**Câu hỏi 1. Sử dụng thuật toán DDA vẽ đoạn thẳng đi qua 2 điểm A(2;3) và B(12;8)**

**1.1.** Trình bày các bước để thực hiện giải thuật trên

**1.2.** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

**A**

**B**

* 1. Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Và xA<xB, yA<yB

Nên: và

Ta có:

Suy ra:

và

Vậy

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | x*i* | yi | ROUND |
| 0  1  2  3  4  5  6  7  8  9  10 | 2  3  4  5  6  7  8  9  10  11  12 | 3  = 3 + 0.5 = 3.5  = 3.5 + 0.5 = 4  = 4 + 0.5 = 4.5  = 4.5 + 0.5 = 5  = 5 + 0.5 = 5.5  = 5.5 + 0.5 = 6  = 6 + 0.5 = 6.5  = 6.5 + 0.5 = 7  = 7 + 0.5 = 7.5  = 7.5 + 0.5 = 8 | 3  4  4  5  5  6  6  7  7  8  8 |

* 1. Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void DDA()

{

int x;

float y;

initwindow(400,400);

x=xa;

y=ya;

putpixel(x,ROUND(y),125);

while(x<xb)

{

x=x+1;

y=y+m;

putpixel(x,ROUND(y),255);

delay(100);

printf("(%d;%d)\n",x,ROUND(y));

}

}

//chuong trinh chinh

main()

{

nhapxy();

DDA();

getch();

}

**Câu hỏi 2. Sử dụng thuật toán DDA vẽ đoạn thẳng đi qua 2 điểm A(12;8) và B(2;3)**

**1.1.** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

**1.1**. Từ phương trình đường thẳng đi qua 2 điểm A và B

**B**

**A**

Với

Và xA>xB, yA>yB

Nên: và

Ta có:

Suy ra:

và

Vậy

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | x*i* | yi | ROUND |
| 0  1  2  3  4  5  6  7  8  9  10 | 12  11  10  9  8  7  6  5  4  3  2 | 8  = 8 - 0.5 = 7.5  = 7.5 - 0.5 = 7  = 7 - 0.5 = 6.5  = 6.5 - 0.5 = 6  = 6 - 0.5 = 5.5  = 5.5 - 0.5 = 5  = 5 - 0.5 = 4.5  = 4.5 - 0.5 = 4  = 4 - 0.5 = 3.5  = 3.5 - 0.5 = 3 | 8  8  7  7  6  6  5  5  4  4  3 |

**1.2. Lập trình mô phỏng**

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void DDA()

{

int x;

float y;

initwindow(400,400);

x=xa;

y=ya;

putpixel(x,ROUND(y),125);

while(x>xb)

{

x=x-1;

y=y-m;

putpixel(x,ROUND(y),255);

delay(100);

printf("(%d;%d)\n",x,ROUND(y));

}

}

//chuong trinh chinh

main()

{

nhapxy();

DDA();

getch();

}

**Câu hỏi 3. Sử dụng thuật toán DDA vẽ đoạn thẳng đi qua 2 điểm A(12;3) và B(2;8)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

**1.1**. Từ phương trình đường thẳng đi qua 2 điểm A và B

**B**

**A**

Với

Và xA>xB, yA<yB

Nên: và

Ta có:

Suy ra:

và

Vậy

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | x*i* | yi | ROUND |
| 0  1  2  3  4  5  6  7  8  9  10 | 12  11  10  9  8  7  6  5  4  3  2 | 3  = 3 + 0.5 = 3.5  = 3.5 + 0.5 = 4  = 4 + 0.5 = 4.5  = 4.5 + 0.5 = 5  = 5 + 0.5 = 5.5  = 5.5 + 0.5 = 6  = 6 + 0.5 = 6.5  = 6.5 + 0.5 = 7  = 7 + 0.5 = 7.5  = 7.5 + 0.5 = 8 | 3  4  4  5  5  6  6  7  7  8  8 |

**1.2. Lập trình mô phỏng**

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void DDA()

{

int x;

float y;

initwindow(400,400);

x=xa;

y=ya;

putpixel(x,ROUND(y),125);

while(x>xb)

{

x=x-1;

y=y-m;

putpixel(x,ROUND(y),255);

delay(100);

printf("(%d;%d)\n",x,ROUND(y));

}

}

//chuong trinh chinh

main()

{

nhapxy();

DDA();

getch();

}

**Câu hỏi 4. Sử dụng thuật toán DDA vẽ đoạn thẳng đi qua 2 điểm A(2;8) và B(12;3)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

**1.1**. Từ phương trình đường thẳng đi qua 2 điểm A và B

**A**

**B**

Với

Và xA<xB, yA<yB

Nên: và

Ta có:

Suy ra:

và

Vậy

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | x*i* | yi | ROUND |
| 0  1  2  3  4  5  6  7  8  9  10 | 2  3  4  5  6  7  8  9  10  11  12 | 8  = 8 - 0.5 = 7.5  = 7.5 - 0.5 = 7  = 7 - 0.5 = 6.5  = 6.5 - 0.5 = 6  = 6 - 0.5 = 5.5  = 5.5 - 0.5 = 5  = 5 - 0.5 = 4.5  = 4.5 - 0.5 = 4  = 4 - 0.5 = 3.5  = 3.5 - 0.5 = 3 | 8  8  7  7  6  6  5  5  4  4  3 |

