}
}
else if(flag==1005){  /* cut(4,4, */
     if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,2);jmpflag=1001;}
else if(nh(x,y,nx,ny,4)==1 && in(nx,ny)==0) {rot(4,2);jmpflag=1001;}
if(jmpflag>1000){
dlt=y-4*(n-1);x-=dlt;y=4*(n-1);
```

```
dlt=1*(n-1)-x;x=x;y-=dlt;
}
}
else if(flag==1006){  /* cut(4,0, */
     if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,-2);jmpflag=1001;}
else if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,-2);jmpflag=1001;}
if(jmpflag>1000){
x=x;y=4*(n-1);
dlt=1*(n-1)-x;x=1*(n-1);y+=dlt;
}
}
else if(flag==1007){  /* cut(5,4, */
     if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,2);jmpflag=1001;}
else if(nh(x,y,nx,ny,4)==1 && in(nx,ny)==0) {rot(4,2);jmpflag=1001;}
if(jmpflag>1000){
dlt=y-6*(n-1);x-=dlt;y=6*(n-1);
dlt=2*(n-1)-x;x=x;y-=dlt;
}
}
else if(flag==1008){  /* cut(5,0, */
     if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,-2);jmpflag=1001;}
else if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,-2);jmpflag=1001;}
if(jmpflag>1000){
x=x;y=6*(n-1);
dlt=2*(n-1)-x;x=2*(n-1);y+=dlt;
}
}
else if(flag==1009){  /* cut(3,2, */
     if(nh(x,y,nx,ny,1)==1 && in(nx,ny)==0) {rot(1,2);jmpflag=1001;}
else if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,2);jmpflag=1001;}
if(jmpflag>1000){
dlt=x-2*(n-1);x=x;y+=dlt;
x=2*(n-1);y=y;
}
}
else if(flag==1010){  /* cut(3,4, */
     if(nh(x,y,nx,ny,0)==1 && in(nx,ny)==0) {rot(0,-2);jmpflag=1001;}
else if(nh(x,y,nx,ny,1)==1 && in(nx,ny)==0) {rot(1,-2);jmpflag=1001;}
if(jmpflag>1000){
dlt=y-3*(n-1);x+=dlt;y=y;
x=x;y=3*(n-1);
}
}
else if(flag==1011){  /* cut(6,5, */
     if(nh(x,y,nx,ny,4)==1 && in(nx,ny)==0) {rot(4,2);jmpflag=1001;}
else if(nh(x,y,nx,ny,5)==1 && in(nx,ny)==0) {rot(5,2);jmpflag=1001;}
```

```
if(jmpflag>1000){
dlt=3*(n-1)-x;x=x;y-=dlt;
x=3*(n-1);y=y;
}
}
else if(flag==1012){  /* cut(6,1, */
     if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,-2);jmpflag=1001;}
else if(nh(x,y,nx,ny,4)==1 && in(nx,ny)==0) {rot(4,-2);jmpflag=1001;}
if(jmpflag>1000){
dlt=5*(n-1)-y;x-=dlt;y=y;
x=x;y=5*(n-1);
}
}

if(jmpflag>100){
X=x+tmp0;
Y=y+tmp1;
}
}/**if(flag)**/

end:
X_=x;
Y_=y;

X+=sn*(n-1);  /* restore */
X_+=sn*(n-1);

X+=dsn*(n-1);  /* jump */
X_+=dsn*(n-1);
sn_=sn+dsn; /* new sn */

return pixel[X][Y];
}/** getpixel_ **/


int random_(int n)
{
int val;

val=(int)((rand()/(RAND_MAX+1.))*n);

return val;
}/** random_ **/


long ftell_mem(int i)
```

```
{
return fp_mem[i];
}/** ftell_mem **/


void fwrite_mem(int i)
{
rtn[i][fp_mem[i]]=s;
fp_mem[i]++;if(fp_mem[i]>asize-1) refill=0;
}/** fwrite_mem **/


void fread_mem(int i)
{
fp_mem[i]--;if(fp_mem[i]<0) fp_mem[i]=0;
s=rtn[i][fp_mem[i]];
}/** fread_mem **/


int fen(char *str,int j,int jmax)
{
int j_;

if(j==jmax+1) j_=0;
else if(j==-1) j_=jmax;
else j_=j;

