

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. 5×12 polyhedron

Figure 1 is a development of dodecahedron and the polygon which composes dodecahedron is regular pentagon.

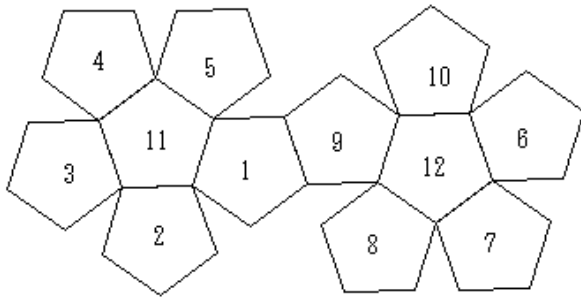


Figure 1

We cannot apply oblique coordinate system to regular pentagon. So, we consider tiling replacing a regular pentagon with a group of five regular triangles. If each regular pentagon of dodecahedron is replaced with the group of five regular triangles like Figure 2, the development of this polyhedron becomes like Figure 3. We call this polyhedron 5×12 polyhedron (polyhedron made up of 5×12 surfaces).



Figure 2

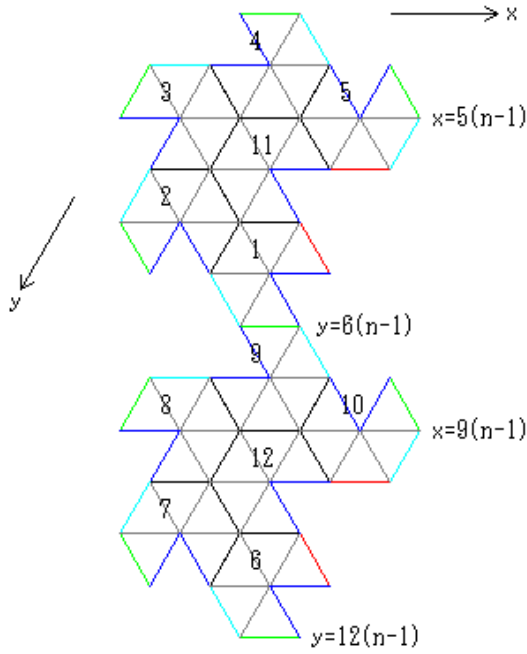


Figure 3

2. Boundary condition

On 5×12 polyhedron, a group of semi-hexagons like Figure 3 is used. In semi-hexagon, one regular triangle is eliminated in comparison with hexagon and we unify the other regular triangles using a boundary condition to the place. We call the side of which color is blue in the figure cut.

A number with an alphabet in Figure 4 is a number with which we distinguish vertex, side, cut of semi-hexagon.

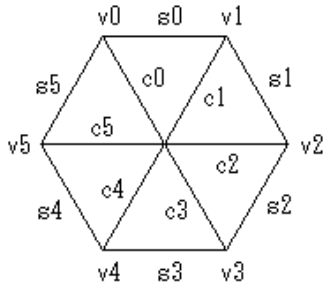


Figure 4

v, s, c represent vertex, side, cut respectively.

A number in Figure 3 is a number of semi-hexagon. We distinguish semi-hexagons adding alphabet h to the number. Vertexes which agree are as follows. Semicolon is symbol for separation.

- h1-v2, h1-v3, h5-v3 ; h1-v4, h9-v5 ; h2-v3, h2-v4, h8-v0 ; h2-v5, h8-v5, h8-v4 ;
h3-v4, h3-v5, h7-v5 ; h3-v0, h7-v4, h7-v3 ; h4-v5, h4-v0, h6-v4 ; h4-v1, h6-v3, h6-v2, h10-v3 ;
h5-v0, h5-v1, h10-v2 ; h5-v2, h10-v0, h10-v1 ;
- h11-v2, h11-v3 ; h12-v2, h12-v3

Two sides which agree are as follows. A description in () represents a direction of a side.

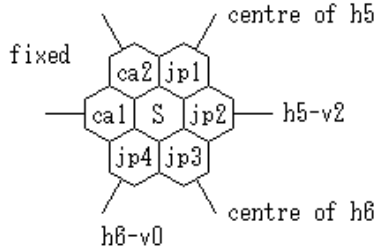
- h1-s4(v4-v5), h8-s0(v1-v0) ; h2-s4(v4-v5), h8-s5(v0-v5);h2-s5(v5-v0), h7-s5(v0-v5) ;
h3-s5(v5-v0), h7-s4(v5-v4) ; h3-s0(v0-v1), h6-s4(v5-v4) ; h4-s0(v0-v1), h6-s3(v4-v3) ;

- $h4-s1(v1-v2), h10-s2(v3-v2) ; h5-s1(v1-v2), h10-s1(v2-v1) ; h5-s2(v2-v3), h9-s1(v2-v1)$
- $h1-s1(v1-v2), h5-s3(v4-v3) ; h10-s3(v4-v3), h6-s1(v1-v2)$

Two cuts which agree are expressed with blue line segment of semi-hexagon in Figure 3.

3. Neighborhood view

Figure 5 is neighborhood view on vertex $h1-v2$.



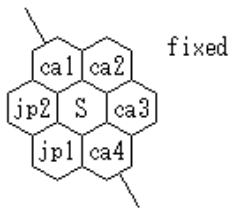
$$\begin{aligned}
 S: & (5(n-1), 5(n-1)) & S': & (X', Y') \\
 X' & = 5(n-1) & Y' & = 3(n-1) \\
 jp1: & (X'-1, Y'-1) \\
 jp2: & (X', Y'-1) \\
 X' & = 5(n-1) & Y' & = 6(n-1) \\
 jp3: & (X'-1, Y'+1) \\
 jp4: & (X'-1, Y'-1) \\
 & h1-v2
 \end{aligned}$$

Figure 5

We jump at $jp1 \sim jp4$. If one of $ca1, ca2, jp1 \sim jp4$ is chosen, the present searching point S in the figure becomes the previous searching point S' . If one of $jp1 \sim jp4$ is chosen, the previous searching point S' jumps too. The coordinates (X', Y') are

- $X' = 5(n - 1), Y' = 3(n - 1) \quad (jp1, jp2)$
- $X' = 5(n - 1), Y' = 6(n - 1) \quad (jp3, jp4)$

Figure 6 is neighborhood view on vertex $h1-s4$.



$$\begin{aligned}
 S: & (x, y) & S': & (X', Y') \\
 X' & = x & Y' & = 7(n-1) \\
 jp1: & (X'+1, Y'+1) \\
 jp2: & (X', Y'+1) \\
 & h1-s4
 \end{aligned}$$

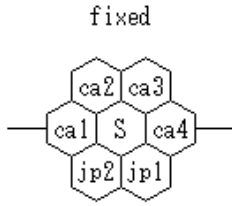
Figure 6

The coordinates (X', Y') of the previous searching point S' after jump are

- $\Delta = x - 3(n - 1)$

- $X' = 3(n - 1) + \Delta$, $Y' = 7(n - 1)$

Figure 7 is neighborhood view on vertex h11-c2.



$$\begin{aligned}
 S: (x, 3(n-1)) & \quad S': (X', Y') \\
 X' = x & \quad Y' = x \\
 jp1: (X', Y'+1) & \\
 jp2: (X'-1, Y') & \\
 & \quad \quad \quad h11-c2
 \end{aligned}$$

Figure 7

The coordinates (X', Y') of the previous searching point S' after jump are

- $X' = x$, $Y' = y + \Delta$

If cut h11-c2, h11-c3, h12-c2, h12-c3, h9-c5, h9-c0 are extended, the neighborhood views on vertexes h11-v2, h11-v3, h12-v2, h12-v3, h9-v5, h9-v0 and sides h5-s3, h1-s1, h10-s3, h6-s1, h8-s0, h1-s4 can be omitted.

4. Logical angle method

As we can understand from the neighborhood view of Figure 5, in 5×12 polyhedron, there are six pixels around searching point S always. If 5×12 polyhedrons are connected, the number exceeds six. Therefore, angle method which had been used until the last cannot be used in this case. Because in this angle method, an angle is calculated actually, we rename this angle method actual angle method.

In logical angle method, not calculating angle, we memorize coordinates and so on of pixels to an array and use the array circularly. We assume that there are six pixels around searching point S like Figure 8.

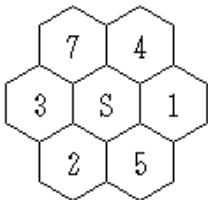


Figure 8

We memorize coordinates (X, Y) of a pixel and coordinates (X', Y') of a previous searching point S' in the order of 1, 4, 7, 3, 2, 5 to the following array.

- $\{j = 0 : X_0, Y_0, X'_0, Y'_0\}$, $\{j = 1 : X_1, Y_1, X'_1, Y'_1\}$, $\{j = 2 : X_2, Y_2, X'_2, Y'_2\}$,
 $\{j = 3 : X_3, Y_3, X'_3, Y'_3\}$, $\{j = 4 : X_4, Y_4, X'_4, Y'_4\}$, $\{j = 5 : X_5, Y_5, X'_5, Y'_5\}$
- $(j_{max} = 5)$

We call the array neighborhood array. Next, we find j' of a previous searching point S' . For example, we assume $j' = 2$. We search a pixel of which color code satisfies the relation of ca-g in the order of $j = 1, 0, 5, 4, 3$ if CW and in the order of $j = 3, 4, 5, 0, 1$ if CCW.

5. Connection of 5×12 polyhedrons(even tunnel)

We consider the case of connection of two 5×12 polyhedrons like Figure 9. For the sake of ease, a regular pentagon is used instead of a group of five regular triangles in the figure.

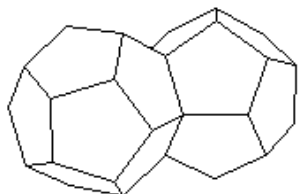


Figure 9

The two 5×12 polyhedrons which have a hole touch on vertexes and sides which form the hole. The development is Figure 10.

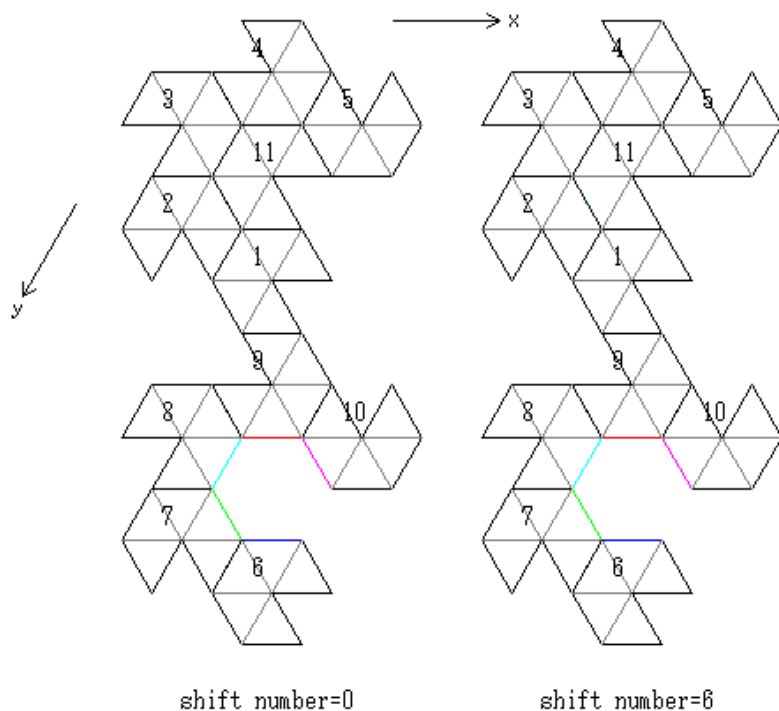
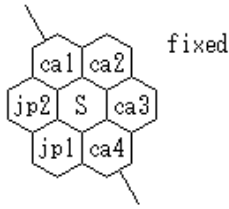


Figure 10

The number of semihexagon is the same. Neighborhood view on the side h10-s4 of the left figure becomes like Figure 11.



$$\begin{aligned}
 S:(x,y) \quad S':(X',Y') \\
 X'=x+6(n-1) \quad Y'=y \\
 jp1:(X'+1,Y') \\
 jp2:(X',Y'-1)
 \end{aligned}$$

h10-s4

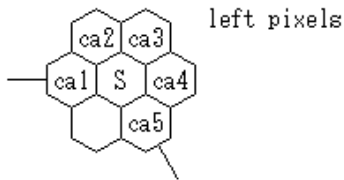
Figure 11

If jp1 or jp2 is chosen, the searching point S jumps to the right figure. The coordinates (X', Y') of the previous searching point S' after jump are

- $\Delta x = 6(n - 1)$
- $X' = x + \Delta x, Y' = y$

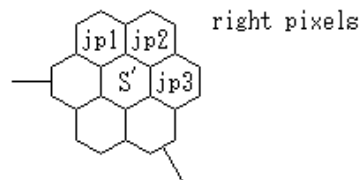
In Figure 10, If the left figure is moved by $6(n - 1)$ in the direction of x axis, the right figure is got. The number 6 is called shift number. If the same choice is done on the side h10-s4 of the right figure, the searching point S jumps to the left figure and the previous searching point S' jumps by $-\Delta x$.

Neighborhood view on the vertex h10-v5 of the left figure becomes like Figure 12.



$$S:(x,y)$$

h10-v5



$$\begin{aligned}
 S':(X',Y') \\
 X'=x+6(n-1) \quad Y'=y \\
 jp1:(X'-1,Y'-1) \\
 jp2:(X',Y'-1) \\
 jp3:(X'+1,Y')
 \end{aligned}$$

Figure 12

We memorise two kinds of coordinates and shift number to neighborhood array in the order of base pixels ca5, ca4, ca3, ca2, ca1, additional pixels jp1, jp2, jp3. We call the row extended neighborhood. j_{max} becomes 7. If one of jp1, jp2, jp3 is chosen, the searching point S jumps to the right figure. The coordinates (X', Y') of the previous searching point S' after jump are

- $\Delta x = 6(n - 1)$
- $X' = x + \Delta x, Y' = y$

If the same choice is done on the vertex h10-v5 of the right figure, the searching point S jumps to the left figure and the previous searching point S' jumps by $-\Delta x$.

In judgement of coordinates on vertex, side, cut in the right figure, we make use of judgement of coordinates on vertex, side, cut in the left figure changing x coordinate by $-(shiftnumber) \times (n - 1)$. If judgement of coordinates is finished, we restore x coordinate changing x coordinate by $(shiftnumber) \times (n - 1)$. The system resembles EMS memory. Namely, there is a correspondence like the following:

- shift number 0 \Leftrightarrow EMS page frame
- shift number 6 \Leftrightarrow logical page

6. Connection of 5×12 polyhedrons(odd tunnel)

Figure 13 is a development in the case of connection of three 5×12 polyhedrons.