**1.2. Lập trình mô phỏng**

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void DDA()

{

int x;

float y;

initwindow(400,400);

x=xa;

y=ya;

putpixel(x,ROUND(y),125);

while(x<xb)

{

x=x+1;

y=y+m;

putpixel(x,ROUND(y),255);

delay(100);

printf("(%d;%d)\n",x,ROUND(y));

}

}

//chuong trinh chinh

main()

{

nhapxy();

DDA();

getch();

}

**Câu hỏi 5. Sử dụng thuật toán DDA vẽ đoạn thẳng đi qua 2 điểm A(3;2) và B(8;12)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

**1.1**. Từ phương trình đường thẳng đi qua 2 điểm A và B

**A**

**B**

Với

Và xA<xB, yA<yB

Nên: và

Ta có:

Suy ra:

và

Vậy

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | y*i* | xi | ROUND |
| 0  1  2  3  4  5  6  7  8  9  10 | 2  3  4  5  6  7  8  9  10  11  12 | 3  = 3 + 0.5 = 3.5  = 3.5 + 0.5 = 4  = 4 + 0.5 = 4.5  = 4.5 + 0.5 = 5  = 5 + 0.5 = 5.5  = 5.5 + 0.5 = 6  = 6 + 0.5 = 6.5  = 6.5 + 0.5 = 7  = 7 + 0.5 = 7.5  = 7.5 + 0.5 = 8 | 3  4  4  5  5  6  6  7  7  8  8 |

**1.2. Lập trình mô phỏng**

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void DDA()

{

float x;

int y;

initwindow(400,400);

x=xa;

y=ya;

putpixel(ROUND(x),y,125);

while(y<yb)

{

y=y+1;

x=x+1/m;

putpixel(ROUND(x),y,255);

delay(100);

printf("(%d;%d)\n",ROUND(x),y);

}

}

//chuong trinh chinh

main()

{

nhapxy();

DDA();

getch();

}

**Câu hỏi 6. Sử dụng thuật toán DDA vẽ đoạn thẳng đi qua 2 điểm A(8;12) và B(3;2)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

**1.1**. Từ phương trình đường thẳng đi qua 2 điểm A và B

**B**

**A**

Với

Và xA>xB, yA>yB

Nên: và

Ta có:

Suy ra:

và

Vậy

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | y*i* | xi | ROUND |
| 0  1  2  3  4  5  6  7  8  9  10 | 12  11  10  9  8  7  6  5  4  3  2 | 8  = 8 - 0.5 = 7.5  = 7.5 - 0.5 = 7  = 7 - 0.5 = 6.5  = 6.5 - 0.5 = 6  = 6 - 0.5 = 5.5  = 5.5 - 0.5 = 5  = 5 - 0.5 = 4.5  = 4.5 - 0.5 = 4  = 3 - 0.5 = 3.5  = 3.5 - 0.5 = 3 | 8  8  7  7  6  6  5  5  4  4  3 |

**1.2. Lập trình mô phỏng**

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void DDA()

{

float x;

int y;

initwindow(400,400);

x=xa;

y=ya;

putpixel(ROUND(x),y,125);

while(y>yb)

{

y=y-1;

x=x-1/m;

putpixel(ROUND(x),y,255);

delay(100);

printf("(%d;%d)\n",ROUND(x),y);

}

}

//chuong trinh chinh

main()

{

nhapxy();

DDA();

getch();

}

**Câu hỏi 7. Sử dụng thuật toán DDA vẽ đoạn thẳng đi qua 2 điểm A(8;2) và B(3;12)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

**1.1**. Từ phương trình đường thẳng đi qua 2 điểm A và B

**A**

**B**

Với

Và xA>xB, yA<yB

Nên: và

Ta có:

Suy ra:

và

Vậy

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | y*i* | xi | ROUND |
| 0  1  2  3  4  5  6  7  8  9  10 | 2  3  4  5  6  7  8  9  10  11  12 | 8  = 8 - 0.5 = 7.5  = 7.5 - 0.5 = 7  = 7 - 0.5 = 6.5  = 6.5 - 0.5 = 6  = 6 - 0.5 = 5.5  = 5.5 - 0.5 = 5  = 5 - 0.5 = 4.5  = 4.5 - 0.5 = 4  = 3 - 0.5 = 3.5  = 3.5 - 0.5 = 3 | 8  8  7  7  6  6  5  5  4  4  3 |

**1.2. Lập trình mô phỏng**

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void DDA()

{

float x;

int y;

initwindow(400,400);

x=xa;

y=ya;

putpixel(ROUND(x),y,125);

while(y>yb)

{

y=y-1;

x=x+1/m;

putpixel(ROUND(x),y,255);

delay(100);

printf("(%d;%d)\n",ROUND(x),y);

}

}

//chuong trinh chinh

main()

{

nhapxy();

DDA();

getch();

}

**Câu hỏi 8. Sử dụng thuật toán DDA vẽ đoạn thẳng đi qua 2 điểm A(3;12) và B(8;2)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

**1.1**. Từ phương trình đường thẳng đi qua 2 điểm A và B

**B**

**A**

Với

Và xA<xB, yA>yB

Nên: và

Ta có:

Suy ra:

và

Vậy

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | y*i* | xi | ROUND |
| 0  1  2  3  4  5  6  7  8  9  10 | 12  11  10  9  8  7  6  5  4  3  2 | 3  = 3 + 0.5 = 3.5  = 3.5 + 0.5 = 4  = 4 + 0.5 = 4.5  = 4.5 + 0.5 = 5  = 5 + 0.5 = 5.5  = 5.5 + 0.5 = 6  = 6 + 0.5 = 6.5  = 6.5 + 0.5 = 7  = 7 + 0.5 = 7.5  = 7.5 + 0.5 = 8 | 3  4  4  5  5  6  6  7  7  8  8 |

**1.2. Lập trình mô phỏng**

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void DDA()

{

float x;

int y;

initwindow(400,400);

x=xa;

y=ya;

putpixel(ROUND(x),y,125);

while(y>yb)

{

y=y-1;

x=x-1/m;

putpixel(ROUND(x),y,255);

delay(100);

printf("(%d;%d)\n",ROUND(x),y);

}

}

//chuong trinh chinh

main()

{

nhapxy();

DDA();

getch();

}

**Câu hỏi 1. Sử dụng thuật toán Bresenham vẽ đoạn thẳng đi qua 2 điểm A(2;3) và B(12;8)**

**1.1.** Trình bày các bước để thực hiện giải thuật trên

**1.2.** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

(xi,yi)

Q(x,y)

(xi+1,yi)

(xi+1,yi + 1)

d1

d2

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Đặt:

pt đường thẳng đi qua Q: dx(d1-d2) =

Xét Qi thuộc đường thẳng tại thời điểm i:

Suy ra:

Xét Qi+1là điểm kế tiếp sau Qi

Vì dx = xB – xA > 0

Nếu d1<=d2 hay Qi<0:

Ngược lại Qi>=0:

Tại thời điểm ban đầu:

Hay

* ***Q0***: dx - 2dy = (12-2) – 2(8-3) = 0
* ***Qi<0 (yi+1 = yi+1):*** Qi + 2dx – 2dy = Qi + 2(12-2) – 2(8-3) = 10
* ***Qi>=0 (yi+1 = yi):*** Qi - 2dy = Qi - 2(8-3) = -10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bước thứ i | Công thức Qi | Q*i* | xi | yi |
| 0  1  2  3  4  5  6  7  8  9  10 | dx-2dy  Qi - 2dy  Qi + 2dx – 2dy  Qi - 2dy  Qi + 2dx – 2dy  Qi - 2dy  Qi + 2dx – 2dy  Qi - 2dy  Qi + 2dx – 2dy  Qi - 2dy  Qi + 2dx – 2dy | 0  -10  10  -10  10  -10  10  -10  10  -10  10 | 2  3  4  5  6  7  8  9  10  11  12 | 3  4  4  5  5  6  6  7  7  8  8 |

**1.2.** Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Bresenham()

{

int x;

int y;

int q0;

int q;

int dy;

int dx;

initwindow(400,400);

putpixel(xa,ya,255);

dy=yb-ya;

dx=xb-xa;

q0=dx-2\*dy;

x=xa;

y=ya;

putpixel(x,y,125);

q=q0;

while(x<xb)

{

if(q<0)

{

q=q-2\*dy+2\*dx;

y++;

}

else

{

q=q-2\*dy;

}

x=x+1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

}

//chuong trinh chinh

main()

{

nhapxy();

Bresenham();

getch();

}

**Câu hỏi 2. Sử dụng thuật toán Bresenham vẽ đoạn thẳng đi qua 2 điểm A(12;8) và B(2;3)**

**1.1.** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

(xi,yi)

Q(x,y)

(xi - 1,yi -1)

(xi - 1,yi)

d1

d2

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Đặt:

pt đường thẳng đi qua Q: dx(d1-d2) =

Xét Qi thuộc đường thẳng tại thời điểm i:

Suy ra:

Xét Qi+1là điểm kế tiếp sau Qi

Vì dx = xB – xA < 0

Nếu d1<=d2 hay Qi>=0:

Ngược lại Qi<0:

Tại thời điểm ban đầu:

Hay

* ***Q0***: dx + 2dy = (2-12) + 2(3-8) = 0
* ***Qi>=0 (yi+1 = yi):*** 2dy = 2(3-8) = -10
* ***Qi<0 (yi+1 = yi - 1):*** - 2dx + 2dy = -2(2-12) + 2(3-8) = 10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bước thứ i | Công thức Qi | Q*i* | xi | yi |
| 0  1  2  3  4  5  6  7  8  9  10 | dx+2dy  Qi + 2dy  Qi - 2dx + 2dy  Qi + 2dy  Qi - 2dx + 2dy  Qi + 2dy  Qi - 2dx + 2dy  Qi + 2dy  Qi - 2dx + 2dy  Qi + 2dy  Qi - 2dx + 2dy | 0  -10  10  -10  10  10  -10  10  -10  10  -10 | 2  3  4  5  6  7  8  9  10  11  12 | 3  4  4  5  5  6  6  7  7  8  8 |

**1.2.** Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Bresenham()

{

int x;

int y;

int q0;

int q;

int dy;

int dx;

initwindow(400,400);

putpixel(xa,ya,255);

dy=yb-ya;

dx=xb-xa;

q0=dx-2\*dy;

x=xa;

y=ya;

putpixel(x,y,125);

q=q0;

while(x>xb)