if(strcmp(str,"X")==0) return enX[j_];
else if(strcmp(str,"Y")==0) return enY[j_];

else if(strcmp(str,"X_")==0) return enX_[j_];
else if(strcmp(str,"Y_")==0) return enY_[j_];

else if(strcmp(str,"SN")==0) return enSN[j_];
}/** fen **/


int check_v(int x,int y)
{
int val;

x-=sn*(RESO-1);

if(0) ;

else if(v(3,4,x,y)==1 && h6[sn/5]==0) val=1;  /* v1_ */
```

```
else if(v(3,2,x,y)==1 && h6[sn/5]==0) val=3;  /* v1_ */
else if(v(2,2,x,y)==1 && h6[sn/5]==0) val=1;  /* v3_ */
else if(v(5,2,x,y)==1 && h6[sn/5]==0) val=3;  /* v3_ */
else if(v(3,5,x,y)==1 && h1[sn/5]==0) val=4;  /* v0 */
else if(v(2,1,x,y)==1 && h1[sn/5]==0) val=4;  /* v2 */
else if(v(1,1,x,y)==1 && h1[sn/5]==0) val=0;  /* v2 */

else if(v(6,5,x,y)==1 && h6[sn/5]==0) val=2;  /* v0_ */
else if(v(2,3,x,y)==1 && h6[sn/5]==0) val=2;  /* v2_ */
else if(v(4,0,x,y)==1 && h1[sn/5]==0) val=-3;  /* v1 */
else if(v(4,4,x,y)==1 && h1[sn/5]==0) val=5;  /* v1 */
else if(v(1,0,x,y)==1 && h1[sn/5]==0) val=1;  /* v1 */
else if(v(2,0,x,y)==1 && h1[sn/5]==0) val=-3;  /* v3 */
else if(v(2,4,x,y)==1 && h1[sn/5]==0) val=5;  /* v3 */

else val=-1;

x+=sn*(RESO-1);

return val;
}/** check_v **/


void set_vals(int j)
{
int snold;

    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;

snold=sn;
sn=sn_;

if(pixel[X][Y]==0) getpixel_(X_,Y_,X,Y);
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;

sn=snold;
}/** set_vals **/


int cag_r(void)
{
int i,j,dx,dy,n,sn1,sn2,jmax,dsn,x0,y0;
int flag_[CPMAX],flag_pp[CPMAX],acolor[5*2];
int nx[CPMAX],ny[CPMAX],nx_[CPMAX],ny_[CPMAX],nax[5*2],nay[5*2];
int x[8],y[8],x_[8],y_[8];
int cp,ssize,posflag,pos,count;
```

```
int ca,c1,c2,c3,c4,c5,c7,cflag,cc1,cc2;
int nxp,nxm,nyp,nym,oldx,oldy;
int jmp[6+6],sntmp1,sntmp2,snold;

if(0){
c1=15;
putpixel_(2*(RESO-1),7*(RESO-1),c1);
putpixel_(3*(RESO-1),7*(RESO-1),c1);
putpixel_(4*(RESO-1),8*(RESO-1),c1);
putpixel_(4*(RESO-1),9*(RESO-1),c1);

sn=/*5*//*10*/10;
putpixel_((sn+3)*(RESO-1),4*(RESO-1),c1);
putpixel_((sn+4)*(RESO-1),5*(RESO-1),c1);
putpixel_((sn+4)*(RESO-1),6*(RESO-1),c1);
putpixel_((sn+3)*(RESO-1),6*(RESO-1),c1);
sn=0;
}

ssize=sizeof(ss);
cp=CPMAX;
n=RESO;

acolor[0]=9;acolor[1]=10;acolor[2]=11;acolor[3]=12;
acolor[4]=1;acolor[5]=2;acolor[6]=3;acolor[7]=4;

for(i=0;i<CPMAX;i++){
rcount[i]=0;
flag_[i]=1;
fp_mem[i]=0;
}

ca=15;

/*999*/
if(CPMAX==4 && DIV==1){
dx=0;dy=0;
nax[0]=(7-3+dx)*(n-1)-1     ;nay[0]=(9+dy)*(n-1)-1;
nax[1]=(6-3+dx)*(n-1)     ;nay[1]=(7+dy)*(n-1)+1;

     if(FNUM==2) dx=5;
else if(FNUM==3) dx=10;
if(CROSS==0){  /* // */
nax[2]=(7-3+dx)*(n-1)-1     ;nay[2]=(5+dy)*(n-1);
nax[3]=(6-3+dx)*(n-1)     ;nay[3]=(6+dy)*(n-1)-1;
}
```