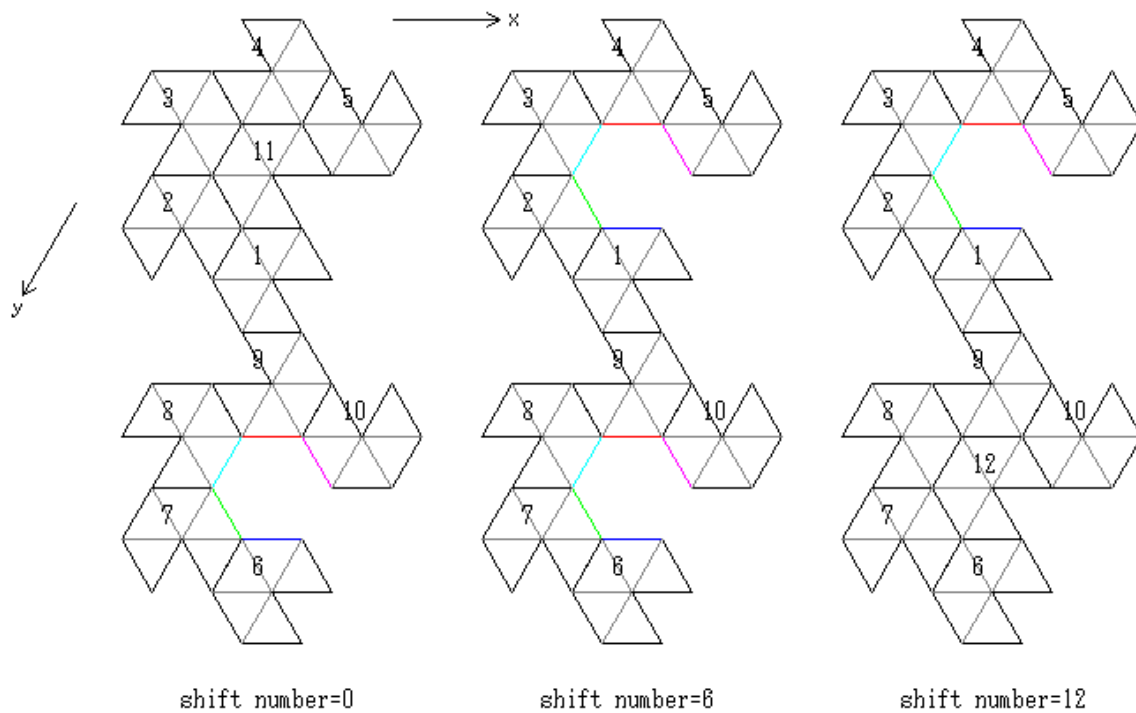


Figure 13

Jump between shift number 0 and shift number 6 is done at vertexes and sides which form the lower hole and jump between shift number 6 and shift number 12 is done at vertexes and sides which form the upper hole. In shift number 12, we make use of judgement of coordinates on shift number 0 in the same way as shift number 6 changing x coordinate by $-(shiftnumber) \times (n - 1)$ and if judgement of coordinates is finished, we restore x coordinate changing x coordinate by $(shiftnumber) \times (n - 1)$.

7. Assignment

Complete neighborhood views on vertexes, sides and cuts of the figure of Figure 13.

8. Concrete example

Figure 14 is a symmetrical graphics by Figure 13 and the following are data of program.

- $SP(5 \times 2):5(\text{left h11})+5(\text{right h12}):h?-c$
- $n = 4$
- coordinates of painting number 1 : painting point a: $(4(n - 1), 4(n - 1))$
 painting point b: $(3(n - 1), 4(n - 1))$ 、 painting point c: $(2(n - 1), 3(n - 1))$
 painting point d: $(2(n - 1), 2(n - 1))$ 、 painting point e: $(3(n - 1), 2(n - 1))$

painting point f: $((4 + 3 + 12)(n - 1), (4 + 6)(n - 1))$

painting point g: $((3 + 3 + 12)(n - 1), (4 + 6)(n - 1))$

painting point h: $((2 + 3 + 12)(n - 1), (3 + 6)(n - 1))$

painting point i: $((2 + 3 + 12)(n - 1), (2 + 6)(n - 1))$

painting point j: $((3 + 3 + 12)(n - 1), (2 + 6)(n - 1))$

- coordinates of painting number 2 : painting point a: $\Delta x = -1$
painting point b: $\Delta x = -1; \Delta y = -1$, painting point c: $\Delta y = -1$
painting point d: $\Delta x = 1$, painting point e: $\Delta x = 1; \Delta y = 1$
painting point f: $\Delta x = -1$
painting point g: $\Delta x = -1; \Delta y = -1$
painting point h: $\Delta y = -1$
painting point i: $\Delta x = 1$
painting point j: $\Delta x = 1; \Delta y = 1$

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

Lump painting cannot be used on tunnel(odd tunnel, even tunnel).

Array which is used for painting is initialized as follows:

- target pixel : 15
- centre $(3(n - 1), 3(n - 1))$, $((3 + 3 + 12)(n - 1), (3 + 6)(n - 1))$ of semi-hexagon : -1
- wall pixel : 0

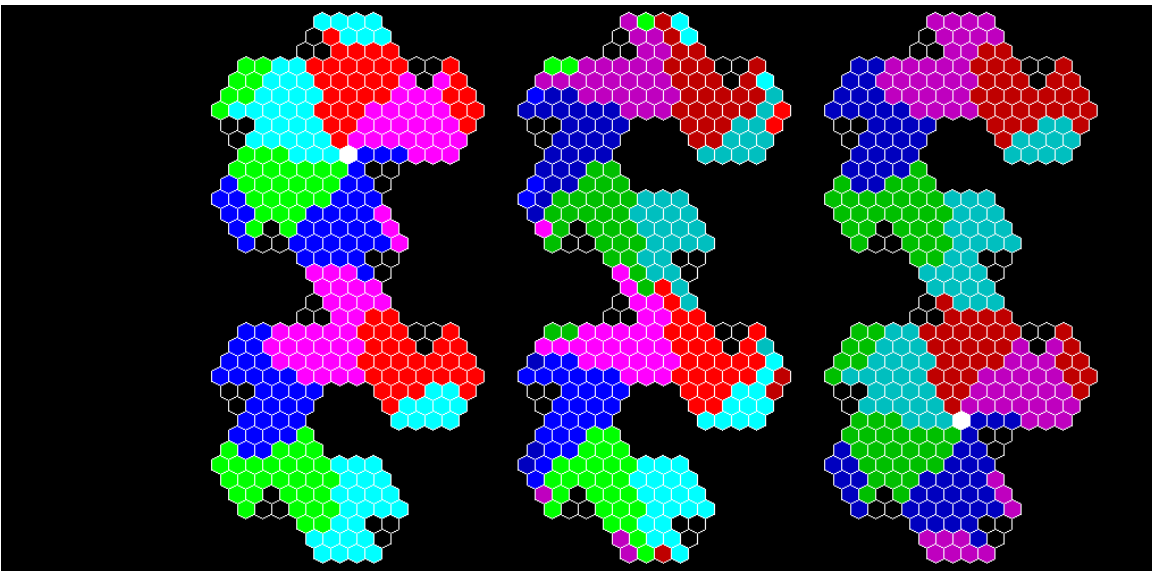


Figure 14

セルラーオートマトングラフィクス (4)

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. 5×12 面体

図 1 は正 12 面体の展開図であり、正 12 面体を構成する多角形は正五角形です。

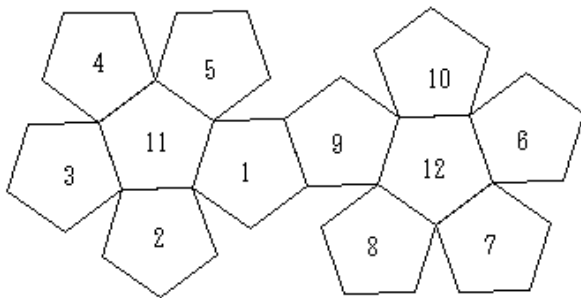


図 1

正五角形には斜交座標を張ることができません。そこで正五角形を正三角形の集まりに置き換えて敷き詰めを考えます。正 12 面体の各正五角形を図 2 のような五つの正三角形の集まりに置き換えると、この多面体の展開図は図 3 のようになります。この多面体を 5×12 面体と称します。

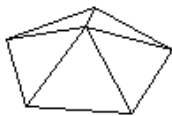


図 2

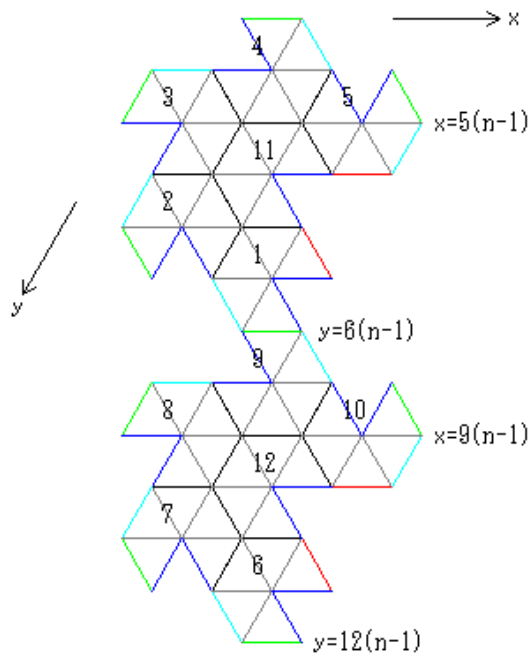


図 3

2. 境界条件

5 × 12 面体では準正六角形の周りに準正六角形を五つ配置したものを二つ用います。準正六角形では準正六角形に比して正三角形が一つ除去されており、除去された個所に境界条件を用いて残りの正三角形を一体化します。正三角形が除去されたところにある一方の端が準正六角形の中心である辺をカットと称します。

図 4 中のアルファベット付の数字は準正六角形の頂点、辺、カットを区別する番号です。

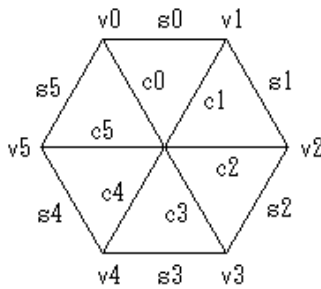


図 4

v、s、c は各々頂点、辺、カットを表します。

図 3 中の数字は準正六角形の番号です。これにアルファベット h を付して準正六角形を区別します。一致する頂点は以下のようになっています。セミコロンは区切りを示すシンボルです。

- h1-v2, h1-v3, h5-v3 ; h1-v4, h9-v5 ; h2-v3, h2-v4, h8-v0 ; h2-v5, h8-v5, h8-v4 ;
h3-v4, h3-v5, h7-v5 ; h3-v0, h7-v4, h7-v3 ; h4-v5, h4-v0, h6-v4 ; h4-v1, h6-v3, h6-v2, h10-v3 ;
h5-v0, h5-v1, h10-v2 ; h5-v2, h10-v0, h10-v1 ;
- h11-v2, h11-v3 ; h12-v2, h12-v3

一致する 2 辺は以下のようになっています。() 内は辺の向きを表します。

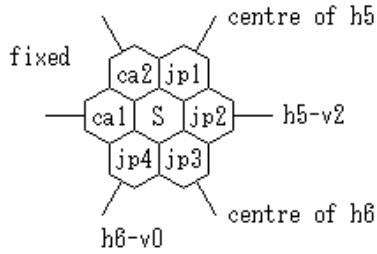
- h1-s4(v4-v5), h8-s0(v1-v0) ; h2-s4(v4-v5), h8-s5(v0-v5); h2-s5(v5-v0), h7-s5(v0-v5) ;
h3-s5(v5-v0), h7-s4(v5-v4) ; h3-s0(v0-v1), h6-s4(v5-v4) ; h4-s0(v0-v1), h6-s3(v4-v3) ;

- $h4-s1(v1-v2), h10-s2(v3-v2) ; h5-s1(v1-v2), h10-s1(v2-v1) ; h5-s2(v2-v3), h9-s1(v2-v1)$
- $h1-s1(v1-v2), h5-s3(v4-v3) ; h10-s3(v4-v3), h6-s1(v1-v2)$

一致する 2 カットは図 3 において準正六角形の青い線分で表されます。

3. 近傍図

図 5 は頂点 $h1-v2$ に関する近傍図です。



$$\begin{aligned}
 S &: (5(n-1), 5(n-1)) & S' &: (X', Y') \\
 X' &= 5(n-1) & Y' &= 3(n-1) \\
 jp1 &: (X'-1, Y'-1) \\
 jp2 &: (X', Y'-1) \\
 X' &= 5(n-1) & Y' &= 6(n-1) \\
 jp3 &: (X'-1, Y'+1) \\
 jp4 &: (X'-1, Y'-1) \\
 h1-v2 &
 \end{aligned}$$

図 5

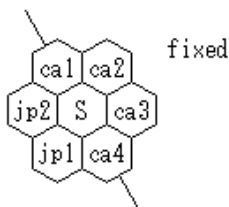
$jp1 \sim jp4$ ではジャンプします。 $ca1, ca2, jp1 \sim jp4$ のいずれかが選択されれば、図中の現在の探索点 S は一つ前の探索点 S' となります。もし、 $jp1 \sim jp4$ のいずれかが選択されれば、一つ前の探索点 S' もジャンプします。その座標 (X', Y') は、 $jp1, jp2$ では

- $X' = 5(n - 1), Y' = 3(n - 1)$

$jp3, jp4$ では

- $X' = 5(n - 1), Y' = 6(n - 1)$

図 6 は辺 $h1-s4$ に関する近傍図です。



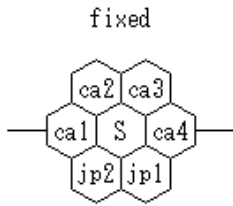
$$\begin{aligned}
 S &: (x, y) & S' &: (X', Y') \\
 X' &= x & Y' &= 7(n-1) \\
 jp1 &: (X'+1, Y'+1) \\
 jp2 &: (X', Y'+1) \\
 h1-s4 &
 \end{aligned}$$

図 6

ジャンプ後の一つ前の探索点 S' の座標 (X', Y') は

- $\Delta = x - 3(n - 1)$
- $X' = 3(n - 1) + \Delta, Y' = 7(n - 1)$

図 7 はカット h11-c2 に関する近傍図です。



$$\begin{aligned}
 S: (x, 3(n-1)) & \quad S': (X', Y') \\
 X' = x & \quad Y' = x \\
 jp1: (X', Y'+1) & \\
 jp2: (X'-1, Y') & \\
 & \quad h11-c2
 \end{aligned}$$

