Contents

[Giải thuật Bresenham vẽ đoạn thẳng k < -1 1](#_Toc45537730)

[Thuật toán Bresenham cho đường thẳng với k>1 4](#_Toc45537731)

[Giải thuật Bresenham với hệ số góc k<1 6](#_Toc45537732)

[Giải thuật Bresenham (-1 < k < 0) 9](#_Toc45537733)

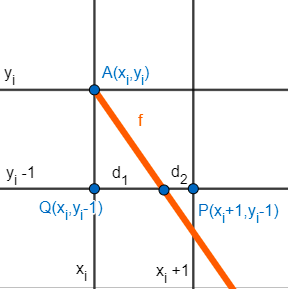
[Giải thuật Bresemham sinh đường Ellipse 12](#_Toc45537734)

[GIẢI THUẬT SINH ĐƯỜNG ELLIPSE 19](#_Toc45537735)

# Giải thuật Bresenham vẽ đoạn thẳng k < -1

Trình bày, sơ đồ khối, code C/C++

Xét đoạn thẳng với k < -1



Gọi (x,yi+1) là điểm thuộc đường thẳng, ta có x = k(yi - 1) + b

d1 = xi+1 – xi = k(yi - 1) + b – xi

d2 = xi + 1 – xi+1 = xi + 1 – k(yi - 1) - b

* Nếu d1 <= d2 => xi+1 = xi
* Ngược lại d1 > d2 => xi+1 = xi + 1

Đặt D = d1 – d2 = 2k(yi - 1) - 2xi + 2b – 1

Có k =

Đặt Pi = =

* Pi =
* Pi  = (yi – 1) – xi + b -

Ta tính bước tiếp:

Pi+1 = (yi+1 – 1) – xi+1 + b -

Pi+1 – Pi = (yi+1 – yi) – (xi+1 - xi)

Có yi+1 = yi – 1 nên:

* P­i+1 – Pi = – (xi+1 - xi)

Vì y < 0 => Pi ngược dấu với D

Nếu Pi <= 0 thì xi+1 = xi + 1

* Pi+1 = Pi - –

Nếu Pi > 0 thì xi+1 = xi

* Pi+1 = Pi -

Tính giá trị đầu P1 ?

P1 = (d1 – d2)

* P1 =
* P1  = (y1 – 1) – x1 + b -

Có x1 = ky1 + b = y1 + b

* P1 = (y1 – 1) – (y1 + b) + b -
* P1 = y1 - - y1 - b + b -
* P1  = - -

|  |  |
| --- | --- |
| y >= y2  No  No  No  Yes  Yes  Yes  p = -2/dx – 1/dy;  const1 = -2/dx;  const2 = -2/dx – 2/dy;  p = p + const2;  p = p + const1;  x = x + 1;  y = y – 1;  p <= 0  putpixel(x,y,c);  vc = 1e7;  x = x1; y = y1;  dx = x2 – x1;  dy = y2 – y1  x1 = x2  p = -vc;  const1 = -vc;  const2 = -vc; | /\*Thuat toan Bresenham ve dthang (k<-1) \*/ void Bre-line(int x1, int y1, int x2, int y2, int c) { int x, y, dx, dy,p,const1,const2, vc;  vc = 1e7;  x = x1;  dx = x2 - x1;  dy = y2 - y1;  if (x1 == x2) {  p = -vc;  const1 = -vc;  const2 = -vc;  } else {  p = -2/dx – 1/dy;  const1 = -2/dx;  const2 = -2/dx – 2/dy;  }  for (y = y1; y >= y2; y--) {  putpixel(x, y, c);  if (p <= 0)  p += const2; // p=p – 2/dx – 2/dy  else {  p +=const1; //p=p – 2/dx  x++;  }  } } |

# Thuật toán Bresenham cho đường thẳng với k>1

Mô tả:

y

d1

yi+1=yi+1 d2

yi

O xi xi+1 xi+1 x

Ta có: d1=xi+1-xi

d2=xi+1-xi+1

mà yi+1=k\*xi+1 + b => xi+1 =yi+1 - b) = (yi + 1 - b)

* d1 = (yi + 1 - b) - xi

d2=xi+1 - (yi + 1- b)

* D = d1 - d2 = (yi + 1 - b) - 2xi – 1 = (yi + 1 - b) - 2xi – 1
* D = 2yi+ 2 - 2b – 2xi -
* Pi = 2yi+ 2 - 2b – 2xi -