```
else{  /* x */
nax[2]=(2+dx)*(n-1)+1     ;nay[2]=(5+dy)*(n-1);
nax[3]=(7-3+dx)*(n-1)-1     ;nay[3]=(6+dy)*(n-1)-1;
}
}
else if(DIV==0){
dx=0;dy=0;
/* h1 */
nax[0]=(5-3+dx)*(n-1)+1     ;nay[0]=(7+dy)*(n-1)+1;
nax[1]=(6-3+dx)*(n-1)     ;nay[1]=(7+dy)*(n-1)+1;
nax[2]=(7-3+dx)*(n-1)-1     ;nay[2]=(8+dy)*(n-1);
nax[3]=(7-3+dx)*(n-1)-1     ;nay[3]=(9+dy)*(n-1)-1;
/* h6 */
dx=/*0*//*5*//*10*/10;
/*nax[0]=(6-3+dx)*(n-1)     ;nay[0]=(4+dy)*(n-1)+1;
nax[1]=(7-3+dx)*(n-1)-1     ;nay[1]=(5+dy)*(n-1);
nax[2]=(7-3+dx)*(n-1)-1     ;nay[2]=(6+dy)*(n-1)-1;
nax[3]=(6-3+dx)*(n-1)     ;nay[3]=(6+dy)*(n-1)-1;*/
/* centre h5 */
dx=5;dy=-2;
/*nax[0]=(5-4+dx)*(n-1)+1     ;nay[0]=(7+dy)*(n-1)+1;
nax[1]=(6-4+dx)*(n-1)     ;nay[1]=(7+dy)*(n-1)+1;
nax[2]=(7-4+dx)*(n-1)-1     ;nay[2]=(8+dy)*(n-1);
nax[3]=(7-4+dx)*(n-1)-1     ;nay[3]=(9+dy)*(n-1)-1;*/
}
else if(DIV==1){
dx=0;dy=0;
nax[0]=(5-3+dx)*(n-1)+1     ;nay[0]=(7+dy)*(n-1)+1;
nax[1]=(6-3+dx)*(n-1)     ;nay[1]=(7+dy)*(n-1)+1;
nax[2]=(7-3+dx)*(n-1)-1     ;nay[2]=(8+dy)*(n-1);
nax[3]=(7-3+dx)*(n-1)-1     ;nay[3]=(9+dy)*(n-1)-1;

     if(FNUM==2) dx=5;
else if(FNUM==3) dx=10;
nax[4]=(6-3+dx)*(n-1)     ;nay[4]=(4+dy)*(n-1)+1;
nax[5]=(7-3+dx)*(n-1)-1     ;nay[5]=(5+dy)*(n-1);
nax[6]=(7-3+dx)*(n-1)-1     ;nay[6]=(6+dy)*(n-1)-1;
nax[7]=(6-3+dx)*(n-1)     ;nay[7]=(6+dy)*(n-1)-1;
}

if(0){
dx=10;dy=0;
nax[0]=(6-3+dx)*(n-1)     ;nay[0]=(4+dy)*(n-1)+1;
nax[1]=(7-3+dx)*(n-1)-1     ;nay[1]=(6+dy)*(n-1)-1;
}
```

```
/* here */
     if(FNUM==1) {sn1=0;sn2=0;}
else if(FNUM==2) {sn1=0;sn2=5;}
else             {sn1=/*0*//*5*//*10*/0;sn2=10;}

i=0;
while(1){
if(flag_[i]){                        /* CP_? */
if(i<=CPHALF-1) sn=sn1;else sn=sn2;
ig=i;

nx[i]=nax[i];ny[i]=nay[i];
putpixel_(nx[i],ny[i],acolor[i]);
}/**if(flag_[i])**/

i++;if(i==CPMAX) break;
}/**while(1)**/

i=0;
while(1){
if(flag_[i]){                        /* CP_? */
nx_[i]=nax[i];ny_[i]=nay[i];
if(i<=CPHALF-1) sn=sn1;else sn=sn2;
ig=i;