図 7

ジャンプ後の一つ前の探索点 S' の座標 (X', Y') は

- $X' = x, Y' = y + \Delta$

カット h11-c2、h11-c3、カット h12-c2、h12-c3、カット h9-c5、h9-c0 は延長可能であり、それぞれ、頂点 h11-v2、h11-v3 と辺 h5-s3、h1-s1 に関する近傍図、頂点 h12-v2、h12-v3 と辺 h10-s3、h6-s1 に関する近傍図、頂点 h9-v5、h9-v0 と辺 h8-s0、h1-s4 に関する近傍図が省略可能です。

4. 論理角度法

図 5 の近傍図からわかるように、 5×12 面体では探索点 S の周りにはピクセルが常に六個あります。 5×12 面体を連結するとこの数は六個を越えます。したがって、前回まで用いていた角度法はこの場合は使えません。この角度法では実際に角度を算出するので、この角度法を実角度法とリネームします。

論理角度法では角度を算出するのではなく、配列にピクセルの座標等を記憶し、この配列を環状に利用します。図 8 のように探索点 S の周りに六個のピクセルがあるとします。

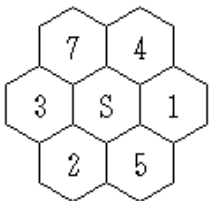


図 8

これらのピクセルの座標 (X, Y) 、一つ前の探索点 S' の座標 (X', Y') を 1、4、7、3、2、5 の順に以下のような配列に記憶します。

- $\{j = 0: X_0, Y_0, X'_0, Y'_0\}$ 、 $\{j = 1: X_1, Y_1, X'_1, Y'_1\}$ 、 $\{j = 2: X_2, Y_2, X'_2, Y'_2\}$ 、 $\{j = 3: X_3, Y_3, X'_3, Y'_3\}$ 、 $\{j = 4: X_4, Y_4, X'_4, Y'_4\}$ 、 $\{j = 5: X_5, Y_5, X'_5, Y'_5\}$
- $(j_{max} = 5)$

この配列を近傍配列と称します。次に、一つ前の探索点 S' の j' を求めます。たとえば、 $j' = 2$ とします。カラーコードが ca-g の関係を満たすピクセルを、CW では $j = 1, 0, 5, 4, 3$ の順に、CCW では $j = 3, 4, 5, 0, 1$ の順にサーチします。

5. 5×12 面体の連結 (偶トンネル)

図9のように 5×12 面体を2連結した場合を考えます。簡単のため、図では五つの正三角形の集まりの代わりに正五角形を用いています。

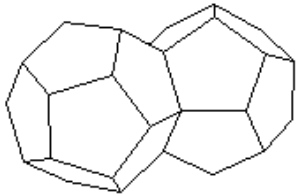


図 9

穴のあいた二つの 5×12 面体は穴を形成する頂点と辺で接しています。展開図は図10です。

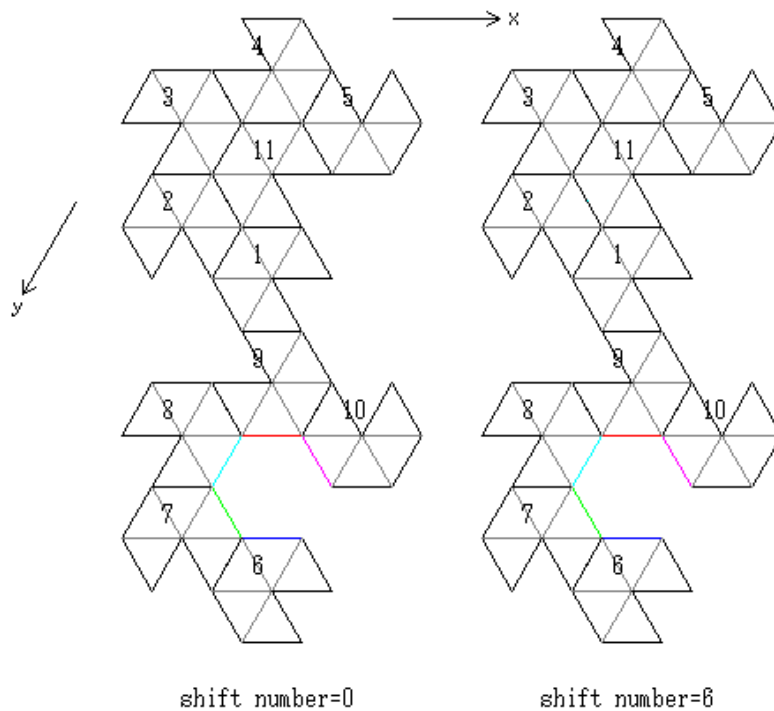
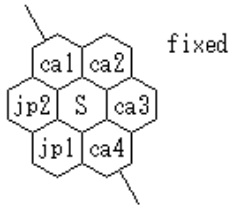


図 10

準正六角形の番号は同じです。左の図形の辺 h_{10-s_4} における近傍図は図11のようになります。



$$\begin{aligned}
 S: (x, y) \quad S': (X', Y') \\
 X' = x + 6(n-1) \quad Y' = y \\
 jp1: (X'+1, Y') \\
 jp2: (X', Y'-1) \\
 h10-s4
 \end{aligned}$$

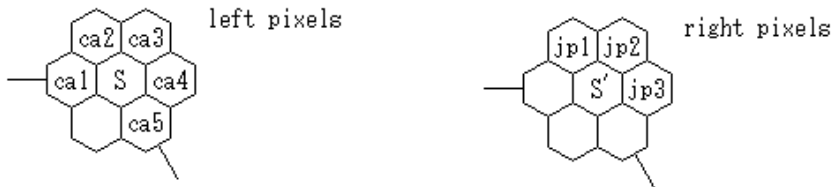
図 11

jp1 または jp2 が選択されれば探索点 S は右の図形にジャンプします。ジャンプ後の一つ前の探索点 S' の座標 (X', Y') は

- $\Delta x = 6(n - 1)$
- $X' = x + \Delta x, Y' = y$

図 10 において、左の図形を x 軸方向に $6(n - 1)$ 移動させると右の図形になります。6 をシフト数と称します。右の図形の辺 h10-s4 において同様な選択がなされれば、探索点 S は左の図形にジャンプし、一つ前の探索点 S' は $-\Delta x$ だけジャンプします。

左の図形の頂点 h10-v5 における近傍図は図 12 のようになります。



$$\begin{aligned}
 S: (x, y) \qquad S': (X', Y') \\
 X' = x + 6(n-1) \quad Y' = y \\
 jp1: (X'-1, Y'-1) \\
 jp2: (X', Y'-1) \\
 jp3: (X'+1, Y') \\
 h10-v5
 \end{aligned}$$

図 12

近傍配列には二種類の座標とシフト数を、基礎ピクセル ca5、ca4、ca3、ca2、ca1、付加ピクセル jp1、jp2、jp3 の順に記憶します。このピクセルの並びを拡張近傍と称します。 j_{max} は 7 となります。jp1、jp2、jp3 のいずれかが選択されれば探索点 S は右の図形にジャンプします。ジャンプ後の一つ前の探索点 S' の座標 (X', Y') は

- $\Delta x = 6(n - 1)$
- $X' = x + \Delta x, Y' = y$

右の図形の頂点 h10-v5 において同様な選択がなされれば、探索点 S は左の図形にジャンプし、一つ前の探索点 S' は $-\Delta x$ だけジャンプします。

右の図形の頂点、辺、カットにおける座標判定では、 $-シフト数 \times (n - 1)$ だけ x 座標を変化させて左の図形の頂点、辺、カットにおける座標判定を利用します。座標判定が終わればシフト数 $\times (n - 1)$ だけ x 座標を変化させて x 座標を復元します。この仕組みは EMS メモリに似ています。すなわち

- ・シフト数 0 \Leftrightarrow EMS ページフレーム
- ・シフト数 6 \Leftrightarrow 論理ページ

のような対応があります。

6. 5×12 面体の連結 (奇トンネル)

5×12 面体を 3 連結した場合の展開図は図 13 です。

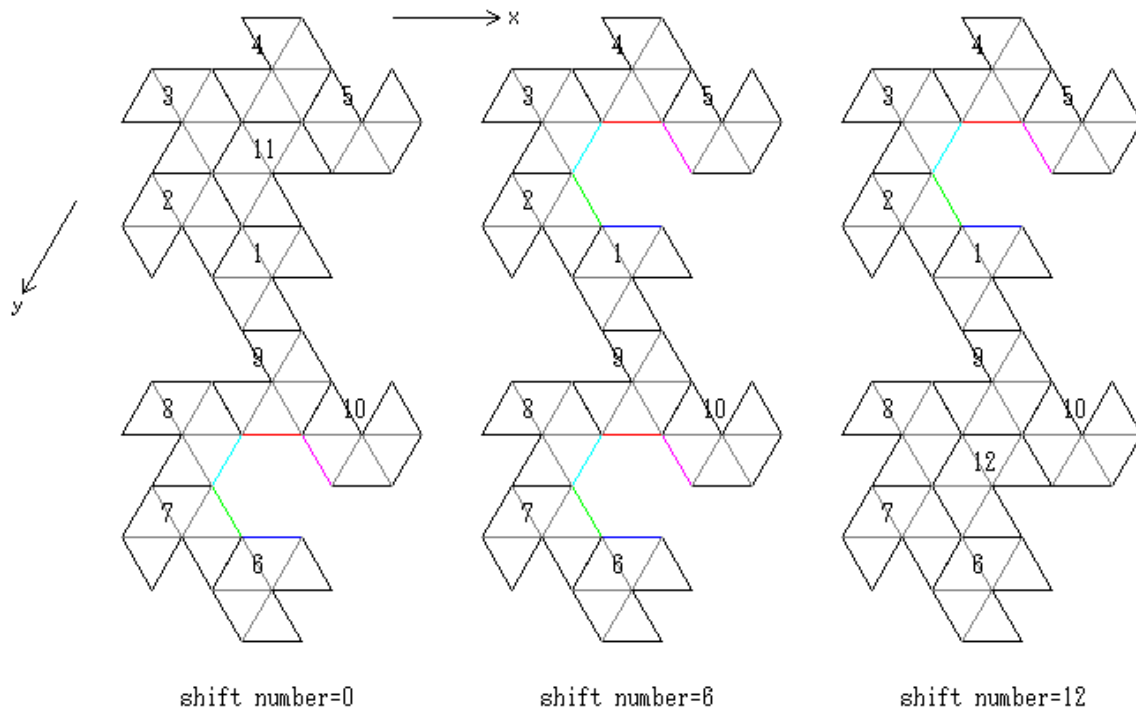


図 13

シフト数 0 とシフト数 6 の間のジャンプは下の穴を形成する頂点と辺において、シフト数 6 とシフト数 12 の間のジャンプは上の穴を形成する頂点と辺において行います。シフト数 12 においては、 $-シフト数 \times (n - 1)$ だけ x 座標を変化させてシフト数 6 と同様、シフト数 0 における座標判定を利用し、座標判定が終わればシフト数 $\times (n - 1)$ だけ x 座標を変化させて x 座標を復元します。

7. 課題

図 13 の図形の頂点、辺、カットに関する近傍図を完成させてください。

8. 具体例

図 14 は図 13 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。

- ・ $SP(5 \times 2):5(\text{left h11})+5(\text{right h12}):h?-c$
- ・ $n = 4$
- ・ 塗番号 1 の座標：塗点 a: $(4(n - 1), 4(n - 1))$ 、塗点 b: $(3(n - 1), 4(n - 1))$
 塗点 c: $(2(n - 1), 3(n - 1))$ 、塗点 d: $(2(n - 1), 2(n - 1))$ 、塗点 e: $(3(n - 1), 2(n - 1))$
 塗点 f: $((4 + 3 + 12)(n - 1), (4 + 6)(n - 1))$ 、塗点 g: $((3 + 3 + 12)(n - 1), (4 + 6)(n - 1))$
 塗点 h: $((2 + 3 + 12)(n - 1), (3 + 6)(n - 1))$ 、塗点 i: $((2 + 3 + 12)(n - 1), (2 + 6)(n - 1))$

塗点 j: $((3 + 3 + 12)(n - 1), (2 + 6)(n - 1))$

- 塗番号 2 の座標 : 塗点 a: $\Delta x = -1$ 、塗点 b: $\Delta x = -1; \Delta y = -1$
塗点 c: $\Delta y = -1$ 、塗点 d: $\Delta x = 1$ 、塗点 e: $\Delta x = 1; \Delta y = 1$
塗点 f: $\Delta x = -1$ 、塗点 g: $\Delta x = -1; \Delta y = -1$
塗点 h: $\Delta y = -1$ 、塗点 i: $\Delta x = 1$
塗点 j: $\Delta x = 1; \Delta y = 1$

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

トンネル (奇トンネル、偶トンネル) では一括塗りつぶしは使えません。
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
- ・準正六角形の中心 $(3(n - 1), 3(n - 1))$, $((3 + 3 + 12)(n - 1), (3 + 6)(n - 1)) : -1$
- ・壁ピクセル : 0

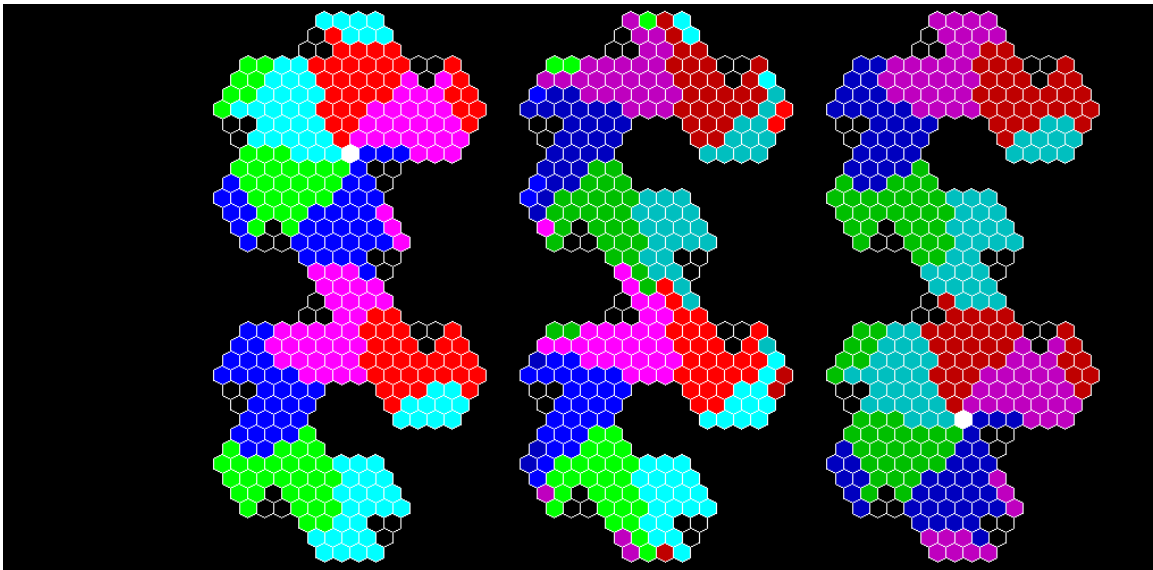


図 14

List 1:cag_4.c

```
/* t2.26 */
/* 2018 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 /*5*//*10*/10 /* number of painting points */

#if CPMAX==10
#define DIV 1
#else
#define DIV 0
#endif

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

#define X0 (230)
#define Y0 (10)
#define dyMAX 603
#define Odd /*0*//*1*//*3*/3
#define RESO /*4*//*6*//*8*/4 /* n */
#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_;
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),BitBlit_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)
{
int val;
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(0,0,12);
field(0,0,6);
field(0,0,0);
cnt=0;

/*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;

val=1;                                     /* 0:no hall, 1:15, 2:8 */

while(1){
        field(val,0,12);
if(Odd==3) field(val,val,6); /* for odd */
else      field(0,val,6); /* for even */
if(Odd!=1) field(0,val,0);
else      field(0,0,0);
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;
}/**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);
}

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

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.lpfWndProc   =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);

/*EndPoint(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++){
    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,XRES0,YRES0,0,0);
}/** 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;
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(nx==0 || ny==0 || nx==RESO-1 || ny==RESO-1) pcolor=12;*/