Pi+1 = 2yi+1+ 2 - 2b – 2xi+1 -

* Pi+1-Pi = 2(yi+1 - yi)- 2(xi+1 - xi) = 2(yi +1 - yi)-2(xi+1-xi)

= 2 - 2(xi+1-xi)

* Trường hợp 1: d1<=d2

xi+1 = xi => Pi+1 – Pi = 2

* Trường hợp 2: d1>d2

xi+1 = xi+1 => Pi+1 – Pi = 2 - 2

* Giá trị đầu:

P1 = 2y1+ 2 - 2b – 2x1 -

Mà y1 = x1 + b

* P1 = 2x1 +2b + 2 - 2b – 2x1 - = 2-

Sơ đồ thuật toán Bresenham cho đường thẳng với k>1

|  |  |
| --- | --- |
| Sơ đồ thuật toán  x = x1 ; y=y1; dx = x2 - x1; dy = y2 - y1; p = 2dx – dy;  putpixel(x,y);  p<=0  no    y = y + 1  p = p + 2dx -2dy  yes  p = p + 2dx  x = x + 1  x<x2  yes  no | void Bre\_line(int x1, int y1,   int x2, int y2, int c){  int x, y, dx, dy, p;  x = x1;  dx = x2 – x1;  dy = y2 – y1;  p = 2\*dx - dy;  for (y=y1 ; y<=y2 ; y++){  putpixel( x, y, c);  if(p<=0){  p+=2\*(dx – dy);  x++;  }  else p+=2\*dx;  }  } |

# Giải thuật Bresenham với hệ số góc k<1

yi

yi+1

d1

d2

xi xxi+1

Mô tả giải thuật Bresenham với k<1

Giả sử đã vẽ được điểm (xi ,yi) và điểm (i+1) có 2 lựa chọn đó là: (xi,yi+1 ) hoặc (xi +1,yi+1) do hệ số góc k < 1

Gọi (xi+1 ,yi+1) là điểm thuộc đoạn thẳng, ta có yi+1 = yi -1 = k(xi+1) +b (1)

d1 = xi+1 - xi

d2 = xi + 1 – xi+1

D = d1 – d2 = 2\*xi+1 – 2\*xi  - 1

* Nếu d1 <= d2 🡺 xi+1 = xi
* Ngược lại d1 > d2 🡺 xi+1 = xi +1

Ta có k = y/x và đặt

Pi = y\*D = y\*(d1 – d2)

= y\*(2\*xi+1 - 2\*xi - 1) (2)

Mà yi+1= k\*xi+1 + b = yi – 1

🡺 xi+1 = (yi – 1 – b)/k = x \*(yi – 1 – b)/ y (3)

từ (2) và (3) ta được:

Pi = y\*(2\*x \*(yi – 1 – b)/ y - 2\*xi - 1)

= 2\*x \*(yi – 1 – b) - y\*(2\*xi + 1)

Pi+1 = 2\*x \*(yi+1 – 1 – b) - y\*(2\*xi+1 + 1)

= 2\*x \*(yi – 2 – b) – 2\*y\*(xi + 1)

Pi+1 – Pi = - 2\*x– 2\*y\*(xi+1 - xi)

Nếu xi+1 = xi 🡺 Pi+1 – Pi = - 2\*x 🡺 Pi+1 = Pi - 2\*x

Nếu xi+1= xi+1🡺 Pi+1 – Pi = - 2\*x - 2\*y

🡺 Pi+1 = Pi - 2\*x - 2\*y

* Tính P1:
* Pi = 2\*x \*(yi – 1 – b) - y\*(2\*xi + 1)
* Với y1 = kx1  + b và k = y/x
* P1 = 2\*x \*( y\*x1/x  + b – 1 – b) - y\*(2\*xi + 1)
* P1 = 2\*y\*x1 -2\*x - 2\*y\*x1 - y = -2\*x - y

**Sơ đồ khối**

x = x1 ; y = y1;

dx = x2-x1;

dy = y2 - y1;

Y <y2

y = y + 1

P = P - 2\*dx

x = x+1

P = P - 2dx-2dy

P<=0

Putpixel(x,y)

**Code minh họa :**

void bre\_line(int x1, int y1, int x2, int y2, int c)

{

int x, y, dx, dy, p;

x = x1;

dx = x2 – x1;

dy = y2 – y1;

p = -2dx – dy;

for(y = y1; y <= y2; y++)

{ putpixel(x, y, c);

if(p <= 0)

{ p = p - 2dx;

else{

p = p - 2dx – 2dy;

x++;

} }}}

# Giải thuật Bresenham (-1 < k < 0)

Trình bày giải thuật.