if(combination==0){
/* CW */
}/**if(combination)**/
else{
/* CCW */
if(drn==4) {/* 217 */
if(CPMAX==4 && DIV==1){
if(CROSS==0){  /* // */
     if(i==0) {nay[i]--;}
else if(i==1) {nax[i]--;}
else if(i==2) {nax[i]--;nay[i]--;}  /* odd, even */
else if(i==3) {nax[i]++;}
}
else{  /* x */
     if(i==0) {nay[i]--;}
else if(i==1) {nax[i]--;}
else if(i==2) {nax[i]++;nay[i]++;}  /* even, even */
else if(i==3) {nay[i]--;}
}
}
else{
```

```
     if(i==0) {nax[i]++;}
else if(i==1) {nax[i]++;nay[i]++;}
else if(i==2) {nay[i]++;}
else if(i==3) {nax[i]--;}
/* h6 */
/*     if(i==0) {nax[i]++;nay[i]++;}
else if(i==1) {nay[i]++;}
else if(i==2) {nax[i]--;}
else if(i==3) {nax[i]--;nay[i]--;}*/

else if(i==4) {nax[i]++;nay[i]++;}
else if(i==5) {nay[i]++;}
else if(i==6) {nax[i]--;}
else if(i==7) {nax[i]--;nay[i]--;}
}
}
}/**else(combination)**/

nx[i]=nax[i];ny[i]=nay[i];

putpixel_(nx[i],ny[i],acolor[i]);
}/**if(flag_[i])**/

i++;if(i==CPMAX) break;
}/**while(1)**/

if(GRPH==1 && cnt==0) use_subroop();
cnt++;
/***************************** while(cp) -> ******************************/

while(cp){
kbhit_();
if(refill==0) break;

algo=random_(2);
/*if(cnt>=282 && rcount[0]>=72){
use_subroop();
printf(" %ld %ld %d\n",rcount[0],rcount[1],algo);
}*/

i=0;
while(1){

if(flag_[i]){                        /* CP_? */
if(i<=CPHALF-1) sn=sn1;else sn=sn2;
ig=i;
```

```
Nx=nx[i];Ny=ny[i];

nxp=nx[i]+1;nyp=ny[i]+1;nxm=nx[i]-1;nym=ny[i]-1;
c1=getpixel_(nx[i],ny[i],nxp,ny[i]);
x[1]=X;y[1]=Y;x_[1]=X_;y_[1]=Y_;jmp[1]=sn_;
c2=getpixel_(nx[i],ny[i],nx[i],nyp);
x[2]=X;y[2]=Y;x_[2]=X_;y_[2]=Y_;jmp[2]=sn_;
c3=getpixel_(nx[i],ny[i],nxm,ny[i]);
x[3]=X;y[3]=Y;x_[3]=X_;y_[3]=Y_;jmp[3]=sn_;
c4=getpixel_(nx[i],ny[i],nx[i],nym);
x[4]=X;y[4]=Y;x_[4]=X_;y_[4]=Y_;jmp[4]=sn_;
c5=getpixel_(nx[i],ny[i],nxp,nyp);
x[5]=X;y[5]=Y;x_[5]=X_;y_[5]=Y_;jmp[5]=sn_;
c7=getpixel_(nx[i],ny[i],nxm,nym);
x[7]=X;y[7]=Y;x_[7]=X_;y_[7]=Y_;jmp[7]=sn_;



if(sn==0 && (Nx>=(sn+2)*(n-1) && Ny<=6*(n-1))) dsn=5;
else if(sn==5 && (Nx<(sn+2)*(n-1) || Ny>6*(n-1))) dsn=-5;
else if(sn==5 && (Nx>=(sn+2)*(n-1) && Ny<=6*(n-1))) dsn=5;
else if(sn==10 && (Nx<(sn+2)*(n-1) || Ny>6*(n-1))) dsn=-5;

posflag=check_v(Nx,Ny);
cflag=0;

if(posflag==-1){
if((c1==ca)||(c2==ca)||(c3==ca)||(c4==ca)||(c5==ca)||(c7==ca)) cflag=1;
else cflag=0;
}
else if(posflag==0){
if((c1==ca)||(c2==ca)||(c3==ca)||(c4==ca)||(c5==ca)||(c7==ca)) cflag=1;
else{
Ny+=-1*(n-1);

j=5;rot(2,1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(2,0);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(2,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);