```

```

hbrush=CreateSolidBrush(PALETTE(pcolor));

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:
pixel[nx][ny]=pcolor;

return 0;
}/** putpixel **/

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){
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_hex(int n,int x,int y)
{
int i,j,XDP,YDP,begin;

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;
}
}/** 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 field(int upper,int lower,int begin)
{
int n;

fieldflag=1;

n=RES0;
sn=begin;
tmp0=15;

field_hex(n,(sn+1)*(n-1),0*(n-1));
field_hex(n,(sn+0)*(n-1),1*(n-1));
if(upper==0) field_hex(n,(sn+2)*(n-1),2*(n-1));
field_hex(n,(sn+3)*(n-1),1*(n-1));
field_hex(n,(sn+1)*(n-1),3*(n-1));
field_hex(n,(sn+3)*(n-1),4*(n-1));

field_hex(n,(sn+0+3)*(n-1),(1+6)*(n-1));

field_hex(n,(sn+1+3)*(n-1),(0+6)*(n-1));
if(lower==0) field_hex(n,(sn+2+3)*(n-1),(2+6)*(n-1));
field_hex(n,(sn+3+3)*(n-1),(1+6)*(n-1));
field_hex(n,(sn+1+3)*(n-1),(3+6)*(n-1));
if(0) field_hex(n,(sn+4+3)*(n-1),(3+6)*(n-1));

```

```

field_hex(n, (sn+3+3)*(n-1), (4+6)*(n-1));

putdelta(n, 0, (sn+1)*(n-1), 0*(n-1), 0, 0);
putdelta(n, 1, (sn+3)*(n-1), 1*(n-1), 0, 0);
putdelta(n, 1, (sn+0)*(n-1), 2*(n-1), 0, 1);
if(upper==0) putdelta(n, 1, (sn+3)*(n-1), 3*(n-1), 1, 1);
putdelta(n, 0, (sn+2)*(n-1), 4*(n-1), 1, 1);
putdelta(n, 1, (sn+4)*(n-1), 5*(n-1), 1, 1);

putdelta(n, 0, (sn+1+3)*(n-1), (0+6)*(n-1), 0, 0);
putdelta(n, 1, (sn+3+3)*(n-1), (1+6)*(n-1), 0, 0);
/*if(lower==0) putdelta(n, 0, (sn+5)*(n-1), 8*(n-1), 0, 0);*/
putdelta(n, 1, (sn+0+3)*(n-1), (2+6)*(n-1), 0, 1);
putdelta(n, 0, (sn+2+3)*(n-1), (4+6)*(n-1), 1, 1);
if(lower==0) putdelta(n, 1, (sn+3+3)*(n-1), (3+6)*(n-1), 1, 1);
putdelta(n, 1, (sn+4+3)*(n-1), (5+6)*(n-1), 1, 1);

if(cnt>-1){
pixel[(sn+3)*(n-1)][3*(n-1)]=-1;
pixel[(sn+3+3)*(n-1)][(3+6)*(n-1)]=-1;
}

if(upper>1){
putpixel((sn+3-1)*(n-1), (3-1)*(n-1), 8);
putpixel((sn+3-0)*(n-1), (3-1)*(n-1), 8);
putpixel((sn+3+1)*(n-1), (3-0)*(n-1), 8);
putpixel((sn+3+1)*(n-1), (3+1)*(n-1), 8);
putpixel((sn+3+0)*(n-1), (3+1)*(n-1), 8);
putpixel((sn+3-1)*(n-1), (3-0)*(n-1), 8);
}
else if(upper==2){
putpixel((sn+3-1)*(n-1), (3-1)*(n-1), 15);
putpixel((sn+3-0)*(n-1), (3-1)*(n-1), 15);
putpixel((sn+3+1)*(n-1), (3-0)*(n-1), 15);
putpixel((sn+3+1)*(n-1), (3+1)*(n-1), 15);
putpixel((sn+3+0)*(n-1), (3+1)*(n-1), 15);
putpixel((sn+3-1)*(n-1), (3-0)*(n-1), 15);
}

if(lower>1){
putpixel((sn+3+3-1)*(n-1), (3+6-1)*(n-1), 8);
putpixel((sn+3+3-0)*(n-1), (3+6-1)*(n-1), 8);
putpixel((sn+3+3+1)*(n-1), (3+6-0)*(n-1), 8);
putpixel((sn+3+3+1)*(n-1), (3+6+1)*(n-1), 8);
putpixel((sn+3+3+0)*(n-1), (3+6+1)*(n-1), 8);
putpixel((sn+3+3-1)*(n-1), (3+6-0)*(n-1), 8);
}

```

```

}
else if(lower==-2){
putpixel((sn+3+3-1)*(n-1),(3+6-1)*(n-1),15);
putpixel((sn+3+3-0)*(n-1),(3+6-1)*(n-1),15);
putpixel((sn+3+3+1)*(n-1),(3+6-0)*(n-1),15);
putpixel((sn+3+3+1)*(n-1),(3+6+1)*(n-1),15);
putpixel((sn+3+3+0)*(n-1),(3+6+1)*(n-1),15);
putpixel((sn+3+3-1)*(n-1),(3+6-0)*(n-1),15);
}

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

void pp_(int nx,int ny,int pcolor)
{
nx+=sn;
nx*=RES0-1;ny*=RES0-1;
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 side(int h,int side,int nx,int ny)

```

```
{
```

```
int n;
```

```
n=RES0;
```

```

if(h==1 && side==4){
if(nx>=3*(n-1)+1 && nx<=4*(n-1)-1 && ny==nx+2*(n-1)) return 1;else return 0;
}
else if(h==8 && side==0){
if(nx>=3*(n-1)+1 && nx<=4*(n-1)-1 && ny==7*(n-1)) return 1;else return 0;
}
else if(h==2 && side==4){
if(nx>=1*(n-1)+1 && nx<=2*(n-1)-1 && ny==nx+3*(n-1)) return 1;else return 0;
}
else if(h==8 && side==5){
if(nx==3*(n-1) && ny>=7*(n-1)+1 && ny<=8*(n-1)-1) return 1;else return 0;
}
else if(h==2 && side==5){
if(nx==1*(n-1) && ny>=3*(n-1)+1 && ny<=4*(n-1)-1) return 1;else return 0;
}
else if(h==7 && side==5){
if(nx==4*(n-1) && ny>=9*(n-1)+1 && ny<=10*(n-1)-1) return 1;else return 0;
}
else if(h==3 && side==5){
if(nx==0*(n-1) && ny>=1*(n-1)+1 && ny<=2*(n-1)-1) return 1;else return 0;
}

```

```

}
else if(h==7 && side==4){
if(nx>=4*(n-1)+1 && nx<=5*(n-1)-1 && ny==nx+6*(n-1)) return 1;else return 0;
}
else if(h==3 && side==0){
if(nx>=0*(n-1)+1 && nx<=1*(n-1)-1 && ny==1*(n-1)) return 1;else return 0;
}
else if(h==6 && side==4){
if(nx>=6*(n-1)+1 && nx<=7*(n-1)-1 && ny==nx+5*(n-1)) return 1;else return 0;
}
else if(h==4 && side==0){
if(nx>=1*(n-1)+1 && nx<=2*(n-1)-1 && ny==0*(n-1)) return 1;else return 0;
}
else if(h==6 && side==3){
if(nx>=7*(n-1)+1 && nx<=8*(n-1)-1 && ny==12*(n-1)) return 1;else return 0;
}
else if(h==4 && side==1){
if(nx>=2*(n-1)+1 && nx<=3*(n-1)-1 && ny==nx-2*(n-1)) return 1;else return 0;
}
else if(h==10 && side==2){
if(nx==8*(n-1) && ny>=8*(n-1)+1 && ny<=9*(n-1)-1) return 1;else return 0;
}
else if(h==5 && side==1){
if(nx>=4*(n-1)+1 && nx<=5*(n-1)-1 && ny==nx-3*(n-1)) return 1;else return 0;
}
else if(h==10 && side==1){
if(nx>=7*(n-1)+1 && nx<=8*(n-1)-1 && ny==nx-0*(n-1)) return 1;else return 0;
}
else if(h==5 && side==2){
if(nx==5*(n-1) && ny>=2*(n-1)+1 && ny<=3*(n-1)-1) return 1;else return 0;
}
else if(h==9 && side==1){
if(nx>=5*(n-1)+1 && nx<=6*(n-1)-1 && ny==nx+1*(n-1)) return 1;else return 0;
}
/*OK*/

else if(h==12 && side==0){
if(nx>=5*(n-1)+1 && nx<=6*(n-1)-1 && ny==8*(n-1)) return 1;else return 0;
}
else if(h==12 && side==1){
if(nx>=6*(n-1)+1 && nx<=7*(n-1)-1 && ny==nx+2*(n-1)) return 1;else return 0;
}
else if(h==12 && side==3){
if(nx>=6*(n-1)+1 && nx<=7*(n-1)-1 && ny==10*(n-1)) return 1;else return 0;
}
else if(h==12 && side==4){

```

```

if(nx>=5*(n-1)+1 && nx<=6*(n-1)-1 && ny==nx+4*(n-1)) return 1;else return 0;
}
else if(h==12 && side==5){
if(nx==5*(n-1) && ny>=8*(n-1)+1 && ny<=9*(n-1)-1) return 1;else return 0;
}

else if(h==11 && side==0){
if(nx>=2*(n-1)+1 && nx<=3*(n-1)-1 && ny==2*(n-1)) return 1;else return 0;
}
else if(h==11 && 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==11 && side==3){
if(nx>=3*(n-1)+1 && nx<=4*(n-1)-1 && ny==4*(n-1)) return 1;else return 0;
}
else if(h==11 && side==4){
if(nx>=2*(n-1)+1 && nx<=3*(n-1)-1 && ny==nx+1*(n-1)) return 1;else return 0;
}
else if(h==11 && side==5){
if(nx==2*(n-1) && ny>=2*(n-1)+1 && ny<=3*(n-1)-1) return 1;else return 0;
}

return 0;
}/** side **/

```

```

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

n=RES0;

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

```

```

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

else if(h==6 && cut==2){
if(nx>=7*(n-1)+1 && nx<=8*(n-1)-1 && ny==11*(n-1)) return 1;else return 0;
}
else if(h==6 && cut==3){
if(nx>=7*(n-1)+1 && nx<=8*(n-1)-1 && ny==nx+4*(n-1)) return 1;else return 0;
}
else if(h==7 && cut==3){
if(nx>=5*(n-1)+1 && nx<=6*(n-1)-1 && ny==nx+5*(n-1)) return 1;else return 0;
}
else if(h==7 && cut==4){
if(nx==5*(n-1) && ny>=10*(n-1)+1 && ny<=11*(n-1)-1) return 1;else return 0;
}
else if(h==8 && cut==4){
if(nx==4*(n-1) && ny>=8*(n-1)+1 && ny<=9*(n-1)-1) return 1;else return 0;
}
else if(h==8 && cut==5){
if(nx>=3*(n-1)+1 && nx<=4*(n-1)-1 && ny==8*(n-1)) return 1;else return 0;
}
else if(h==9 && cut==5){
/* wide */
if(nx>=3*(n-1)+1 && nx<=5*(n-1)-1 && ny==7*(n-1)) return 1;else return 0;
}
else if(h==9 && cut==0){
/* wide */
if(nx>=3*(n-1)+1 && nx<=5*(n-1)-1 && ny==nx+2*(n-1)) return 1;else return 0;
}
else if(h==10 && cut==0){
if(nx>=6*(n-1)+1 && nx<=7*(n-1)-1 && ny==nx+1*(n-1)) return 1;else return 0;
}

```



```

else if(h==10 && cut==1){
if(nx==7*(n-1) && ny>=7*(n-1)+1 && ny<=8*(n-1)-1) return 1;else return 0;
}

else if(h==11 && cut==2){
/* wide */
if(nx>=3*(n-1)+1 && nx<=5*(n-1)-1 && ny==3*(n-1)) return 1;else return 0;
}
else if(h==11 && cut==3){
/* wide */
if(nx>=3*(n-1)+1 && nx<=5*(n-1)-1 && ny==nx+0*(n-1)) return 1;else return 0;
}

else if(h==12 && cut==2){
/* wide */
if(nx>=6*(n-1)+1 && nx<=8*(n-1)-1 && ny==9*(n-1)) return 1;else return 0;
}
else if(h==12 && cut==3){
/* wide */
if(nx>=6*(n-1)+1 && nx<=8*(n-1)-1 && ny==nx+3*(n-1)) return 1;else return 0;
}
/*OK*/

return 0;
}/** cut **/

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

n=RES0;

if(h==1 && v==5){
if(nx==3*(n-1) && ny==5*(n-1)) return 1;else return 0;
}
else if(h==2 && v==3){
if(nx==3*(n-1) && ny==5*(n-1)) return 1;else return 0;
}
else if(h==2 && v==4){
if(nx==2*(n-1) && ny==5*(n-1)) return 1;else return 0;
}
else if(h==10 && v==5){
if(nx==6*(n-1) && ny==8*(n-1)) return 1;else return 0;
}
}