**M**

**B**

Xét đường thẳng với   
-1 < k < 1.

Chọn tỉ lệ xích là 1 nên  
Gọi là điểm thuộc đoạn thẳng, ta có:

**A**

d2

d1

Ta có:

* TH1: chọn A =>
* TH2: chọn B =>

Đặt

Tính giá trị đầu

**Sơ đồ khối**

S

Đ

Đ

x+=1

P += 2dy + dx

y--

P += 2dy

P <= 0

Putpixel(x,y)

S

**Giải thuật**

***void Bre\_line(int x1, int y1, int x2, int y2, int c ){  
 int x = x1, y = y1;***

***int dx = x2 – x1, dy = y2 – y1;***

***int p = 2\*dy + 2\*dx;***

***while(x < x2){  
 putpixel(x, y, c);***

***if(p <= 0){  
 p += 2\*dy + 2\*dx;  
 y-=1;***

***}***

***else {***

***p+=2dy;***

***}***

***x+=1;***

***}***

***}***

# Giải thuật Bresemham sinh đường Ellipse

Ta có phương trình tổng quát Ellipse:

Suy ra, phương trình Ellipse dạng tường minh:

Do ellipse đối xứng 4 , nên ta chỉ cần vẽ Ellipse trên góc phần tư thứ 1 và lấy đối xứng

qua Ox, Ox, O.

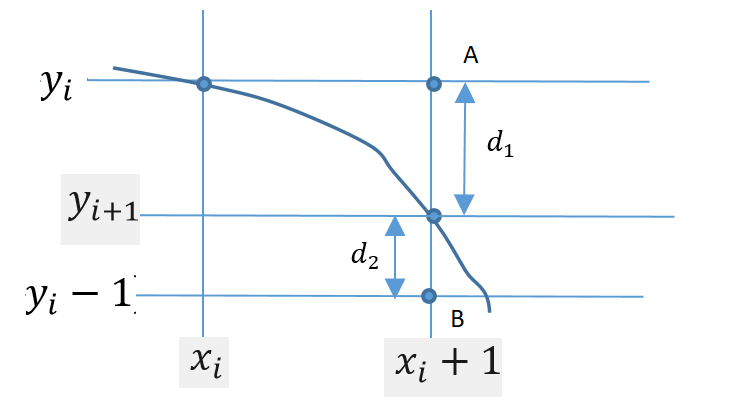
Có đạo hàm riêng theo x:

Đạo hàm riêng theo y:

Tại góc phần tư thứ nhất, chia ellipse thành 2 đoạn:

* Đoạn 1 đồng biến theo x.( hay )
* Đoạn 2 đồng biến theo y.( )

Xét trên đoạn 1:



Bắt đầu tại (0,b) với . (x luôn tăng)

Đặt :

TT:

**Tính**

**\***

Suy ra:

( thay vào \* )