/*cflag=0;
for(j=5;j<=7;j++) {if(pixel[enX[j]][enY[j]]==ca) {cflag=1;break;}}*/
}
```

```
}
else if(posflag==1){
if((c1==ca)||(c2==ca)||(c3==ca)||(c4==ca)||(c5==ca)||(c7==ca)) cflag=1;
else{
if(Ny==1*(n-1) || Ny==4*(n-1)){
Nx+=-1*(n-1);Ny+=-1*(n-1);
j=5;rot(1,1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(1,0);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(1,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
}
else if(Ny==7*(n-1)){
Ny+=-2*(n-1);
j=5;rot(1,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(1,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(1,1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
}

/*cflag=0;
for(j=5;j<=7;j++) {if(pixel[enX[j]][enY[j]]==ca) {cflag=1;break;}}*/
}
}
else if(posflag==2){
if((c1==ca)||(c2==ca)||(c3==ca)||(c4==ca)||(c5==ca)||(c7==ca)) cflag=1;
else{
if(Ny==2*(n-1)){
Nx+=-1*(n-1);
j=5;rot(0,1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(0,0);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(0,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
}
else if(Ny==5*(n-1)){
Ny+=2*(n-1);
j=5;rot(0,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(0,-2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(0,-3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
```

```
set_vals(j);
}

/*cflag=0;
for(j=5;j<=7;j++) {if(pixel[enX[j]][enY[j]]==ca) {cflag=1;break;}}*/
}
}
else if(abs(posflag)==3){
if((c1==ca)||(c2==ca)||(c3==ca)||(c4==ca)||(c5==ca)||(c7==ca)) cflag=1;
else{
if(posflag==3){
Nx+=-2*(n-1);if(Ny==6*(n-1)) Ny+=-6*(n-1);
j=5;rot(5,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(5,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(5,1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
}
else if(posflag==-3){
Nx+=2*(n-1);Ny+=2*(n-1);
j=5;rot(5,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(5,-2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(5,-3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
}


/*cflag=0;
for(j=5;j<=7;j++) {if(pixel[enX[j]][enY[j]]==ca) {cflag=1;break;}}*/
}
}
else if(posflag==4){
if((c1==ca)||(c2==ca)||(c3==ca)||(c4==ca)||(c5==ca)||(c7==ca)) cflag=1;
else{
Nx+=2*(n-1);
if(Ny==0*(n-1)) Ny+=6*(n-1);

j=5;rot(4,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(4,-2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(4,-3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
```

```
/*cflag=0;
for(j=5;j<=7;j++) {if(pixel[enX[j]][enY[j]]==ca) {cflag=1;break;}}*/
}
}
else if(posflag==5){
if((c1==ca)||(c2==ca)||(c3==ca)||(c4==ca)||(c5==ca)||(c7==ca)) cflag=1;
else{
Nx+=1*(n-1);

j=5;rot(3,1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(3,0);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(3,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);

/*cflag=0;
for(j=5;j<=7;j++) {if(pixel[enX[j]][enY[j]]==ca) {cflag=1;break;}}*/
}
}

if(posflag!=-1 && cflag==0){
snold=sn;
sn=sn_;

for(j=5;j<=7;j++){
cc1=/*getpixel_(X_,Y_,enX[j],enY[j]);*/pixel[enX[j]][enY[j]];
if(cc1==ca) {cflag=1;break;}
}

sn=snold;
}


if(cflag){
s.xx=nx[i];s.yy=ny[i];s.xx_=nx_[i];s.yy_=ny_[i];s.sn=sn;fwrite_mem(i);

std_x=nx[i];std_y=ny[i];            /* S */
last_x=nx_[i];last_y=ny_[i];
/*nx_[i]=nx[i];ny_[i]=ny[i];*/          /* new b */