```

```

else if(h==2 && v==0){
if(nx==1*(n-1) && ny==3*(n-1)) return 1;else return 0;
}
else if(h==3 && v==4){
if(nx==1*(n-1) && ny==3*(n-1)) return 1;else return 0;
}
else if(h==3 && v==5){
if(nx==0*(n-1) && ny==2*(n-1)) return 1;else return 0;
}
else if(h==7 && v==5){
if(nx==4*(n-1) && ny==10*(n-1)) return 1;else return 0;
}
else if(h==9 && v==4){
if(nx==5*(n-1) && ny==8*(n-1)) return 1;else return 0;
}
else if(h==3 && v==1){
if(nx==1*(n-1) && ny==1*(n-1)) return 1;else return 0;
}
else if(h==4 && v==0){
if(nx==1*(n-1) && ny==0*(n-1)) return 1;else return 0;
}
else if(h==4 && v==5){
if(nx==1*(n-1) && ny==1*(n-1)) return 1;else return 0;
}
else if(h==8 && v==0){
if(nx==3*(n-1) && ny==7*(n-1)) return 1;else return 0;
}
else if(h==8 && v==5){
if(nx==3*(n-1) && ny==8*(n-1)) return 1;else return 0;
}
else if(h==4 && v==2){
if(nx==3*(n-1) && ny==1*(n-1)) return 1;else return 0;
}
else if(h==5 && v==1){
if(nx==4*(n-1) && ny==1*(n-1)) return 1;else return 0;
}
else if(h==7 && v==2){
if(nx==8*(n-1) && ny==8*(n-1)) return 1;else return 0;
}
else if(h==5 && v==3){
if(nx==5*(n-1) && ny==3*(n-1)) return 1;else return 0;
}
else if(h==1 && v==2){
if(nx==5*(n-1) && ny==5*(n-1)) return 1;else return 0;
}
else if(h==1 && v==3){

```

```

if(nx==5*(n-1) && ny==6*(n-1)) return 1;else return 0;
}
else if(h==6 && v==1){
if(nx==5*(n-1) && ny==6*(n-1)) return 1;else return 0;
}
else if(h==11 && v==2){
if(nx==4*(n-1) && ny==3*(n-1)) return 1;else return 0;
}
else if(h==11 && v==3){
if(nx==4*(n-1) && ny==4*(n-1)) return 1;else return 0;
}

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

else if(h==6 && v==2){
if(nx==8*(n-1) && ny==11*(n-1)) return 1;else return 0;
}
else if(h==6 && v==3){
if(nx==8*(n-1) && ny==12*(n-1)) return 1;else return 0;
}
else if(h==7 && v==1){
if(nx==7*(n-1) && ny==7*(n-1)) return 1;else return 0;
}
else if(h==5 && v==2){
if(nx==5*(n-1) && ny==2*(n-1)) return 1;else return 0;
}
else if(h==7 && v==3){
if(nx==6*(n-1) && ny==11*(n-1)) return 1;else return 0;
}
else if(h==4 && v==1){
if(nx==2*(n-1) && ny==0*(n-1)) return 1;else return 0;
}
else if(h==8 && v==4){
if(nx==4*(n-1) && ny==9*(n-1)) return 1;else return 0;
}

```

```

}
else if(h==9 && v==2){
if(nx==6*(n-1) && ny==7*(n-1)) return 1;else return 0;
}
else if(h==9 && v==3){
if(nx==6*(n-1) && ny==8*(n-1)) return 1;else return 0;
}
else if(h==3 && v==0){
if(nx==0*(n-1) && ny==1*(n-1)) return 1;else return 0;
}
else if(h==7 && v==4){
if(nx==5*(n-1) && ny==11*(n-1)) return 1;else return 0;
}
else if(h==9 && v==5){
if(nx==4*(n-1) && ny==7*(n-1)) return 1;else return 0;
}
else if(h==10 && v==1){
if(nx==7*(n-1) && ny==7*(n-1)) return 1;else return 0;
}
else if(h==10 && v==2){
if(nx==8*(n-1) && ny==8*(n-1)) return 1;else return 0;
}
else if(h==10 && v==3){
if(nx==8*(n-1) && ny==9*(n-1)) return 1;else return 0;
}
else if(h==10 && v==4){
if(nx==7*(n-1) && ny==9*(n-1)) return 1;else return 0;
}
else if(h==2 && v==5){
if(nx==1*(n-1) && ny==4*(n-1)) return 1;else return 0;
}
else if(h==10 && v==0){
if(nx==6*(n-1) && ny==7*(n-1)) return 1;else return 0;
}
else if(h==6 && v==4){
if(nx==7*(n-1) && ny==12*(n-1)) return 1;else return 0;
}
else if(h==6 && v==5){
if(nx==4*(n-1) && ny==7*(n-1)) return 1;else return 0;
}
else if(h==1 && v==4){
if(nx==4*(n-1) && ny==6*(n-1)) return 1;else return 0;
}
else if(h==12 && v==5){
if(nx==5*(n-1) && ny==9*(n-1)) return 1;else return 0;
}
}

```

```

else if(h==12 && v==0){
if(nx==5*(n-1) && ny==8*(n-1)) return 1;else return 0;
}

else if(h==12 && v==1){
if(nx==6*(n-1) && ny==8*(n-1)) return 1;else return 0;
}
else if(h==12 && v==2){
if(nx==7*(n-1) && ny==9*(n-1)) return 1;else return 0;
}
else if(h==12 && v==3){
if(nx==7*(n-1) && ny==10*(n-1)) return 1;else return 0;
}
else if(h==12 && v==4){
if(nx==6*(n-1) && ny==10*(n-1)) return 1;else return 0;
}

return 0;
}/** v **/

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

if(cnt<0) return;
n=RES0;

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

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

/* h1-v2, h1-v3, h5-v3 */
if(v(1,2,nx,ny)==1 || v(1,3,nx,ny)==1 || v(5,3,nx,ny)==1){
pp_(5,5,pcolor);pp_(5,6,pcolor);pp_(5,3,pcolor);
}
/* h1-v4, h9-v5 */
else if(0){
}
/* h2-v3, h2-v4, h8-v0 */
else if(v(2,3,nx,ny)==1 || v(2,4,nx,ny)==1 || v(8,0,nx,ny)==1){
pp_(3,5,pcolor);pp_(2,5,pcolor);pp_(3,7,pcolor);
}
}

```

```

/* h2-v5, h8-v5, h8-v4 */
else if(v(2,5,nx,ny)==1 || v(8,5,nx,ny)==1 || v(8,4,nx,ny)==1){
pp_(1,4,pcolor);pp_(3,8,pcolor);pp_(4,9,pcolor);
}
/* h3-v4, h3-v5, h7-v5 */
else if(v(3,4,nx,ny)==1 || v(3,5,nx,ny)==1 || v(7,5,nx,ny)==1){
pp_(1,3,pcolor);pp_(0,2,pcolor);pp_(4,10,pcolor);
}
/* h3-v0, h7-v4, h7-v3 */
else if(v(3,0,nx,ny)==1 || v(7,4,nx,ny)==1 || v(7,3,nx,ny)==1){
pp_(0,1,pcolor);pp_(5,11,pcolor);pp_(6,11,pcolor);
}
/* h4-v5, h4-v0, h6-v4 */
else if(v(4,5,nx,ny)==1 || v(4,0,nx,ny)==1 || v(6,4,nx,ny)==1){
pp_(1,1,pcolor);pp_(1,0,pcolor);pp_(7,12,pcolor);
}
/* h4-v1, h6-v3, h6-v2, h10-v3 */
else if(v(4,1,nx,ny)==1 || v(6,3,nx,ny)==1 || v(6,2,nx,ny)==1 || v(10,3,nx,ny)==1){
pp_(2,0,pcolor);pp_(8,12,pcolor);pp_(8,11,pcolor);pp_(8,9,pcolor);
}
/* no modification */
/* h5-v0, h5-v1, h10-v2 */
else if(v(4,2,nx,ny)==1 || v(5,1,nx,ny)==1 || v(10,2,nx,ny)==1){
pp_(3,1,pcolor);pp_(4,1,pcolor);pp_(8,8,pcolor);
}
/* h5-v2, h10-v0, h10-v1 */
else if(v(9,2,nx,ny)==1 || v(10,1,nx,ny)==1 || v(5,2,nx,ny)==1){
pp_(6,7,pcolor);pp_(7,7,pcolor);pp_(5,2,pcolor);
}

/*999*/
else if(v(12,2,nx,ny)==1 || v(12,3,nx,ny)==1){
pp_(7,9,pcolor);pp_(7,10,pcolor);

#if Odd!=1
if(sn<12){
snold=sn;
if(sn==0) sn=6;
else if(sn==6) sn=0;

pp_(7,9,pcolor);pp_(7,10,pcolor);
sn=snold;
}
#endif
}
#endif

```

```

else if(v(12,0,nx,ny)==1 && sn<12){
snold=sn;
if(sn==0) sn=6;
else if(sn==6) sn=0;

pp_(5,8,pcolor);
sn=snold;
}
else if(v(12,1,nx,ny)==1 && sn<12){
snold=sn;
if(sn==0) sn=6;
else if(sn==6) sn=0;

pp_(6,8,pcolor);
sn=snold;
}
else if(v(12,4,nx,ny)==1 && sn<12){
snold=sn;
if(sn==0) sn=6;
else if(sn==6) sn=0;

pp_(6,10,pcolor);
sn=snold;
}
else if(v(12,5,nx,ny)==1 && sn<12){
snold=sn;
if(sn==0) sn=6;
else if(sn==6) sn=0;

pp_(5,9,pcolor);
sn=snold;
}
#endif

else if(v(11,2,nx,ny)==1 || v(11,3,nx,ny)==1){
pp_(4,3,pcolor);pp_(4,4,pcolor);

/* for odd */
#if Odd==3
if(sn>0){
snold=sn;
if(sn==6) sn=12;
else if(sn==12) sn=6;

pp_(4,3,pcolor);pp_(4,4,pcolor);
sn=snold;

```

```

}
#endif
}
/* for odd */
#if Odd==3
else if(v(11,0,nx,ny)==1 && sn>0){
snold=sn;
if(sn==6) sn=12;
else if(sn==12) sn=6;

pp_(2,2,pcolor);
sn=snold;
}
else if(v(11,1,nx,ny)==1 && sn>0){
snold=sn;
if(sn==6) sn=12;
else if(sn==12) sn=6;

pp_(3,2,pcolor);
sn=snold;
}
else if(v(11,4,nx,ny)==1 && sn>0){
snold=sn;
if(sn==6) sn=12;
else if(sn==12) sn=6;

pp_(3,4,pcolor);
sn=snold;
}
else if(v(11,5,nx,ny)==1 && sn>0){
snold=sn;
if(sn==6) sn=12;
else if(sn==12) sn=6;

pp_(2,3,pcolor);
sn=snold;
}
#endif

else{
/* if(side(1,4,x,y)) flag=101;
else if(side(8,0,x,y)) flag=102;*/
if(side(2,4,nx,ny)) flag=103;
else if(side(8,5,nx,ny)) flag=104;
else if(side(2,5,nx,ny)) flag=105;
else if(side(7,5,nx,ny)) flag=106;

```



```

else if(side(3,5,nx,ny)) flag=107;
else if(side(7,4,nx,ny)) flag=108;
else if(side(3,0,nx,ny)) flag=109;
else if(side(6,4,nx,ny)) flag=110;
else if(side(4,0,nx,ny)) flag=111;
else if(side(6,3,nx,ny)) flag=112;
/* no modification */
else if(side(4,1,nx,ny)) flag=113;
else if(side(10,2,nx,ny)) flag=114;
else if(side(5,1,nx,ny)) flag=115;
else if(side(10,1,nx,ny)) flag=116;
else if(side(5,2,nx,ny)) flag=117;
else if(side(9,1,nx,ny)) flag=118;

#if Odd!=1
else if(side(12,0,nx,ny)==1 && sn<12) flag=119;
else if(side(12,1,nx,ny)==1 && sn<12) flag=120;
else if(side(12,3,nx,ny)==1 && sn<12) flag=121;
else if(side(12,4,nx,ny)==1 && sn<12) flag=122;
else if(side(12,5,nx,ny)==1 && sn<12) flag=123;
#endif

/* for odd */
#if Odd==3
else if(side(11,0,nx,ny)==1 && sn>0) flag=124;
else if(side(11,1,nx,ny)==1 && sn>0) flag=125;
else if(side(11,3,nx,ny)==1 && sn>0) flag=126;
else if(side(11,4,nx,ny)==1 && sn>0) flag=127;
else if(side(11,5,nx,ny)==1 && sn>0) flag=128;
#endif

else if(cut(1,2,nx,ny)) flag=1001;
else if(cut(1,3,nx,ny)) flag=1002;
else if(cut(2,3,nx,ny)) flag=1003;
else if(cut(2,4,nx,ny)) flag=1004;
else if(cut(3,4,nx,ny)) flag=1005;
else if(cut(3,5,nx,ny)) flag=1006;
else if(cut(4,5,nx,ny)) flag=1007;
else if(cut(4,0,nx,ny)) flag=1008;
else if(cut(5,0,nx,ny)) flag=1009;
else if(cut(5,1,nx,ny)) flag=1010;
else if(cut(11,2,nx,ny)) flag=1011;
else if(cut(11,3,nx,ny)) flag=1012;

else if(cut(6,2,nx,ny)) flag=1013;

```

```

else if(cut(6,3,nx,ny)) flag=1014;
else if(cut(7,3,nx,ny)) flag=1015;
else if(cut(7,4,nx,ny)) flag=1016;
else if(cut(8,4,nx,ny)) flag=1017;
else if(cut(8,5,nx,ny)) flag=1018;
else if(cut(9,5,nx,ny)) flag=1019;
else if(cut(9,0,nx,ny)) flag=1020;
else if(cut(10,0,nx,ny)) flag=1021;
else if(cut(10,1,nx,ny)) flag=1022;
else if(cut(12,2,nx,ny)) flag=1023;
else if(cut(12,3,nx,ny)) flag=1024;

else flag=-1;

if(flag==0){
}
else if(flag>0){
if(flag==101){
}
else if(flag==102){
}
else if(flag==103){
/*pcolor+=nx-1*(n-1);*/
dlt=nx-1*(n-1); nx=3*(n-1);ny=8*(n-1)-dlt;
}
else if(flag==104){
/*pcolor+=nx-4*(n-1);*/
dlt=ny-7*(n-1); nx=2*(n-1)-dlt;ny=5*(n-1)-dlt;
}
else if(flag==105){
/*pcolor+=ny-3*(n-1);*/
dlt=ny-3*(n-1); nx=4*(n-1);ny=10*(n-1)-dlt;
}
else if(flag==106){
/*pcolor+=nx-6*(n-1);*/
dlt=ny-9*(n-1); nx=1*(n-1);ny=4*(n-1)-dlt;
}
else if(flag==107){
/*pcolor+=ny-1*(n-1);*/
dlt=ny-1*(n-1); nx=5*(n-1)-dlt;ny=11*(n-1)-dlt;
}
else if(flag==108){
/*pcolor+=nx-7*(n-1);*/
dlt=nx-4*(n-1); nx=0*(n-1);ny=2*(n-1)-dlt;
}
else if(flag==109){