**Xét**

**\*\***

Xác định dấu của ta sẽ xác định được và

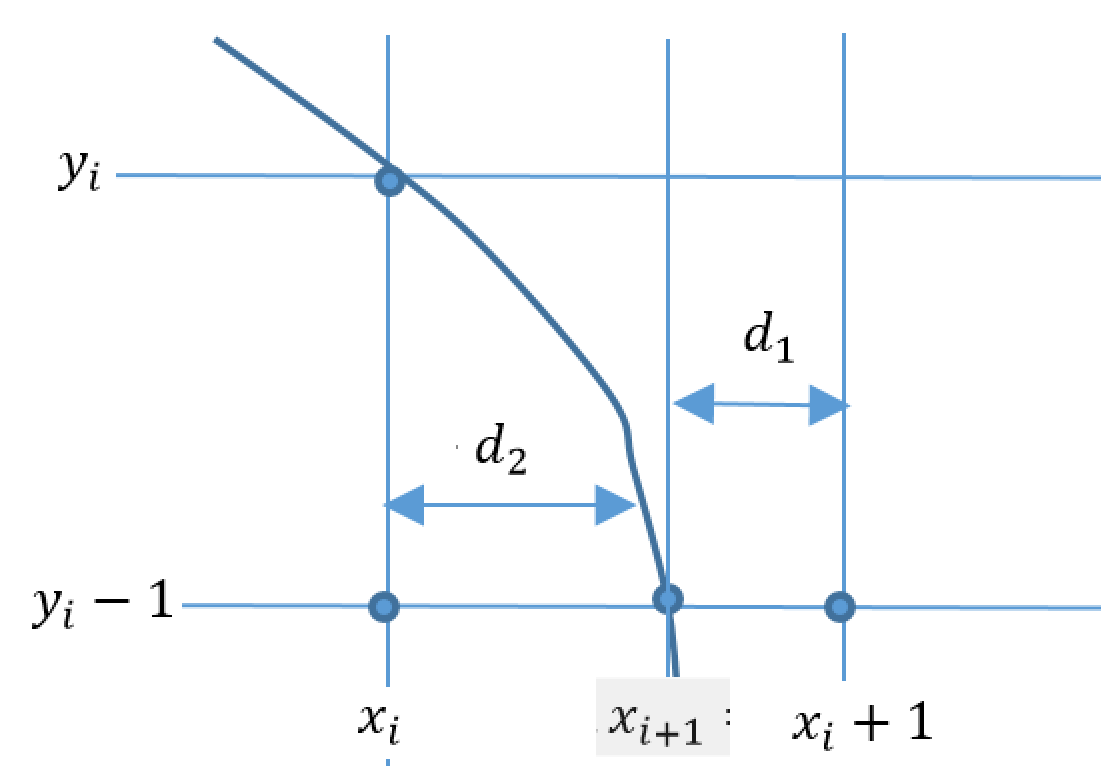
* , thay vào \*\*

* , thay vào \*\*

**Tính giá trị đoạn 1 tại có**

**Trên đoạn 2 ,** đổi vai trò của x, y , hàm đồng biến theo y:

(y luôn giảm)



Nếu :

Nếu :

Ta lấy luôn giá trị của và của đoạn 1để khởi tạo giá trị

**Sơ đồ thuật toán:**

Sai

Sai

Đúng

Đúng

y++;

x++;

P <0 ?

Putpixel(x,y,c)

**Vẽ đoạn 1**

**Từ trên xuống**

x= 0;

y=b;

p=

**Vẽ đoạn 2**

Sai

Sai

Đúng

Đúng

y--;

x ++;

?

P <0 ?

Putpixel(x,y,c)

Vẽ đoạn phần tư thứ 1 gồm 2 đoạn ta dung putpixel( x, y, c)

Trong trường hợp vẽ cả ellipse , ta thay putpixel(x,y,c) bằng hàm vẽ 4 điểm

**Đoạn chương trình:**

Chú thích: xc, yc là tọa độ của tâm ellipse, c là màu.

void ve4diem(int xc,int yc,int x, int y,int c){

putpixel(xc+x,yc+y,c);

putpixel(xc-x,yc+y,c);

putpixel(xc-x,yc-y,c);

putpixel(xc+x,yc-y,c);

}

**Có hiện tượng overstrike tại các điểm (0,b) (a,0) (0, -b), (-a, 0)**

void Elipse(int xc,int yc,int a,int b,int color){

float p,a2,b2;

int x,y;

a2=pow(a,2);

b2=pow(b,2);

x=0;

y=b;

putpixel(x\_center,y\_center+y,color);

putpixel(x\_center,y\_center-y,color); // Vẽ 2 điểm đặc biệt trên Oy

p=2\*(b2/a2)-(2\*b)+1;

**//ve nhanh thu 1**

while((b2/a2)\*x<=y){

x++;

if(p<0){

p=p+2\*(b2/a2)\*(2\*x+3);

}

else{

p= p+4- 4\*y + 2\*(b2/a2)\*(2\*x+3);

y--;

}

Ve4diem(xc,yc,x,y,color);

}

**//ve nhanh thu 2**

p=2\*x\*x -2\*x +1 +2\*(a2/b2)\*(y+1)\*(y+1)-2\*a2;

while(y>0){

Ve4diem(xc,yc,x,y,color);

if(p<0){

p=p +2\*(a2/b2)\*(2\*y+3);

}

else{

p=p+ 2\*(a2/b2)\*(2\*y+3)+4\*(1-x);

x=x+1;

}

y=y-1;

}

// Vẽ 2 điểm đặc biệt trên Ox

putpixel(x\_center+x,y\_center,color);

putpixel(x\_center-x,y\_center,color);