/*if(CPMAX==10){
if(i==5){
```

```
if(algo==0) algo=1;else algo=0;
}
}*/

/* for out */
Nx=nx[i];Ny=ny[i];
/*if(sn==0 && (Nx>=(sn+2)*(n-1) && Ny<=6*(n-1))) dsn=5;
else if(sn==5 && (Nx<(sn+2)*(n-1) || Ny>6*(n-1))) dsn=-5;
else if(sn==5 && (Nx>=(sn+2)*(n-1) && Ny<=6*(n-1))) dsn=5;
else if(sn==10 && (Nx<(sn+2)*(n-1) || Ny>6*(n-1))) dsn=-5;*/

posflag=check_v(Nx,Ny);

if(posflag==-1){
j=-1;

if(c1!=0){
j++;  /* ca1 */
    enX[j]=x[1];enY[j]=y[1];enX_[j]=x_[1];enY_[j]=y_[1];enSN[j]=jmp[1];
}
if(c4!=0){
j++;  /* ca4 */
    enX[j]=x[4];enY[j]=y[4];enX_[j]=x_[4];enY_[j]=y_[4];enSN[j]=jmp[4];
}
if(c7!=0){
j++;  /* ca7 */
    enX[j]=x[7];enY[j]=y[7];enX_[j]=x_[7];enY_[j]=y_[7];enSN[j]=jmp[7];
}
if(c3!=0){
j++;  /* ca3 */
    enX[j]=x[3];enY[j]=y[3];enX_[j]=x_[3];enY_[j]=y_[3];enSN[j]=jmp[3];
}
if(c2!=0){
j++;  /* ca2 */
    enX[j]=x[2];enY[j]=y[2];enX_[j]=x_[2];enY_[j]=y_[2];enSN[j]=jmp[2];
}
if(c5!=0){
j++;  /* ca5 */
    enX[j]=x[5];enY[j]=y[5];enX_[j]=x_[5];enY_[j]=y_[5];enSN[j]=jmp[5];
}

jmax=j;
}
else if(posflag==0){
j=0;  /* ca5 */
    enX[j]=x[5];enY[j]=y[5];enX_[j]=x_[5];enY_[j]=y_[5];enSN[j]=jmp[5];
```

```
j=1;  /* ca1 */
    enX[j]=x[1];enY[j]=y[1];enX_[j]=x_[1];enY_[j]=y_[1];enSN[j]=jmp[1];
j=2;  /* ca4 */
    enX[j]=x[4];enY[j]=y[4];enX_[j]=x_[4];enY_[j]=y_[4];enSN[j]=jmp[4];
j=3;  /* ca7 */
    enX[j]=x[7];enY[j]=y[7];enX_[j]=x_[7];enY_[j]=y_[7];enSN[j]=jmp[7];
j=4;  /* ca3 */
    enX[j]=x[3];enY[j]=y[3];enX_[j]=x_[3];enY_[j]=y_[3];enSN[j]=jmp[3];

                        /* out */
Ny+=-1*(n-1);

j=5;rot(2,1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(2,0);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(2,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);

jmax=7;
}
else if(posflag==1){
j=0;  /* ca1 */
    enX[j]=x[1];enY[j]=y[1];enX_[j]=x_[1];enY_[j]=y_[1];enSN[j]=jmp[1];
j=1;  /* ca4 */
    enX[j]=x[4];enY[j]=y[4];enX_[j]=x_[4];enY_[j]=y_[4];enSN[j]=jmp[4];
j=2;  /* ca7 */
    enX[j]=x[7];enY[j]=y[7];enX_[j]=x_[7];enY_[j]=y_[7];enSN[j]=jmp[7];
j=3;  /* ca3 */
    enX[j]=x[3];enY[j]=y[3];enX_[j]=x_[3];enY_[j]=y_[3];enSN[j]=jmp[3];
j=4;  /* ca2 */
    enX[j]=x[2];enY[j]=y[2];enX_[j]=x_[2];enY_[j]=y_[2];enSN[j]=jmp[2];

                        /* out */
if(Ny==1*(n-1) || Ny==4*(n-1)){
Nx+=-1*(n-1);Ny+=-1*(n-1);
j=5;rot(1,1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(1,0);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(1,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
}
else if(Ny==7*(n-1)){
Ny+=-2*(n-1);
j=5;rot(1,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
```