{

if(q<0)

{

q=q-2\*dy+2\*dx;

y++;

}

else

{

q=q-2\*dy;

}

x=x-1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

}

//chuong trinh chinh

main()

{

nhapxy();

Bresenham();

getch();

}

**Câu hỏi 3. Sử dụng thuật toán Bresenham vẽ đoạn thẳng đi qua 2 điểm A(12;3) và B(2;8)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

(xi,yi)

Q(x,y)

(xi - 1,yi)

(xi - 1,yi+1)

d1

d2

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Đặt:

pt đường thẳng đi qua Q: dx(d1-d2) =

Xét Qi thuộc đường thẳng tại thời điểm i:

Suy ra:

Xét Qi+1là điểm kế tiếp sau Qi

Vì dx = xB – xA < 0

Nếu d1<=d2 hay Qi>=0:

Ngược lại Qi<0:

Tại thời điểm ban đầu:

Hay

* ***Q0***: dx + 2dy = (2-12) + 2(8-3) = 0
* ***Qi>=0 (yi+1 = yi):*** 2dx + 2dy = 2(8-3) + 2(2-12) = -10
* ***Qi<0 (yi+1 = yi + 1):*** 2dy = 2(8-3) = 10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bước thứ i | Công thức Qi | Q*i* | xi | yi |
| 0  1  2  3  4  5  6  7  8  9  10 | dx+2dy  Qi + 2dx + 2dy  Qi + 2dy  Qi + 2dx + 2dy  Qi + 2dy  Qi + 2dx + 2dy  Qi + 2dy  Qi + 2dx + 2dy  Qi + 2dy  Qi + 2dx + 2dy  Qi + 2dy | 0  -10  0  -10  0  -10  0  -10  0  -10  0 | 12  11  10  9  8  7  6  5  4  3  2 | 3  3  4  4  5  5  6  6  7  7  8 |

**1.2.** Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Bresenham()

{

int x;

int y;

int q0;

int q;

int dy;

int dx;

initwindow(400,400);

putpixel(xa,ya,255);

dy=yb-ya;

dx=xb-xa;

q0=dx+2\*dy;

x=xa;

y=ya;

putpixel(x,y,125);

q=q0;

while(x>xb)

{

if(q<=0)

{

q=q+2\*dy;

y++;

}

else

{

q=q+2dx+2\*dy;

}

x=x-1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

}

//chuong trinh chinh

main()

{

nhapxy();

Bresenham();

getch();

}

**Câu hỏi 4. Sử dụng thuật toán Bresenham vẽ đoạn thẳng đi qua 2 điểm A(2;8) và B(12;3)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

(xi,yi)

Q(x,y)

(xi + 1,yi)

(xi + 1,yi-1)

d1

d2

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Đặt:

pt đường thẳng đi qua Q: dx(d1-d2) =

Xét Qi thuộc đường thẳng tại thời điểm i:

Suy ra:

Xét Qi+1là điểm kế tiếp sau Qi

Vì dx = xB – xA > 0

Nếu d1<=d2 hay Qi<=0:

Ngược lại Qi>0:

Tại thời điểm ban đầu:

Hay

* ***Q0***: -dx -2dy = -(12-2) - 2(3-8) = 0
* ***Qi<=0 (yi+1 = yi):*** -2dx =- 2(3-8)= 10
* ***Qi>0 (yi+1 = yi -1):*** -2dx-2dy = -2(3-8)-2(12-2) = -10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bước thứ i | Công thức Qi | Q*i* | xi | yi |
| 0  1  2  3  4  5  6  7  8  9  10 | dx+2dy  Qi - 2dx  Qi - 2dx - 2dy  Qi - 2dx  Qi - 2dx - 2dy  Qi - 2dx  Qi - 2dx - 2dy  Qi - 2dx  Qi - 2dx - 2dy  Qi - 2dx  Qi - 2dx - 2dy | 0  10  0  10  0  10  0  10  0  10  0 | 2  3  4  5  6  7  8  9  10  11  12 | 8  8  7  7  6  6  5  5  4  4  3 |

**1.2.** Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Bresenham()

{

int x;

int y;

int q0;

int q;

int dy;

int dx;

initwindow(400,400);

putpixel(xa,ya,255);

dy=yb-ya;

dx=xb-xa;

q0=-dx-2\*dy;

x=xa;

y=ya;

putpixel(x,y,125);

q=q0;

while(x<xb)

{

if(q<=0)

{

q=q-2\*dx;

}

else

{

q=q-2dx-2\*dy;

y--;

}

x=x+1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

}

//chuong trinh chinh

main()

{

nhapxy();

Bresenham();

getch();

}

**Câu hỏi 5. Sử dụng thuật toán Bresenham vẽ đoạn thẳng đi qua 2 điểm A(3;2) và B(8;12)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

(xi,yi)

Q(x,y)

(xi + 1,yi+1)

(xi,yi + 1)

d1

d2

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Đặt:

pt đường thẳng đi qua Q: dy(d1-d2) =

Xét Qi thuộc đường thẳng tại thời điểm i:

Suy ra:

Xét Qi+1là điểm kế tiếp sau Qi

Vì dy = yB – yA > 0

Nếu d1<=d2 hay Qi<=0:

Ngược lại Qi>0:

Tại thời điểm ban đầu:

Hay

* ***Q0***: 2dx -dy = 2(8-3) - (12-2) = 0
* ***Qi<=0 (xi+1 = xi):*** 2dx =2(8-3)= 10
* ***Qi>0 (xi+1 = xi +1):*** -2dy+2dx = -2(12-2)+2(3-8) = -10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bước thứ i | Công thức Qi | Q*i* | yi | xi |
| 0  1  2  3  4  5  6  7  8  9  10 | dx+2dy  Qi + 2dx  Qi - 2dy + 2dx  Qi + 2dx  Qi - 2dy + 2dy  Qi + 2dx  Qi - 2dy + 2dy  Qi + 2dx  Qi - 2dy + 2dy  Qi + 2dx  Qi - 2dy - 2dx | 0  10  0  10  0  10  0  10  0  10  0 | 2  3  4  5  6  7  8  9  10  11  12 | 3  4  4  5  5  6  6  7  7  8  8 |

**1.2.** Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Bresenham()

{

int x;

int y;

int q0;

int q;

int dy;

int dx;

initwindow(400,400);

putpixel(xa,ya,255);

dy=yb-ya;

dx=xb-xa;

q0=2\*dx-dy;

x=xa;

y=ya;

putpixel(x,y,125);

q=q0;

while(y<xb)

{

if(q<=0)

{

q=q+2\*dx;

}

else

{

q=q-2dx+2\*dy;

x++;

}

y=y+1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

}

//chuong trinh chinh

main()

{

nhapxy();

Bresenham();

getch();

}

**Câu hỏi 6. Sử dụng thuật toán Bresenham vẽ đoạn thẳng đi qua 2 điểm A(8;12) và B(3;2)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

(xi,yi)

Q(x,y)

(xi ,yi-1)

(xi - 1,yi - 1)

d1

d2

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Đặt:

pt đường thẳng đi qua Q: dy(d1-d2) =

Xét Qi thuộc đường thẳng tại thời điểm i:

Suy ra:

Xét Qi+1là điểm kế tiếp sau Qi

Vì dy = yB – yA < 0

Nếu d1<=d2 hay Qi>=0:

Ngược lại Qi<0:

Tại thời điểm ban đầu:

Hay

* ***Q0***: -2dx +dy = -2(3-8)+ (2-12) = 0
* ***Qi>=0 (xi+1 = xi-1):*** 2dy-2dx =2(2-12)-2(3-8)= -10
* ***Qi<0 (xi+1 = xi ):*** -2dx = -2(3-8) = 10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bước thứ i | Công thức Qi | Q*i* | yi | xi |
| 0  1  2  3  4  5  6  7  8  9  10 | dy-2dx  Qi + 2dy - 2dx  Qi - 2dx  Qi + 2dy - 2dx  Qi - 2dx  Qi + 2dy - 2dx  Qi - 2dx  Qi + 2dy - 2dx  Qi - 2dx  Qi + 2dy - 2dx  Qi - 2dx | 0  -10  10  -10  10  -10  10  -10  10  -10  10 | 12  11  10  9  8  7  6  5  4  3  2 | 8  7  7  6  6  5  5  4  4  3  3 |

**1.2.** Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Bresenham()

{

int x;

int y;

int q0;

int q;

int dy;

int dx;

initwindow(400,400);

putpixel(xa,ya,255);

dy=yb-ya;

dx=xb-xa;

q0=dy-2\*dx;

x=xa;

y=ya;

putpixel(x,y,125);

q=q0;

while(y>yb)

{

if(q>=0)

{

q=q+2\*dy-2\*dx;

x--;

}

else

{

q=q-2dx;

}

y=y-1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

}

//chuong trinh chinh

main()

{

nhapxy();

Bresenham();

getch();

}

**Câu hỏi 7. Sử dụng thuật toán DDA vẽ đoạn thẳng đi qua 2 điểm A(8;2) và B(3;12)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

(xi,yi)

Q(x,y)

(xi -1,yi +1)

(xi - 1,yi)

d1

d2

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Đặt:

pt đường thẳng đi qua Q: dy(d1- d2) =

Xét Qi thuộc đường thẳng tại thời điểm i:

Suy ra:

Xét Qi+1là điểm kế tiếp sau Qi

Vì dy = yB – yA > 0

Nếu d1<=d2 hay Qi<=0:

Ngược lại Qi>0:

Tại thời điểm ban đầu:

Hay

* ***Q0***: 2dx +dy = 2(3-8)+ (12-2) = 0
* ***Qi<=0 (xi+1 = xi-1):*** 2dy+2dx =2(12-2)+2(3-8)= 10
* ***Qi>0 (xi+1 = xi ):*** 2dx = 2(3-8) = -10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bước thứ i | Công thức Qi | Q*i* | yi | xi |
| 0  1  2  3  4  5  6  7  8  9  10 | 2dx+dy  Qi + 2dy +2dx  Qi + 2dx  Qi + 2dy + 2dx  Qi + 2dx  Qi + 2dy + 2dx  Qi + 2dx  Qi + 2dy + 2dx  Qi + 2dx  Qi + 2dy + 2dx  Qi + 2dx | 0  10  -10  10  -10  10  -10  10  -10  10  -10 | 2  3  4  5  6  7  8  9  10  11  12 | 8  7  7  6  6  5  5  4  4  3  3 |