```

```

/*pcolor+=nx-0*(n-1);*/
dlt=nx-0*(n-1); nx=6*(n-1)+dlt;ny=11*(n-1)+dlt;
}
else if(flag==110){
/*pcolor+=nx-8*(n-1);*/
dlt=nx-6*(n-1); nx=0*(n-1)+dlt;ny=1*(n-1);
}
else if(flag==111){
/*pcolor+=nx-1*(n-1);*/
dlt=nx-1*(n-1); nx=7*(n-1)+dlt;ny=12*(n-1);
}
else if(flag==112){
/*pcolor+=ny-10*(n-1);*/
dlt=nx-7*(n-1); nx=1*(n-1)+dlt;ny=0*(n-1);
}
else if(flag==113){
/*pcolor+=nx-2*(n-1);*/
dlt=nx-2*(n-1);nx=8*(n-1);ny=9*(n-1)-dlt;
}
else if(flag==114){
/*pcolor+=ny-8*(n-1);*/
dlt=ny-8*(n-1);nx=3*(n-1)-dlt;ny=1*(n-1)-dlt;
}
else if(flag==115){
/*pcolor+=nx-4*(n-1);*/
dlt=nx-4*(n-1);nx=8*(n-1)-dlt;ny=8*(n-1)-dlt;
}
else if(flag==116){
/*pcolor+=nx-7*(n-1);*/
dlt=nx-7*(n-1);nx=5*(n-1)-dlt;ny=2*(n-1)-dlt;
}
else if(flag==117){
/*pcolor+=ny-2*(n-1);*/
dlt=ny-2*(n-1);nx=6*(n-1)-dlt;ny=7*(n-1)-dlt;
}
else if(flag==118){
/*pcolor+=nx-5*(n-1);*/
dlt=nx-5*(n-1);nx=5*(n-1);ny=3*(n-1)-dlt;
}

else if(flag>=119 && flag<=123){ /* h12 */
snold=sn;
if(sn==0) sn=6;
else if(sn==6) sn=0;
}
else if(flag>=124 && flag<=128){ /* h11 */

```

```

snold=sn;
if(sn==6) sn=12;
else if(sn==12) sn=6;
}

else if(flag==1001){
/*pcolor+=nx-4*(n-1);*/
dlt=nx-4*(n-1);nx=nx;ny+=dlt;
}
else if(flag==1002){
/*pcolor+=nx-4*(n-1);*/
nx=nx;ny=5*(n-1);
}
else if(flag==1003){
/*pcolor+=nx-2*(n-1);*/
nx=2*(n-1);ny=ny;
}
else if(flag==1004){
/*pcolor+=ny-4*(n-1);*/
dlt=ny-4*(n-1);nx+=dlt;ny=ny;
}
else if(flag==1005){
/*pcolor+=ny-2*(n-1);*/
dlt=ny-2*(n-1);nx-=dlt;ny=2*(n-1);
}
else if(flag==1006){
/*pcolor+=1*(n-1)-nx;*/
dlt=1*(n-1)-nx;nx=1*(n-1);ny+=dlt;
}
else if(flag==1007){
/*pcolor+=2*(n-1)-nx;*/
dlt=2*(n-1)-nx;nx=nx;ny-=dlt;
}
else if(flag==1008){
/*pcolor+=2*(n-1)-nx;*/
nx=nx;ny=1*(n-1);
}
else if(flag==1009){
/*pcolor+=4*(n-1)-nx;*/
nx=4*(n-1);ny=ny;
}
else if(flag==1010){
/*pcolor+=2*(n-1)-ny;*/
dlt=2*(n-1)-ny;nx-=dlt;ny=ny;
}

```

```

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

else if(flag==1013){
/*pcolor+=5*(n-1)-nx;*/
dlt=nx-7*(n-1);nx=nx;ny+=dlt;
}
else if(flag==1014){
/*pcolor+=5*(n-1)-nx;*/
nx=nx;ny=11*(n-1);
}
else if(flag==1015){
/*pcolor+=7*(n-1)-nx;*/
nx=5*(n-1);ny=ny;
}
else if(flag==1016){
/*pcolor+=8*(n-1)-ny;*/
dlt=ny-10*(n-1);nx+=dlt;ny=ny;
}
else if(flag==1017){
/*pcolor+=10*(n-1)-ny;*/
dlt=ny-8*(n-1);nx-=dlt;ny=8*(n-1);
}
else if(flag==1018){
/*pcolor+=nx-8*(n-1);*/
dlt=4*(n-1)-nx;nx=4*(n-1);ny+=dlt;
}
else if(flag==1019){
/*pcolor+=nx-7*(n-1);*/
dlt=5*(n-1)-nx;nx=nx;ny-=dlt;
}
else if(flag==1020){
/*pcolor+=nx-7*(n-1);*/
nx=nx;ny=7*(n-1);
}
else if(flag==1021){
/*pcolor+=nx-5*(n-1);*/
nx=7*(n-1);ny=ny;
}
else if(flag==1022){

```

```

/*pcolor+=ny-10*(n-1);*/
dlt=8*(n-1)-ny;nx-=dlt;ny=ny;
}
else if(flag==1023){
/*pcolor-=6*(n-1)-nx;*/
dlt=nx-6*(n-1);nx=nx;ny+=dlt;
}
else if(flag==1024){
/*pcolor-=6*(n-1)-nx;*/
nx=nx;ny=9*(n-1);
}

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

if(flag>=119 && flag<=123){ /* h12 */
sn=snold;
}
else if(flag>=124 && flag<=128){ /* h11 */
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;*/

n=RES0;

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

/* h1-v2, h1-v3, h5-v3 */

```

```

if(v(1,2,x,y)) flag=1;
else if(v(1,3,x,y)) flag=2;
else if(v(5,3,x,y)) flag=3;
/* h1-v4, h9-v5 */
/* h2-v3, h2-v4, h8-v0 */
else if(v(2,3,x,y)) flag=4;
else if(v(2,4,x,y)) flag=5;
else if(v(8,0,x,y)) flag=6;
/* h2-v5, h8-v5, h8-v4 */
else if(v(2,5,x,y)) flag=7;
else if(v(8,5,x,y)) flag=8;
else if(v(8,4,x,y)) flag=9;
/* h3-v4, h3-v5, h7-v5 */
else if(v(3,4,x,y)) flag=10;
else if(v(3,5,x,y)) flag=11;
else if(v(7,5,x,y)) flag=12;
/* h3-v0, h7-v4, h7-v3 */
else if(v(3,0,x,y)) flag=13;
else if(v(7,4,x,y)) flag=14;
else if(v(7,3,x,y)) flag=15;
/* h4-v5, h4-v0, h6-v4 */
else if(v(4,5,x,y)) flag=16;
else if(v(4,0,x,y)) flag=17;
else if(v(6,4,x,y)) flag=18;
/* h4-v1, h6-v3, h6-v2, h10-v3 */
else if(v(4,1,x,y)) flag=19;
else if(v(6,3,x,y)) flag=20;
else if(v(6,2,x,y)) flag=21;
else if(v(10,3,x,y)) flag=22;
/* no modification */
/* h5-v0, h5-v1, h10-v2 */
else if(v(4,2,x,y)) flag=23;
else if(v(5,1,x,y)) flag=24;
else if(v(10,2,x,y)) flag=25;
/* h5-v2, h10-v0, h10-v1 */
else if(v(9,2,x,y)) flag=26;
else if(v(10,1,x,y)) flag=27;
else if(v(5,2,x,y)) flag=28; /* 28:15+13 */

/* if(side(1,4,x,y)) flag=101;
else if(side(8,0,x,y)) flag=102;*/
else if(side(2,4,x,y)) flag=103;
else if(side(8,5,x,y)) flag=104;
else if(side(2,5,x,y)) flag=105;
else if(side(7,5,x,y)) flag=106;
else if(side(3,5,x,y)) flag=107;

```

```

else if(side(7,4,x,y)) flag=108;
else if(side(3,0,x,y)) flag=109;
else if(side(6,4,x,y)) flag=110;
else if(side(4,0,x,y)) flag=111;
else if(side(6,3,x,y)) flag=112;
/* no modification */
else if(side(4,1,x,y)) flag=113;
else if(side(10,2,x,y)) flag=114;
else if(side(5,1,x,y)) flag=115;
else if(side(10,1,x,y)) flag=116;
else if(side(5,2,x,y)) flag=117;
else if(side(9,1,x,y)) flag=118;

#if Odd!=1
else if(side(12,0,x,y)==1 && sn<12) flag=119;
else if(side(12,1,x,y)==1 && sn<12) flag=120;
else if(side(12,3,x,y)==1 && sn<12) flag=121;
else if(side(12,4,x,y)==1 && sn<12) flag=122;
else if(side(12,5,x,y)==1 && sn<12) flag=123;
#endif

/* for odd */
#if Odd==3
else if(side(11,0,x,y)==1 && sn>0) flag=124;
else if(side(11,1,x,y)==1 && sn>0) flag=125;
else if(side(11,3,x,y)==1 && sn>0) flag=126;
else if(side(11,4,x,y)==1 && sn>0) flag=127;
else if(side(11,5,x,y)==1 && sn>0) flag=128;
#endif

else if(cut(1,2,x,y)) flag=1001;
else if(cut(1,3,x,y)) flag=1002;
else if(cut(2,3,x,y)) flag=1003;
else if(cut(2,4,x,y)) flag=1004;
else if(cut(3,4,x,y)) flag=1005;
else if(cut(3,5,x,y)) flag=1006;
else if(cut(4,5,x,y)) flag=1007;
else if(cut(4,0,x,y)) flag=1008;
else if(cut(5,0,x,y)) flag=1009;
else if(cut(5,1,x,y)) flag=1010;
else if(cut(11,2,x,y)) flag=1011;
else if(cut(11,3,x,y)) flag=1012;

else if(cut(6,2,x,y)) flag=1013;
else if(cut(6,3,x,y)) flag=1014;

```



```

else if(cut(7,3,x,y)) flag=1015;
else if(cut(7,4,x,y)) flag=1016;
else if(cut(8,4,x,y)) flag=1017;
else if(cut(8,5,x,y)) flag=1018;
else if(cut(9,5,x,y)) flag=1019;
else if(cut(9,0,x,y)) flag=1020;
else if(cut(10,0,x,y)) flag=1021;
else if(cut(10,1,x,y)) flag=1022;
else if(cut(12,2,x,y)) flag=1023;
else if(cut(12,3,x,y)) flag=1024;

else flag=0;

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

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

```

```

else if(nh(x,y,nx,ny,5)) {x=3*(n-1);y=5*(n-1);X=x+1;Y=y;jmpflag=1;}
}
else if(flag==7){ /* v(2,5, */
    if(nh(x,y,nx,ny,2)) {x=4*(n-1);y=9*(n-1);X=x;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=4*(n-1);y=9*(n-1);X=x+1;Y=y;jmpflag=1;}
else 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==8){ /* v(8,5, */
    if(nh(x,y,nx,ny,1)) {x=4*(n-1);y=9*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,2)) {x=4*(n-1);y=9*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=1*(n-1);y=4*(n-1);X=x;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=1*(n-1);y=4*(n-1);X=x+1;Y=y;jmpflag=1;}
}
else if(flag==9){ /* v(8,4, */
    if(nh(x,y,nx,ny,3)) {x=1*(n-1);y=4*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=1*(n-1);y=4*(n-1);X=x+1;Y=y+1;jmpflag=1;}
}
else if(flag==10){ /* v(3,4, */
    if(nh(x,y,nx,ny,3)) {x=4*(n-1);y=10*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=4*(n-1);y=10*(n-1);X=x+1;Y=y+1;jmpflag=1;}
}
else if(flag==11){ /* v(3,5, */
    if(nh(x,y,nx,ny,1)) {x=1*(n-1);y=3*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,2)) {x=1*(n-1);y=3*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=4*(n-1);y=10*(n-1);X=x;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=4*(n-1);y=10*(n-1);X=x+1;Y=y;jmpflag=1;}
}
else if(flag==12){ /* v(7,5, */
    if(nh(x,y,nx,ny,2)) {x=1*(n-1);y=3*(n-1);X=x;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=1*(n-1);y=3*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=1*(n-1);y=3*(n-1);X=x+1;Y=y+1;jmpflag=1;}
}
else if(flag==13){ /* v(3,0, */
    if(nh(x,y,nx,ny,3)) {x=6*(n-1);y=11*(n-1);X=x-1;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=6*(n-1);y=11*(n-1);X=x;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,5)) {x=6*(n-1);y=11*(n-1);X=x+1;Y=y;jmpflag=1;}
}
else if(flag==14){ /* v(7,4, */
    if(nh(x,y,nx,ny,0)) {x=6*(n-1);y=11*(n-1);X=x;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=6*(n-1);y=11*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,2)) {x=0*(n-1);y=1*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=0*(n-1);y=1*(n-1);X=x+1;Y=y+1;jmpflag=1;}
}
else if(flag==15){ /* v(7,3, */
    if(nh(x,y,nx,ny,2)) {x=0*(n-1);y=1*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=0*(n-1);y=1*(n-1);X=x;Y=y+1;jmpflag=1;}
}

```

```

}
else if(flag==16){ /* v(4,5, */
    if(nh(x,y,nx,ny,4)) {x=7*(n-1);y=12*(n-1);X=x;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,5)) {x=7*(n-1);y=12*(n-1);X=x+1;Y=y;jmpflag=1;}
}
else if(flag==17){ /* v(4,0, */
    if(nh(x,y,nx,ny,2)) {x=1*(n-1);y=1*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=1*(n-1);y=1*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=7*(n-1);y=12*(n-1);X=x-1;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,5)) {x=7*(n-1);y=12*(n-1);X=x;Y=y-1;jmpflag=1;}
}
else if(flag==18){ /* v(6,4, */
    if(nh(x,y,nx,ny,1)) {x=1*(n-1);y=1*(n-1);X=x+1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,2)) {x=1*(n-1);y=1*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,3)) {x=1*(n-1);y=1*(n-1);X=x;Y=y+1;jmpflag=1;}
}
else if(flag==19){ /* v(4,1, */
    if(nh(x,y,nx,ny,4)) {x=8*(n-1);y=11*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,5)) {x=8*(n-1);y=11*(n-1);X=x-1;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=8*(n-1);y=9*(n-1);X=x-1;Y=y-1;jmpflag=1;}
}
else if(flag==20){ /* v(6,3, */
    if(nh(x,y,nx,ny,5)) {x=8*(n-1);y=9*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=8*(n-1);y=9*(n-1);X=x-1;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=2*(n-1);y=0*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,2)) {x=2*(n-1);y=0*(n-1);X=x;Y=y+1;jmpflag=1;}
}
else if(flag==21){ /* v(6,2, */
    if(nh(x,y,nx,ny,5)) {x=8*(n-1);/*Y*/y=9*(n-1);X=x-1;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=8*(n-1);y=9*(n-1);X=x;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=2*(n-1);y=0*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,2)) {x=2*(n-1);/*Y*/y=0*(n-1);X=x-1;Y=y;jmpflag=1;}
}
else if(flag==22){ /* v(10,3, */
    if(nh(x,y,nx,ny,0)) {x=2*(n-1);y=0*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=2*(n-1);y=0*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,2)) {x=8*(n-1);y=12*(n-1);X=x-1;Y=y-1;jmpflag=1;}
}
else if(flag==23){ /* v(4,2, */
    if(nh(x,y,nx,ny,5)) {x=8*(n-1);y=8*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=8*(n-1);y=8*(n-1);X=x-1;Y=y-1;jmpflag=1;}
}
else if(flag==24){ /* v(5,1, */
    if(nh(x,y,nx,ny,3)) {x=3*(n-1);y=1*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=3*(n-1);y=1*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,5)) {x=3*(n-1);y=1*(n-1);X=x-1;Y=y-1;jmpflag=1;}
}