```
set_vals(j);
j=6;rot(1,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(1,1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
}

jmax=7;
}
else if(posflag==2){
j=0;  /* ca4 */
    enX[j]=x[4];enY[j]=y[4];enX_[j]=x_[4];enY_[j]=y_[4];enSN[j]=jmp[4];
j=1;  /* ca7 */
    enX[j]=x[7];enY[j]=y[7];enX_[j]=x_[7];enY_[j]=y_[7];enSN[j]=jmp[7];
j=2;  /* ca3 */
    enX[j]=x[3];enY[j]=y[3];enX_[j]=x_[3];enY_[j]=y_[3];enSN[j]=jmp[3];
j=3;  /* ca2 */
    enX[j]=x[2];enY[j]=y[2];enX_[j]=x_[2];enY_[j]=y_[2];enSN[j]=jmp[2];
j=4;  /* ca5 */
    enX[j]=x[5];enY[j]=y[5];enX_[j]=x_[5];enY_[j]=y_[5];enSN[j]=jmp[5];

                        /* out */
if(Ny==2*(n-1)){
Nx+=-1*(n-1);
j=5;rot(0,1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(0,0);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(0,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
}
else if(Ny==5*(n-1)){
Ny+=2*(n-1);
j=5;rot(0,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(0,-2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(0,-3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
}

jmax=7;
}
else if(abs(posflag)==3){
j=0;  /* ca7 */
    enX[j]=x[7];enY[j]=y[7];enX_[j]=x_[7];enY_[j]=y_[7];enSN[j]=jmp[7];
```

```
j=1;  /* ca3 */
    enX[j]=x[3];enY[j]=y[3];enX_[j]=x_[3];enY_[j]=y_[3];enSN[j]=jmp[3];
j=2;  /* ca2 */
    enX[j]=x[2];enY[j]=y[2];enX_[j]=x_[2];enY_[j]=y_[2];enSN[j]=jmp[2];
j=3;  /* ca5 */
    enX[j]=x[5];enY[j]=y[5];enX_[j]=x_[5];enY_[j]=y_[5];enSN[j]=jmp[5];
j=4;  /* ca1 */
    enX[j]=x[1];enY[j]=y[1];enX_[j]=x_[1];enY_[j]=y_[1];enSN[j]=jmp[1];

                        /* out */
if(posflag==3){
Nx+=-2*(n-1);if(Ny==6*(n-1)) Ny+=-6*(n-1);
j=5;rot(5,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(5,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(5,1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
}
else if(posflag==-3){
Nx+=2*(n-1);Ny+=2*(n-1);
j=5;rot(5,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(5,-2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(5,-3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
}

jmax=7;
}
else if(posflag==4){
j=0;  /* ca3 */
    enX[j]=x[3];enY[j]=y[3];enX_[j]=x_[3];enY_[j]=y_[3];enSN[j]=jmp[3];
j=1;  /* ca2 */
    enX[j]=x[2];enY[j]=y[2];enX_[j]=x_[2];enY_[j]=y_[2];enSN[j]=jmp[2];
j=2;  /* ca5 */
    enX[j]=x[5];enY[j]=y[5];enX_[j]=x_[5];enY_[j]=y_[5];enSN[j]=jmp[5];
j=3;  /* ca1 */
    enX[j]=x[1];enY[j]=y[1];enX_[j]=x_[1];enY_[j]=y_[1];enSN[j]=jmp[1];
j=4;  /* ca4 */
    enX[j]=x[4];enY[j]=y[4];enX_[j]=x_[4];enY_[j]=y_[4];enSN[j]=jmp[4];

                        /* out */
Nx+=2*(n-1);
if(Ny==0*(n-1)) Ny+=6*(n-1);
```

```
j=5;rot(4,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(4,-2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(4,-3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);

jmax=7;
}
else if(posflag==5){
j=0;  /* ca2 */
    enX[j]=x[2];enY[j]=y[2];enX_[j]=x_[2];enY_[j]=y_[2];enSN[j]=jmp[2];
j=1;  /* ca5 */
    enX[j]=x[5];enY[j]=y[5];enX_[j]=x_[5];enY_[j]=y_[5];enSN[j]=jmp[5];
j=2;  /* ca1 */
    enX[j]=x[1];enY[j]=y[1];enX_[j]=x_[1];enY_[j]=y_[1];enSN[j]=jmp[1];
j=3;  /* ca4 */
    enX[j]=x[4];enY[j]=y[4];enX_[j]=x_[4];enY_[j]=y_[4];enSN[j]=jmp[4];
j=4;  /* ca7 */
    enX[j]=x[7];enY[j]=y[7];enX_[j]=x_[7];enY_[j]=y_[7];enSN[j]=jmp[7];