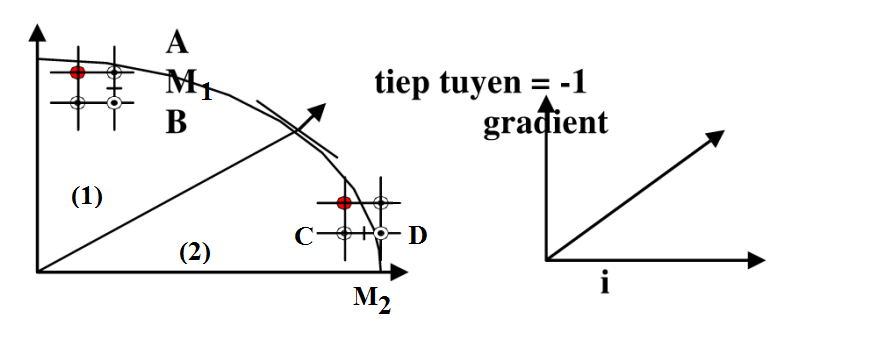
}

# GIẢI THUẬT SINH ĐƯỜNG ELLIPSE

1. Xây dựng giải thuật

Phương trình ellipse có dạng:

→ f(x, y) = b2x2 + a2y2 – a2b2 = 0



Véc tơ với tiếp tuyến gradient = 1

Ta có tiếp tuyến với cung tròn (độ dốc) = -1 = dy / dx = - fx / fy

Trong đó fx = 2b2x là đạo hàm riêng phần của f(x, y) theo x

Và fy = 2a2y là đạo hàm riêng phần của f(x, y) theo y

Giả sử ta chỉ xét trên góc phần tư thứ nhất của ellipse: Giả sử chia cung từ (0, b) đến (a, 0) tại Q có độ dốc -1

Phần (1): x thay đổi thì y thay đổi theo

Phần (2): y thay đổi thì x thay đổi theo

**Xét trên phần 1**

Bắt đầu từ (0, b), bước thứ i (xi, yi) chọn tiếp

A(xi + 1, yi)

B(xi + 1, yi – 1)

Giá trị của hàm f(x, y) tại điểm M1:

Pi = f(xi + 1, yi – 1 / 2) = b2(xi + 1)2 + a2(yi – 1 / 2)2 - a2b2

→ Pi + 1 = f(xi + 1 + 1, yi + 1 – 1 / 2) = b2(xi + 1 + 1)2 + a2(yi + 1 – 1 / 2)2 - a2b2

Pi + 1 – Pi = b2((xi + 1 + 1)2 - (xi + 1)2) + a2((yi + 1 – 1 / 2)2 - (yi – 1 / 2)2)

Pi + 1 = Pi + 2b2xi + 1 + b2 + a2((yi + 1 – 1 / 2)2 - (yi – 1 / 2)2)

* Nếu Pi < 0 chọn A (do điểm M1 nằm dưới đường ellipse)

xi + 1 = xi + 1

yi + 1 = yi

Pi + 1 = Pi + b2 (2xi + 3)

* Nếu Pi 0 chọn B (do M1 nằm phía trên ellipse)

xi + 1 = xi + 1

yi + 1 = yi - 1

Pi + 1 = Pi + b2 (2xi + 3) + a2 ((yi – 1 – 1 / 2)2 - (yi – 1 / 2)2)

= Pi + b2 (2xi + 3) + a2 (-3yi + + yi - )

= Pi + b2 (2xi + 3) + a2 (-2yi + 2)

Tính P1 = ? tại (0, b):

P1 = f(x1 + 1, y1 – 1 / 2) = b2 + a2(b – 1 / 2)2 – a2b2

P1 = b2 – a2b +

**Xét trên phần 2**

Ta lấy tọa độ của pixel sau cùng của phần 1 làm tọa độ của pixel đầu tiên trong phần 2.