```

```

else if(nh(x,y,nx,ny,0)) {x=8*(n-1);/*Y*/y=8*(n-1);X=x-1;Y=y;jmpflag=1;}
}
else if(flag==25){ /* v(10,2, */
    if(nh(x,y,nx,ny,5)) {x=3*(n-1);y=1*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=3*(n-1);y=1*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=3*(n-1);y=1*(n-1);X=x-1;Y=y;jmpflag=1;}
}
else if(flag==26){ /* v(9,2, */
    if(nh(x,y,nx,ny,5)) {x=5*(n-1);y=2*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=5*(n-1);y=2*(n-1);X=x-1;Y=y-1;jmpflag=1;}
}
else if(flag==27){ /* v(10,1, */
    if(nh(x,y,nx,ny,3)) {x=6*(n-1);y=7*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,4)) {x=6*(n-1);y=7*(n-1);X=x-1;Y=y;jmpflag=1;}
else if(nh(x,y,nx,ny,5)) {x=6*(n-1);y=7*(n-1);X=x-1;Y=y-1;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=5*(n-1);y=2*(n-1);X=x-1;Y=y;jmpflag=1;}
}
else if(flag==28){ /* v(5,2, */
    if(nh(x,y,nx,ny,5)) {x=6*(n-1);y=7*(n-1);X=x+1;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,0)) {x=6*(n-1);y=7*(n-1);X=x;Y=y+1;jmpflag=1;}
else if(nh(x,y,nx,ny,1)) {x=6*(n-1);y=7*(n-1);X=x-1;Y=y;jmpflag=1;}
}

else if(flag==101){ /* side(1,4, */
}
else if(flag==102){ /* side(8,0, */
}
else if(flag==103){ /* side(2,4, */
    if(nh(x,y,nx,ny,2)) {rot(2,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,3)) {rot(3,-2);jmpflag=101;}
if(jmpflag>100) {dlt=x-1*(n-1); x=3*(n-1);y=8*(n-1)-dlt;}
}
else if(flag==104){ /* side(8,5, */
    if(nh(x,y,nx,ny,3)) {rot(3,2);jmpflag=101;}
else if(nh(x,y,nx,ny,4)) {rot(4,2);jmpflag=101;}
if(jmpflag>100) {dlt=y-7*(n-1); x=2*(n-1)-dlt;y=5*(n-1)-dlt;}
}
else if(flag==105){ /* side(2,5, */
    if(nh(x,y,nx,ny,3)) {rot(3,-3);jmpflag=101;}
else if(nh(x,y,nx,ny,4)) {rot(4,-3);jmpflag=101;}
if(jmpflag>100) {dlt=y-3*(n-1); x=4*(n-1);y=10*(n-1)-dlt;}
}
else if(flag==106){ /* side(7,5, */
    if(nh(x,y,nx,ny,3)) {rot(3,3);jmpflag=101;}
else if(nh(x,y,nx,ny,4)) {rot(4,3);jmpflag=101;}
}

```

```

if(jmpflag>100) {dlt=y-9*(n-1); x=1*(n-1);y=4*(n-1)-dlt;}
}
else if(flag==107){ /* side(3,5, */
    if(nh(x,y,nx,ny,3)) {rot(3,2);jmpflag=101;}
else if(nh(x,y,nx,ny,4)) {rot(4,2);jmpflag=101;}
if(jmpflag>100) {dlt=y-1*(n-1); x=5*(n-1)-dlt;y=11*(n-1)-dlt;}
}
else if(flag==108){ /* side(7,4, */
    if(nh(x,y,nx,ny,2)) {rot(2,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,3)) {rot(3,-2);jmpflag=101;}
if(jmpflag>100) {dlt=x-4*(n-1); x=0*(n-1);y=2*(n-1)-dlt;}
}
else if(flag==109){ /* side(3,0, */
    if(nh(x,y,nx,ny,4)) {rot(4,1);jmpflag=101;}
else if(nh(x,y,nx,ny,5)) {rot(5,1);jmpflag=101;}
if(jmpflag>100) {dlt=x-0*(n-1); x=6*(n-1)+dlt;y=11*(n-1)+dlt;}
}
else if(flag==110){ /* side(6,4, */
    if(nh(x,y,nx,ny,2)) {rot(2,-1);jmpflag=101;}
else if(nh(x,y,nx,ny,3)) {rot(3,-1);jmpflag=101;}
if(jmpflag>100) {dlt=x-6*(n-1); x=0*(n-1)+dlt;y=1*(n-1);}
}
else if(flag==111){ /* side(4,0, */
    if(nh(x,y,nx,ny,4)) {rot(4,6);jmpflag=101;}
else if(nh(x,y,nx,ny,5)) {rot(5,6);jmpflag=101;}
if(jmpflag>100) {dlt=x-1*(n-1); x=7*(n-1)+dlt;y=12*(n-1);}
}
else if(flag==112){ /* side(6,3, */
    if(nh(x,y,nx,ny,1)) {rot(1,-6);jmpflag=101;}
else if(nh(x,y,nx,ny,2)) {rot(2,-6);jmpflag=101;}
if(jmpflag>100) {dlt=x-7*(n-1); x=1*(n-1)+dlt;y=0*(n-1);}
}
else if(flag==113){ /* side(4,1, */
    if(nh(x,y,nx,ny,5)) {rot(5,4);jmpflag=101;}
else if(nh(x,y,nx,ny,0)) {rot(0,4);jmpflag=101;}
if(jmpflag>100) {dlt=x-2*(n-1);x=8*(n-1);y=9*(n-1)-dlt;}
}
else if(flag==114){ /* side(10,2, */
    if(nh(x,y,nx,ny,0)) {rot(0,-4);jmpflag=101;}
else if(nh(x,y,nx,ny,1)) {rot(1,-4);jmpflag=101;}
if(jmpflag>100) {dlt=y-8*(n-1);x=3*(n-1)-dlt;y=1*(n-1)-dlt;}
}
else if(flag==115){ /* side(5,1, */
    if(nh(x,y,nx,ny,5)) {rot(5,3);jmpflag=101;}
else if(nh(x,y,nx,ny,0)) {rot(0,3);jmpflag=101;}
if(jmpflag>100) {dlt=x-4*(n-1);x=8*(n-1)-dlt;y=8*(n-1)-dlt;}
}

```

```

}
else if(flag==116){ /* side(10,1, */
    if(nh(x,y,nx,ny,5)) {rot(5,-3);jmpflag=101;}
else if(nh(x,y,nx,ny,0)) {rot(0,-3);jmpflag=101;}
if(jmpflag>100) {dlt=x-7*(n-1);x=5*(n-1)-dlt;y=2*(n-1)-dlt;}
}
else if(flag==117){ /* 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-2*(n-1);x=6*(n-1)-dlt;y=7*(n-1)-dlt;}
}
else if(flag==118){ /* side(9,1, */
    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=x-5*(n-1);x=5*(n-1);y=3*(n-1)-dlt;}
}

#if 1
else if(flag==119){ /* side(12,0, */
    if(nh(x,y,nx,ny,1)==1) {rot(1,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,2)==1) {rot(2,2);jmpflag=101;}
if(jmpflag>100) {if(sn==0) dsn=6;else if(sn==6) dsn=-6;}
}
else if(flag==120){ /* side(12,1, */
    if(nh(x,y,nx,ny,2)==1) {rot(2,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,3)==1) {rot(3,2);jmpflag=101;}
if(jmpflag>100) {if(sn==0) dsn=6;else if(sn==6) dsn=-6;}
}
else if(flag==121){ /* side(12,3, */
    if(nh(x,y,nx,ny,4)==1) {rot(4,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,5)==1) {rot(5,2);jmpflag=101;}
if(jmpflag>100) {if(sn==0) dsn=6;else if(sn==6) dsn=-6;}
}
else if(flag==122){ /* side(12,4, */
    if(nh(x,y,nx,ny,5)==1) {rot(5,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,0)==1) {rot(0,2);jmpflag=101;}
if(jmpflag>100) {if(sn==0) dsn=6;else if(sn==6) dsn=-6;}
}
else if(flag==123){ /* side(12,5, */
    if(nh(x,y,nx,ny,0)==1) {rot(0,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,1)==1) {rot(1,2);jmpflag=101;}
if(jmpflag>100) {if(sn==0) dsn=6;else if(sn==6) dsn=-6;}
}
#endif

#if 1

```

```

else if(flag==124){ /* side(11,0, */
    if(nh(x,y,nx,ny,1)==1) {rot(1,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,2)==1) {rot(2,2);jmpflag=101;}
if(jmpflag>100) {if(sn==6) dsn=6;else if(sn==12) dsn=-6;}
}
else if(flag==125){ /* side(11,1, */
    if(nh(x,y,nx,ny,2)==1) {rot(2,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,3)==1) {rot(3,2);jmpflag=101;}
if(jmpflag>100) {if(sn==6) dsn=6;else if(sn==12) dsn=-6;}
}
else if(flag==126){ /* side(11,3, */
    if(nh(x,y,nx,ny,4)==1) {rot(4,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,5)==1) {rot(5,2);jmpflag=101;}
if(jmpflag>100) {if(sn==6) dsn=6;else if(sn==12) dsn=-6;}
}
else if(flag==127){ /* side(11,4, */
    if(nh(x,y,nx,ny,5)==1) {rot(5,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,0)==1) {rot(0,2);jmpflag=101;}
if(jmpflag>100) {if(sn==6) dsn=6;else if(sn==12) dsn=-6;}
}
else if(flag==128){ /* side(11,5, */
    if(nh(x,y,nx,ny,0)==1) {rot(0,-2);jmpflag=101;}
else if(nh(x,y,nx,ny,1)==1) {rot(1,2);jmpflag=101;}
if(jmpflag>100) {if(sn==6) dsn=6;else if(sn==12) dsn=-6;}
}
#endif

else if(flag==1001){ /* cut(1,2, */
    if(nh(x,y,nx,ny,1)==1 && in(nx,ny)==0) {rot(1,1);jmpflag=1001;}
else if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,1);jmpflag=1001;}
if(jmpflag>1000) {dlt=x-4*(n-1);x=x;y+=dlt;}
}
else if(flag==1002){ /* cut(1,3, */
    if(nh(x,y,nx,ny,5)==1 && in(nx,ny)==0) {rot(5,-1);jmpflag=1001;}
else if(nh(x,y,nx,ny,0)==1 && in(nx,ny)==0) {rot(0,-1);jmpflag=1001;}
if(jmpflag>1000) {x=x;y=5*(n-1);}
}
else if(flag==1003){ /* cut(2,3, */
    if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,1);jmpflag=1001;}
else if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,1);jmpflag=1001;}
if(jmpflag>1000) {x=2*(n-1);y=y;}
}
else if(flag==1004){ /* cut(2,4, */
    if(nh(x,y,nx,ny,0)==1 && in(nx,ny)==0) {rot(0,-1);jmpflag=1001;}
else if(nh(x,y,nx,ny,1)==1 && in(nx,ny)==0) {rot(1,-1);jmpflag=1001;}

```

```

if(jmpflag>1000) {dlt=y-4*(n-1);x+=dlt;y=y;}
}
else if(flag==1005){ /* cut(3,4, */
    if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,1);jmpflag=1001;}
else if(nh(x,y,nx,ny,4)==1 && in(nx,ny)==0) {rot(4,1);jmpflag=1001;}
if(jmpflag>1000) {dlt=y-2*(n-1);x-=dlt;y=2*(n-1);}
}
else if(flag==1006){ /* cut(3,5, */
    if(nh(x,y,nx,ny,1)==1 && in(nx,ny)==0) {rot(1,-1);jmpflag=1001;}
else if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,-1);jmpflag=1001;}
if(jmpflag>1000) {dlt=1*(n-1)-x;x=1*(n-1);y+=dlt;}
}
else if(flag==1007){ /* cut(4,5, */
    if(nh(x,y,nx,ny,4)==1 && in(nx,ny)==0) {rot(4,1);jmpflag=1001;}
else if(nh(x,y,nx,ny,5)==1 && in(nx,ny)==0) {rot(5,1);jmpflag=1001;}
if(jmpflag>1000) {dlt=2*(n-1)-x;x=x;y-=dlt;}
}
else if(flag==1008){ /* cut(4,0, */
    if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,-1);jmpflag=1001;}
else if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,-1);jmpflag=1001;}
if(jmpflag>1000) {x=x;y=1*(n-1);}
}
else if(flag==1009){ /* cut(5,0, */
    if(nh(x,y,nx,ny,5)==1 && in(nx,ny)==0) {rot(5,1);jmpflag=1001;}
else if(nh(x,y,nx,ny,0)==1 && in(nx,ny)==0) {rot(0,1);jmpflag=1001;}
if(jmpflag>1000) {x=4*(n-1);y=y;}
}
else if(flag==1010){ /* cut(5,1, */
    if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,-1);jmpflag=1001;}
else if(nh(x,y,nx,ny,4)==1 && in(nx,ny)==0) {rot(4,-1);jmpflag=1001;}
if(jmpflag>1000) {dlt=2*(n-1)-y;x-=dlt;y=y;}
}
else if(flag==1011){ /* cut(11,2, */
    if(nh(x,y,nx,ny,1)==1 && in(nx,ny)==0) {rot(1,1);jmpflag=1001;}
else if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,1);jmpflag=1001;}
if(jmpflag>1000) {dlt=x-3*(n-1);x=x;y+=dlt;}
}
else if(flag==1012){ /* cut(11,3, */
    if(nh(x,y,nx,ny,5)==1 && in(nx,ny)==0) {rot(5,-1);jmpflag=1001;}
else if(nh(x,y,nx,ny,0)==1 && in(nx,ny)==0) {rot(0,-1);jmpflag=1001;}
if(jmpflag>1000) {x=x;y=3*(n-1);}
}

else if(flag==1013){ /* cut(6,2, */
    if(nh(x,y,nx,ny,1)==1 && in(nx,ny)==0) {rot(1,1);jmpflag=1001;}
else if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,1);jmpflag=1001;}