                        /* out */
Nx+=1*(n-1);

j=5;rot(3,1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=6;rot(3,0);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);
j=7;rot(3,-1);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
set_vals(j);

jmax=7;
}


oldx=nx[i];oldy=ny[i];

for(j=0;j<=jmax;j++){
if(enX[j]==nx_[i] && enY[j]==ny_[i]) {pos=j;break;}
}

if(algo==0){
/* CW */
count=0;
for(j=pos;;){
cc1=pixel[fen("X",j,jmax)][fen("Y",j,jmax)];
```

```
cc2=pixel[fen("X",j-1,jmax)][fen("Y",j-1,jmax)];

/*if(pixel[fen("X",j,jmax)][fen("Y",j,jmax)]!=ca &&
   pixel[fen("X",j-1,jmax)][fen("Y",j-1,jmax)]==ca){*/
if(cc1!=ca && cc2==ca){
nx[i]=fen("X",j-1,jmax);ny[i]=fen("Y",j-1,jmax);
nx_[i]=fen("X_",j-1,jmax);ny_[i]=fen("Y_",j-1,jmax);
if(i<=CPHALF-1) sntmp1=fen("SN",j-1,jmax);
else sntmp2=fen("SN",j-1,jmax);
break;
}

j--;if(j<0) j=jmax;
count++;if(count==jmax+1) {printf(" ?CW\n");break;}
}/**for()**/
}/**if(algo)**/
else{
/* CCW */
count=0;
for(j=pos;;){
cc1=pixel[fen("X",j,jmax)][fen("Y",j,jmax)];
cc2=pixel[fen("X",j+1,jmax)][fen("Y",j+1,jmax)];

/*if(pixel[fen("X",j,jmax)][fen("Y",j,jmax)]!=ca &&
   pixel[fen("X",j+1,jmax)][fen("Y",j+1,jmax)]==ca){*/
if(cc1!=ca && cc2==ca){
nx[i]=fen("X",j+1,jmax);ny[i]=fen("Y",j+1,jmax);
nx_[i]=fen("X_",j+1,jmax);ny_[i]=fen("Y_",j+1,jmax);
if(i<=CPHALF-1) sntmp1=fen("SN",j+1,jmax);
else sntmp2=fen("SN",j+1,jmax);
break;
}

j++;if(j>jmax) j=0;
count++;if(count==jmax+1) {printf(" ?CCW\n");break;}
}/**for()**/
}/**else(algo)**/

/*if(cnt>=282 && rcount[0]>=72 && posflag!=-1){
printf(" i:%d posflag:%d x:%d y:%d Nx:%d Ny:%d\n",i,posflag,oldx,oldy,Nx,Ny);
for(j=0;j<=jmax;j++){
snold=sn;
sn=fen("SN",j,jmax);
Nx=fen("X_",j,jmax);Ny=fen("Y_",j,jmax);
cc1=getpixel_(Nx,Ny,fen("X",j,jmax),fen("Y",j,jmax));
sn=snold;
```

```
printf(" j:%d X:%d Y:%d cc:%d cc1:%d\n",j,enX[j],enY[j],pixel[enX[j]][enY[j]],cc1);
}
}*/

if(1){
if(i<CPHALF-1){
sn=sntmp1;
}
else if(i==CPHALF-1){
sn=sntmp1;
sn1=sntmp1;
}
else if(i>CPHALF-1 && i<CPMAX-1){
sn=sntmp2;
}
else if(i==CPMAX-1){
sn=sntmp2;
sn2=sntmp2;
}
putpixel_(nx[i],ny[i],acolor[i]);
}

flag_pp[i]=1;
}/**if(c1,c2,c3,c4)**/
else{
if(ftell_mem(i)==0) {flag_[i]=0;cp--;if(cp==0) break;}
fread_mem(i);
nx[i]=s.xx;ny[i]=s.yy;nx_[i]=s.xx_;ny_[i]=s.yy_;
if(i<=CPHALF-1) sn1=s.sn;
else sn2=s.sn;
flag_pp[i]=0;
}/**else(c1,c2,c3,c4)**/
}/**if(flag_[i])**/

i++;if(i==CPMAX) break;
}/**while(1)**/

if(0){
}
}/**while(cp)**/

return 0;
}/** cag_r **/
```