Giả sử đang vẽ đến pixel (xj, yj), pixel kế tiếp có thể là:

C(xj, yj – 1)

D(xj + 1, yj - 1)

Giá trị của hàm f(x, y) tại điểm M2:

Qj = f(xj + 1 / 2, yj – 1) = b2(xj + 1 / 2)2 + a2(yj - 1)2 – a2b2

→ Qj + 1 = f(xj + 1 – 1 / 2, yj + 1 – 1) = b2(xj + 1 + 1 / 2)2 + a2(yj + 1 - 1)2 – a2b2

Qj + 1 – Qj = b2((xj + 1 + 1 / 2)2 - (xj + 1 / 2)2) + a2((yj + 1 - 1)2 - (yj - 1)2)

Qj + 1 = Qj + b2((xj + 1 + 1 / 2)2 - (xj + 1 / 2)2) – 2a2yj + 1 + a2

* Nếu Qj < 0, chọn D (do M2 nằm dưới ellipse)

yj + 1 = yj – 1

xj + 1 = xj + 1

Qj + 1 = Qj + b2((xj + 3 / 2)2 - (xj + 1 / 2)2) – 2a2(yj – 1) + a2

Qj + 1 = Qj + b2(3xj + – xj - ) – 2a2(yj – 1) + a2

Qj + 1 = Qj + b2(2xj + 2) – 2a2(yj – 1) + a2

Hay Qj + 1 = Qj + b2(2xj + 2) + a2( -2yj + 3)

* Nếu Qj 0, chọn C (do M2 nằm phía trên ellipse)

yj + 1 = yj - 1

xj + 1 = xj

Qj + 1 = Qj – 2a2(yj – 1) + a2

Hay Qj + 1 = Qj + a2(3 - 2yj)

Tính Q1?

Giả sử pixel cuối cùng được vẽ trong phần 1 là (xk, yk).

→ Q1 = f(xk + 1 / 2, yk - 1) = b2(xk + 1 / 2)2 + a2(yk - 1)2 – a2b2

Lưu ý: Lúc lấy đối xứng 4 cách để vẽ 1 ellipse hoàn chỉnh từ góc phần tư thứ nhất các điểm sau bị vẽ 2 lần: (0, b); (0, -b); (a, 0); (-a, 0)

1. Vẽ lưu đồ thuật toán

*Lưu đồ chỉ biểu diễn cho việc vẽ góc phần tư thứ nhất.*

x = 0, y = b  
fx = 2b2x, fy = 2a2y  
P = b2 – a2b +

Putpixel(x, y, c)

P = P + b2(2x + 3) + a2(-2y + 2)

S

P < 0

Đ

y = y – 1  
fy = fy – 2a2

P = P + b2(2x + 3)

Đ

y = y - 1

S

Đ

Q = Q + a2(3 – 2y)

Q < 0

Putpixel(x, y, c)

S

Đ

Q = b2(x + )2 + a2(y - 1)2 – a2b2

fx < fy

x = x + 1  
fx = fx + 2b2

Q = Q + b2(2x + 2) + a2(3 – 2y)

x = x + 1

S

y > 0

1. Cài đặt thuật toán trên c/c++

#include <graphics.h>

#define ROUND(a) ((long) (a + 0.5))

void plot(int xc, int yc, int x, int y, int color){

putpixel(xc + x, yc + y, color);

putpixel(xc - x, yc + y, color);

putpixel(xc + x, yc - y, color);

putpixel(xc - x, yc - y, color);

}

void mid\_ellipse(int xc, int yc, int a, int b, int color){

long x, y, fx, fy, a2, b2, p, q;

x = 0, y = b;

a2 = a \* a;

b2 = b \* b;

fx = 0, fy = 2 \* a2 \* y;

p = ROUND(b2 - a2 \* b + 0.25 \* a2);

putpixel(xc + x, yc + y, color);

putpixel(xc + x, yc - y, color);

while(1){

if(p < 0){

p += b2 \* (2 \* x + 3);

}else{

p += b2 \* (2 \* x + 3) + a2 \* (-2 \* y + 2);

y--;

fy -= 2 \* a2;

}

x++;

fx += 2 \* b2;

if(fx < fy)

plot(xc, yc, x, y, color);

else break;

}

q = ROUND(b2 \* (x + 0.5) \* (x + 0.5) + a2 \* (y - 1) \* (y - 1) - a2 \* b2);

while(y > 0){

plot(xc, yc, x, y, color);

if(q < 0){

q += b2 \* (2 \* x + 2) + a2 \* (3 - 2 \* y);

x++;

}else{

q += a2 \* (3 - 2 \* y);

}

y--;

}

putpixel(xc + x, yc + y, color);

putpixel(xc - x, yc + y, color);

}

int main(int argc, char \*argv[])

{

initwindow(1000, 1000);

mid\_ellipse(getmaxx()/2, getmaxy()/2, 150, 80, 7);

getch();

}