```



```

if(jmpflag>1000) {dlt=x-7*(n-1);x=x;y+=dlt;}
}
else if(flag==1014){ /* cut(6,3, */
    if(nh(x,y,nx,ny,5)==1 && in(nx,ny)==0) {rot(5,-1);jmpflag=1001;}
else if(nh(x,y,nx,ny,0)==1 && in(nx,ny)==0) {rot(0,-1);jmpflag=1001;}
if(jmpflag>1000) {x=x;y=11*(n-1);}
}
else if(flag==1015){ /* cut(7,3, */
    if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,1);jmpflag=1001;}
else if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,1);jmpflag=1001;}
if(jmpflag>1000) {x=5*(n-1);y=y;}
}
else if(flag==1016){ /* cut(7,4, */
    if(nh(x,y,nx,ny,0)==1 && in(nx,ny)==0) {rot(0,-1);jmpflag=1001;}
else if(nh(x,y,nx,ny,1)==1 && in(nx,ny)==0) {rot(1,-1);jmpflag=1001;}
if(jmpflag>1000) {dlt=y-10*(n-1);x+=dlt;y=y;}
}
else if(flag==1017){ /* cut(8,4, */
    if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,1);jmpflag=1001;}
else if(nh(x,y,nx,ny,4)==1 && in(nx,ny)==0) {rot(4,1);jmpflag=1001;}
if(jmpflag>1000) {dlt=y-8*(n-1);x-=dlt;y=8*(n-1);}
}
else if(flag==1018){ /* cut(8,5, */
    if(nh(x,y,nx,ny,1)==1 && in(nx,ny)==0) {rot(1,-1);jmpflag=1001;}
else if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,-1);jmpflag=1001;}
if(jmpflag>1000) {dlt=4*(n-1)-x;x=4*(n-1);y+=dlt;}
}
else if(flag==1019){ /* cut(9,5, */
    if(nh(x,y,nx,ny,4)==1 && in(nx,ny)==0) {rot(4,1);jmpflag=1001;}
else if(nh(x,y,nx,ny,5)==1 && in(nx,ny)==0) {rot(5,1);jmpflag=1001;}
if(jmpflag>1000) {dlt=5*(n-1)-x;x=x;y-=dlt;}
}
else if(flag==1020){ /* cut(9,0, */
    if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,-1);jmpflag=1001;}
else if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,-1);jmpflag=1001;}
if(jmpflag>1000) {x=x;y=7*(n-1);}
}
else if(flag==1021){ /* cut(10,0, */
    if(nh(x,y,nx,ny,5)==1 && in(nx,ny)==0) {rot(5,1);jmpflag=1001;}
else if(nh(x,y,nx,ny,0)==1 && in(nx,ny)==0) {rot(0,1);jmpflag=1001;}
if(jmpflag>1000) {x=7*(n-1);y=y;}
}
else if(flag==1022){ /* cut(10,1, */
    if(nh(x,y,nx,ny,3)==1 && in(nx,ny)==0) {rot(3,-1);jmpflag=1001;}
else if(nh(x,y,nx,ny,4)==1 && in(nx,ny)==0) {rot(4,-1);jmpflag=1001;}
if(jmpflag>1000) {dlt=8*(n-1)-y;x-=dlt;y=y;}
}

```

```

}
else if(flag==1023){ /* cut(12,2, */
    if(nh(x,y,nx,ny,1)==1 && in(nx,ny)==0) {rot(1,1);jmpflag=1001;}
else if(nh(x,y,nx,ny,2)==1 && in(nx,ny)==0) {rot(2,1);jmpflag=1001;}
if(jmpflag>1000) {dlt=x-6*(n-1);x=x;y+=dlt;}
}
else if(flag==1024){ /* cut(12,3, */
    if(nh(x,y,nx,ny,5)==1 && in(nx,ny)==0) {rot(5,-1);jmpflag=1001;}
else if(nh(x,y,nx,ny,0)==1 && in(nx,ny)==0) {rot(0,-1);jmpflag=1001;}
if(jmpflag>1000) {x=x;y=9*(n-1);}
}

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

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 i,int jmax)
{
int val;

if(i==jmax+1) val=0;
else if(i==-1) val=jmax;
else val=i;

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

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

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

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

x-=sn*(RESO-1);

if(0) ;

#if Odd!=1
else if(v(12,1,x,y)==1 && sn<12) val=0;
else if(v(12,0,x,y)==1 && sn<12) val=1;
else if(v(12,5,x,y)==1 && sn<12) val=2;

```

```

else if(v(12,4,x,y)==1 && sn<12) val=3;
else if(v(12,3,x,y)==1 && sn<12) val=4;
else if(v(12,2,x,y)==1 && sn<12) val=5;
#endif

/* for odd */
#if Odd==3
else if(v(11,1,x,y)==1 && sn>0) val=0;
else if(v(11,0,x,y)==1 && sn>0) val=1;
else if(v(11,5,x,y)==1 && sn>0) val=2;
else if(v(11,4,x,y)==1 && sn>0) val=3;
else if(v(11,3,x,y)==1 && sn>0) val=4;
else if(v(11,2,x,y)==1 && sn>0) val=5;
#endif

else val=-1;

x+=sn*(RES0-1);

return val;
}/** check_v **/

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 Nx,Ny,x[8],y[8],x_[8],y_[8];
int cp,ssize,posflag,pos,count;
int ca,c1,c2,c3,c4,c5,c7,cflag;
int nxp,nxm,nyp,nym;
int jmp[6+6],sntmp1,sntmp2;

ssize=sizeof(ss);
cp=CPMAX;
n=RES0;

acolor[0]=9;acolor[1]=10;acolor[2]=11;acolor[3]=12;acolor[4]=13;
acolor[5]=1;acolor[6]=2;acolor[7]=3;acolor[8]=4;acolor[9]=5;

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

```

```

/*999*/
ca=15;

if(Odd>0){
dx=0;dy=0;
nax[0]=(3+1+dx)*(n-1)      ;nay[0]=(3+1+dy)*(n-1);
nax[1]=(3+0+dx)*(n-1)      ;nay[1]=(3+1+dy)*(n-1);
nax[2]=(3-1+dx)*(n-1)      ;nay[2]=(3+0+dy)*(n-1);
nax[3]=(3-1+dx)*(n-1)      ;nay[3]=(3-1+dy)*(n-1);
nax[4]=(3+0+dx)*(n-1)      ;nay[4]=(3-1+dy)*(n-1);

if(Odd==1) {dx=0+3;dy=6;}
else {dx=0+3+12;dy=6;}
/* for odd */
nax[5]=(3+1+dx)*(n-1)      ;nay[5]=(3+1+dy)*(n-1);
nax[6]=(3+0+dx)*(n-1)      ;nay[6]=(3+1+dy)*(n-1);
nax[7]=(3-1+dx)*(n-1)      ;nay[7]=(3+0+dy)*(n-1);
nax[8]=(3-1+dx)*(n-1)      ;nay[8]=(3-1+dy)*(n-1);
nax[9]=(3+0+dx)*(n-1)      ;nay[9]=(3-1+dy)*(n-1);
}
else{
dx=0;dy=0;
nax[0]=(3+1+dx)*(n-1)      ;nay[0]=(3+1+dy)*(n-1);
nax[1]=(3+0+dx)*(n-1)      ;nay[1]=(3+1+dy)*(n-1);
nax[2]=(3-1+dx)*(n-1)      ;nay[2]=(3+0+dy)*(n-1);
nax[3]=(3-1+dx)*(n-1)      ;nay[3]=(3-1+dy)*(n-1);
nax[4]=(3+0+dx)*(n-1)      ;nay[4]=(3-1+dy)*(n-1);

dx=0+6;dy=0;
/* for even */
nax[5]=(3+1+dx)*(n-1)      ;nay[5]=(3+1+dy)*(n-1)-(n-1);
nax[6]=(3+0+dx)*(n-1)      ;nay[6]=(3+1+dy)*(n-1);
nax[7]=(3-1+dx)*(n-1)      ;nay[7]=(3+0+dy)*(n-1);
nax[8]=(3-1+dx)*(n-1)      ;nay[8]=(3-1+dy)*(n-1);
nax[9]=(3+0+dx)*(n-1)      ;nay[9]=(3-1+dy)*(n-1);
}

if(Odd>0){
/* for odd */
if(Odd==1) {sn1=0;sn2=0;}
else {sn1=0;sn2=12;}
}
else{
/* for even */
sn1=0;sn2=6;
}
}

```

```

}

i=0;
while(1){
if(flag_[i]){ /* CP_? */
if(i<=4) 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<=4) sn=sn1;else sn=sn2;
ig=i;

if(combination==0){
/* CW */
}/**if(combination)**/
else{
/* CCW */
if(drn==4) { /* 217 */
if(Odd>0){
if(i%5==0) {nax[i]--;}
else if(i%5==1) {nax[i]--;nay[i]--;}
else if(i%5==2) {nay[i]--;}
else if(i%5==3) {nax[i]++;}
else if(i%5==4) {nax[i]++;nay[i]++;}
}
}
else{
if(i%10==0) {nax[i]--;}
else if(i%10==1) {nax[i]--;nay[i]--;}
else if(i%10==2) {nay[i]--;}
else if(i%10==3) {nax[i]++;}
else if(i%10==4) {nax[i]++;nay[i]++;}

else if(i%10==5) {nax[i]--;nay[i]--;}
else if(i%10==6) {nax[i]++;}
else if(i%10==7) {nax[i]++;nay[i]++;}
else if(i%10==8) {nay[i]++;}

```

```

else if(i%10==9) {nax[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)**/

/*putpixel(3*(n-1),3*(n-1),0);
putpixel((3+3+12)*(n-1),(3+6)*(n-1),0);*/
if(GRPH==1 && cnt==0) use_subroop();
cnt++;
/***** while(cp) -> *****/

while(cp){
kbhit_();
/*if(cnt>=1064) use_subroop();*/
/*if(cnt==18 && rcount[0]>86) refill=0;*/
if(refill==0) break;
if(0) use_subroop();

algo=random_(2);

i=0;
while(1){

if(flag_[i]){ /* CP_? */
if(Odd!=1) {if(i<=4) sn=sn1;else sn=sn2;}
else sn=sn1;
ig=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_;

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

/*if(cnt==620 && rcount[0]==49 && i==2){
printf(" %d\n",sn);
printf(" %d %d %d %d %d %d\n",c1,c5,c2,c3,c7,c4);
putpixel(nx[i],ny[i],8);
use_subroop();
}*/

if(sn==0) dsn=6;
else if(sn==6 && Ny>6*(n-1)) dsn=-6;
else if(sn==6 && Ny<6*(n-1)) dsn=6;
else if(sn==12) dsn=-6;

posflag=check_v(Nx,Ny);

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

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{
j=5;rot(1,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
enX[j]=X;enY[j]=Y;
j=6;rot(1,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;

```



```

    enX[j]=X;enY[j]=Y;
j=7;rot(1,4);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;

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{
j=5;rot(0,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;
j=6;rot(0,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;
j=7;rot(0,4);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;

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

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{
j=5;rot(4,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;

/* cut(11,3,x,y), flag=1012 */
rot(0,-1);
x0=X-1;y0=Y;
y0--=n-1;

```

```

X=x0+tmp0;Y=y0+tmp1;X_=x0;Y_=y0;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;

j=6;rot(4,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;
j=7;rot(4,4);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;

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

/* cut(11,2,x,y), flag=1011 */
rot(1,1);
x0=X-1;y0=Y-1;
y0+=n-1;
X=x0+tmp0;Y=y0+tmp1;X_=x0;Y_=y0;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;

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

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 */
if(sn==0) dsn=6;
else if(sn==6 && Ny>6*(n-1)) dsn=-6;
else if(sn==6 && Ny<6*(n-1)) dsn=6;
else if(sn==12) dsn=-6;

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 */
j=5;rot(2,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;
j=6;rot(2,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;
j=7;rot(2,4);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;

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 */
j=5;rot(1,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;
j=6;rot(1,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;
j=7;rot(1,4);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;

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 */
j=5;rot(0,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;
j=6;rot(0,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;
j=7;rot(0,4);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;

jmax=7;
}
else if(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 */
j=5;rot(5,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;
j=6;rot(5,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;
j=7;rot(5,4);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;

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 */
j=5;rot(4,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;

/* cut(11,3,x,y), flag=1012 */
rot(0,-1);
x0=X-1;y0=Y;
y0-=n-1;
X=x0+tmp0;Y=y0+tmp1;X_=x0;Y_=y0;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;

j=6;rot(4,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;
j=7;rot(4,4);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;

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 */
j=5;rot(3,2);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;
j=6;rot(3,3);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;
j=7;rot(3,4);X=Nx+tmp0;Y=Ny+tmp1;X_=Nx;Y_=Ny;X+=dsn*(n-1);X_+=dsn*(n-1);sn_=sn+dsn;
    enX[j]=X;enY[j]=Y;enX_[j]=X_;enY_[j]=Y_;enSN[j]=sn_;

/* cut(11,2,x,y), flag=1011 */
rot(1,1);
x0=X-1;y0=Y-1;
y0+=n-1;
X=x0+tmp0;Y=y0+tmp1;X_=x0;Y_=y0;

```

```

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

jmax=7;
}

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;;){
if(pixel[fen("X",j,jmax)][fen("Y",j,jmax)]!=ca &&
pixel[fen("X",j-1,jmax)][fen("Y",j-1,jmax)]==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<=4) 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;;){
if(pixel[fen("X",j,jmax)][fen("Y",j,jmax)]!=ca &&
pixel[fen("X",j+1,jmax)][fen("Y",j+1,jmax)]==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<=4) 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==18 && rcount[0]>85){
printf(" i:%d posflag:%d pos:%d\n",i,posflag,pos);

```

```

for(j=0;j<=jmax;j++)
printf(" X:%d Y:%d X_:%d Y_:%d SN:%d pixel[X][Y]:%d\n",
      enX[j],enY[j],enX_[j],enY_[j],enSN[j],pixel[enX[j]][enY[j]]);
printf(" \n");
}*/

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]);
}

/*if(cnt==1024 && i==2) printf(" rcnt:%d i:%d sn:%d\n",rcount[0],i,sn);*/
/*if(rcount[0]==51 && i==1){
printf(" %d %d %d %d %d %d\n",c1,c5,c2,c3,c7,c4);
putpixel(nx[i],ny[i],8);
use_subroop();
}*/

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<=4) 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)**/  
  
/*printf(" %d\n",pixel[5*(n-1)][6*(n-1)-3]);*/  
  
return 0;  
}/** cag_r **/
```