**1.2.** Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Bresenham()

{

int x;

int y;

int q0;

int q;

int dy;

int dx;

initwindow(400,400);

putpixel(xa,ya,255);

dy=yb-ya;

dx=xb-xa;

q0=dy-2\*dx;

x=xa;

y=ya;

putpixel(x,y,125);

q=q0;

while(y<yb)

{

if(q<=0)

{

q=q+2\*dy+2\*dx;

x--;

}

else

{

q=q+2dx;

}

y=y+1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

}

//chuong trinh chinh

main()

{

nhapxy();

Bresenham();

getch();

}

**Câu hỏi 8. Sử dụng thuật toán DDA vẽ đoạn thẳng đi qua 2 điểm A(3;12) và B(8;2)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

(xi,yi)

Q(x,y)

(xi + 1,yi -1)

(xi,yi -1)

d1

d2

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Đặt:

pt đường thẳng đi qua Q: dy(d1- d2) =

Xét Qi thuộc đường thẳng tại thời điểm i:

Suy ra:

Xét Qi+1là điểm kế tiếp sau Qi

Vì dy = yB – yA < 0

Nếu d1<=d2 hay Qi>=0:

Ngược lại Qi<0:

Tại thời điểm ban đầu:

Hay

* ***Q0***: -2dx - dy = -2(8-3) - (2-12) = 0
* ***Qi>=0 (xi+1 = xi):*** -2dx =-2(8-3)= -10
* ***Qi<0 (xi+1 = xi +1):*** -2dy-2dx = -2(2-12)-2(8-3) = 10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bước thứ i | Công thức Qi | Q*i* | yi | xi |
| 0  1  2  3  4  5  6  7  8  9  10 | -2dx-dy  Qi -2dx  Qi -2dy-2dx  Qi -2dx  Qi -2dy-2dx  Qi -2dx  Qi -2dy-2dx  Qi -2dx  Qi -2dy-2dx  Qi -2dx  Qi -2dy-2dx | 0  -10  10  -10  10  -10  10  -10  10  -10  10 | 12  11  10  9  8  7  6  5  4  3  2 | 3  3  4  4  5  5  6  6  7  7  8 |

**1.2.** Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Bresenham()

{

int x;

int y;

int q0;

int q;

int dy;

int dx;

initwindow(400,400);

putpixel(xa,ya,255);

dy=yb-ya;

dx=xb-xa;

q0=-2\*dx-dy;

x=xa;

y=ya;

putpixel(x,y,125);

q=q0;

while(y>yb)

{

if(q>=0)

{

q=q-2\*dx;

}

else

{

q=q-2\*dy-2\*dx;

x++;

}

y=y-1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

}

//chuong trinh chinh

main()

{

nhapxy();

Bresenham();

getch();

}

**Câu hỏi 1. Sử dụng thuật toán Midpoint vẽ đoạn thẳng đi qua 2 điểm A(2;3) và B(12;8)**

**1.1.** Trình bày các bước để thực hiện giải thuật trên

**1.2.** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

xi,yi

xi+1,yi

xi+1,yi+1

Q(x,y)

P(xi+1,yi+1/2)

* 1. Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i:

Với A = dy = (yB-yA) = (8-3) = 5

B = -dx = -(xB-xA) = -(12-2) = -10

Suy ra:

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i+1 là thời điểm liền kề i:

Suy ra:

Đặt

Hay

Vậy:

Xác định P0

Với

Nên

A = 5

B = -10

P0= 0

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | x*i* | Pi | yi |
| 0  1  2  3  4  5  6  7  8  9  10 | 2  3  4  5  6  7  8  9  10  11  12 | 0  -5  5  -5  5  -5  5  -5  5  -5  5 | 3  4  4  5  5  6  6  7  7  8  8 |

Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Midpoint()

{

int x;

int y;

float p0;

float p;

int A;

int B;

initwindow(400,400);

putpixel(xa,ya,255);

x=xa;

y=ya;

putpixel(x,y,125);

//xac dinh cac he so

A=yb-ya;

B=-(xb-xa);

p0= (float) (A+B/2);

p=p0;

while(x<xb)

{

if(p<=0)

{

p=(float)(p+A);

}

else

{

p=(float)(p+A+B);

y++;

}

x=x+1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

//closegraph();

}

**Câu hỏi 2. Sử dụng thuật toán Midpoint vẽ đoạn thẳng đi qua 2 điểm A(12;8) và B(2;3)**

**1.1.** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

xi,yi

xi-1,yi-1

xi-1,yi

Q(x,y)

P(xi-1,yi-1/2)

* 1. Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i:

Với A = dy = (yB-yA) = (3-8) = -5

B = -dx = -(xB-xA) = -(2-12) = 10

Suy ra:

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i+1 là thời điểm liền kề i:

Suy ra:

Đặt

Hay

Vậy:

Xác định P0

Với

Nên

A = -5

B = 10

P0= 0

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | *xi* | Pi | yi |
| 0  1  2  3  4  5  6  7  8  9  10 | 12  11  10  9  8  7  6  5  4  3  2 | 0  -5  5  -5  5  -5  5  -5  5  -5  5 | 8  7  7  6  6  5  5  4  4  3  3 |

Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Midpoint()

{

int x;

int y;

float p0;

float p;

int A;

int B;

initwindow(400,400);

putpixel(xa,ya,255);

x=xa;

y=ya;

putpixel(x,y,125);

//xac dinh cac he so

A=yb-ya;

B=-(xb-xa);

p0= (float) (-A-B/2);

p=p0;

while(x>xb)

{

if(p<0)

{

p=(float)(p-A);

}

else

{

p=(float)(p-A-B);

y--;

}

x=x-1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

//closegraph();

}

**Câu hỏi 3. Sử dụng thuật toán Midpoint vẽ đoạn thẳng đi qua 2 điểm A(12;3) và B(2;8)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

xi,yi

xi-1,yi

xi-1,yi+1

Q(x,y)

P(xi-1,yi+1/2)

* 1. Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i:

Với A = dy = (yB-yA) = (8-3) = 5

B = -dx = -(xB-xA) = -(2-12) = 10

Suy ra:

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i+1 là thời điểm liền kề i:

Suy ra:

Đặt

Hay

Vậy:

Xác định P0

Với

Nên

A = 5

B = 10

P0= 0

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | *xi* | Pi | yi |
| 0  1  2  3  4  5  6  7  8  9  10 | 12  11  10  9  8  7  6  5  4  3  2 | 0  -5  0  -5  0  -5  0  -5  0  -5  0 | 3  3  4  4  5  5  6  6  7  7  8 |

Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Midpoint()

{

int x;

int y;

float p0;

float p;

int A;

int B;

initwindow(400,400);

putpixel(xa,ya,255);

x=xa;

y=ya;

putpixel(x,y,125);

//xac dinh cac he so

A=yb-ya;

B=-(xb-xa);

p0= (float) (-A-B/2);

p=p0;

while(x<xb)

{

if(p>=0)

{

p=(float)(p-A);

}

else

{

p=(float)(p-A+B);

y++;

}

x=x+1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

//closegraph();

}

**Câu hỏi 4. Sử dụng thuật toán Midpoint vẽ đoạn thẳng đi qua 2 điểm A(2;8) và B(12;3)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

xi,yi

xi+1,yi-1

xi+1,yi

Q(x,y)

P(xi+1,yi-1/2)

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i:

Với A = dy = (yB-yA) = (3-8) = -5

B = -dx = -(xB-xA) = (2-12) = -10

Suy ra:

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i+1 là thời điểm liền kề i:

Suy ra:

Đặt

Hay

Vậy:

Xác định P0

Với

Nên

A = -5

B = -10

P0= 0

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | *xi* | Pi | yi |
| 0  1  2  3  4  5  6  7  8  9  10 | 2  3  4  5  6  7  8  9  10  11  12 | 0  -5  0  -5  0  -5  0  -5  0  -5  0 | 8  8  7  7  6  6  5  5  4  4  3 |

Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Midpoint()

{

int x;

int y;

float p0;

float p;

int A;

int B;

initwindow(400,400);

putpixel(xa,ya,255);

x=xa;

y=ya;

putpixel(x,y,125);

//xac dinh cac he so

A=yb-ya;

B=-(xb-xa);

p0= (float) (-A-B/2);

p=p0;

while(x<xb)

{

if(p>0)

{

p=(float)(p+A);

}

else

{

p=(float)(p+A-B);

y--;

}

x=x+1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

//closegraph();

}

**Câu hỏi 5. Sử dụng thuật toán Midpoint vẽ đoạn thẳng đi qua 2 điểm A(3;2) và B(8;12)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

xi,yi

xi,yi+1

xi+1,yi+1

Q(x,y)

P(xi+1/2,yi+1)

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i:

Với A = dy = (yB-yA) = (12-2) = 10

B = -dx = -(xB-xA) = -(8-3) = -5

Suy ra:

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i+1 là thời điểm liền kề i:

Suy ra:

Đặt

Hay

Vậy:

Xác định P0

Với

Nên

A = 10

B = -5

P0= 0

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | *yi* | Pi | xi |
| 0  1  2  3  4  5  6  7  8  9  10 | 2  3  4  5  6  7  8  9  10  11  12 | 0  -5  0  -5  0  -5  0  -5  0  -5  0 | 3  3  4  4  5  5  6  6  7  7  8 |

Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Midpoint()

{

int x;

int y;

float p0;

float p;

int A;

int B;

initwindow(400,400);

putpixel(xa,ya,255);

x=xa;

y=ya;

putpixel(x,y,125);

//xac dinh cac he so

A=yb-ya;

B=-(xb-xa);

p0= (float) (A/2+B);

p=p0;

while(y<yb)

{

if(p>0)

{

p=(float)(p+B);

}

else

{

p=(float)(p+A+B);

x++;

}

y=y+1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

//closegraph();

}

**Câu hỏi 6. Sử dụng thuật toán Midpoint vẽ đoạn thẳng đi qua 2 điểm A(8;12) và B(3;2)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

xi,yi

xi,yi -1

xi-1,yi-1

Q(x,y)

P(xi-1/2,yi-1)

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i:

Với A = dy = (yB-yA) = (2-12) = -10

B = -dx = -(xB-xA) = -(3-8) = 5

Suy ra:

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i+1 là thời điểm liền kề i:

Suy ra:

Đặt

Hay

Vậy:

Xác định P0

Với

Nên

A = -10

B = 5

P0= 0

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | *yi* | Pi | xi |
| 0  1  2  3  4  5  6  7  8  9  10 | 12  11  10  9  8  7  6  5  4  3  2 | 0  -5  0  -5  0  -5  0  -5  0  -5  0 | 8  8  7  7  6  6  5  5  4  4  3 |

Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Midpoint()

{

int x;

int y;

float p0;

float p;

int A;

int B;

initwindow(400,400);

putpixel(xa,ya,255);

x=xa;

y=ya;

putpixel(x,y,125);

//xac dinh cac he so

A=yb-ya;

B=-(xb-xa);

p0= (float) (-A/2-B);

p=p0;

while(y>yb)

{

if(p>0)

{

p=(float)(p-B);

}

else

{

p=(float)(p-A-B);

x--;

}

y=y-1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

//closegraph();

}

**Câu hỏi 7. Sử dụng thuật toán Midpoint vẽ đoạn thẳng đi qua 2 điểm A(8;2) và B(3;12)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

xi,yi

xi,yi +1

xi-1,yi+1

Q(x,y)

P(xi-1/2,yi+1)

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i:

Với A = dy = (yB-yA) = (12-2) = 10

B = -dx = -(xB-xA) = -(3-8) = 5

Suy ra:

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i+1 là thời điểm liền kề i:

Suy ra:

Đặt

Hay

Vậy:

Xác định P0

Với

Nên

A = 10

B = 5

P0= 0

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | *yi* | Pi | xi |
| 0  1  2  3  4  5  6  7  8  9  10 | 2  3  4  5  6  7  8  9  10  11  12 | 0  -5  0  -5  0  -5  0  -5  0  -5  0 | 8  7  7  6  6  5  5  4  4  3  3 |

Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Midpoint()

{

int x;

int y;

float p0;

float p;

int A;

int B;

initwindow(400,400);

putpixel(xa,ya,255);

x=xa;

y=ya;

putpixel(x,y,125);

//xac dinh cac he so

A=yb-ya;

B=-(xb-xa);

p0= (float) (-A/2+B);

p=p0;

while(y<yb)

{

if(p<0)

{

p=(float)(p+B);

}

else

{

p=(float)(p-A+B);

x--;

}

y=y+1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

//closegraph();

}

**Câu hỏi 8. Sử dụng thuật toán Midpoint vẽ đoạn thẳng đi qua 2 điểm A(3;12) và B(8;2)**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xA, yA, xB, yB là các số nhập từ bàn phím

**Bài làm:**

xi,yi

xi+1,yi -1

xi,yi-1

Q(x,y)

P(xi+1/2,yi-1)

**1.1.** Từ phương trình đường thẳng đi qua 2 điểm A và B

Với

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i:

Với A = dy = (yB-yA) = (2-12) = -10

B = -dx = -(xB-xA) = -(8-3) = -5

Suy ra:

Phương trình đường thẳng đi qua điểm Midpoint tại thời điểm i+1 là thời điểm liền kề i:

Suy ra:

Đặt

Hay

Vậy:

Xác định P0

Với

Nên

A = -10

B = -5

P0= 0

|  |  |  |  |
| --- | --- | --- | --- |
| Bước thứ i | *yi* | Pi | xi |
| 0  1  2  3  4  5  6  7  8  9  10 | 12  11  10  9  8  7  6  5  4  3  2 | 0  -5  0  -5  0  -5  0  -5  0  -5  0 | 3  4  4  5  5  6  6  7  7  8  8 |

Lập trình mô phỏng

//khai bao thu vien

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int xa;

int xb;

int ya;

int yb;

float m;

//chuong trinh con

void nhapxy()

{

printf("xA:= ");

scanf("%d",&xa);

printf("yA:= ");

scanf("%d",&ya);

printf("xB:= ");

scanf("%d",&xb);

printf("yB:= ");

scanf("%d",&yb);

m=(float) (yb-ya)/(xb-xa);

printf("he so goc m = %f",m);

}

void Midpoint()

{

int x;

int y;

float p0;

float p;

int A;

int B;

initwindow(400,400);

putpixel(xa,ya,255);

x=xa;

y=ya;

putpixel(x,y,125);

//xac dinh cac he so

A=yb-ya;

B=-(xb-xa);

p0= (float) (-A/2+B);

p=p0;

while(y>yb)

{

if(p<0)

{

p=(float)(p-B);

}

else

{

p=(float)(p+A-B);

x++;

}

y=y-1;

putpixel(x,y,255);

delay(100);

printf("%d,%d\t",x,y);

}

//closegraph();

}

**Câu hỏi. Sử dụng thuật toán Bresenham vẽ đường tròn tâm O(0;0) và R là 5**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xO, yO, và R là các số nhập từ bàn phím

Ta chia đường tròn là làm 8 phần, do tính chất đối xứng của nó.

A close up of a map

Description automatically generatedNên khi ta tính toán được 1 điểm có tọa độ x,y đồng nghĩa với việc ta xác định đầy đủ 8 điểm cho 8 phần như sau:

putpixel (𝑥𝑐+𝑥 , 𝑦𝑐+𝑦)

putpixel (𝑥𝑐+𝑥 , 𝑦𝑐−𝑦)

putpixel (𝑥𝑐+𝑦 , 𝑦𝑐−𝑥)

putpixel (𝑥𝑐−𝑦 , 𝑦𝑐−𝑥)

putpixel (𝑥𝑐−𝑥 , 𝑦𝑐−𝑦)

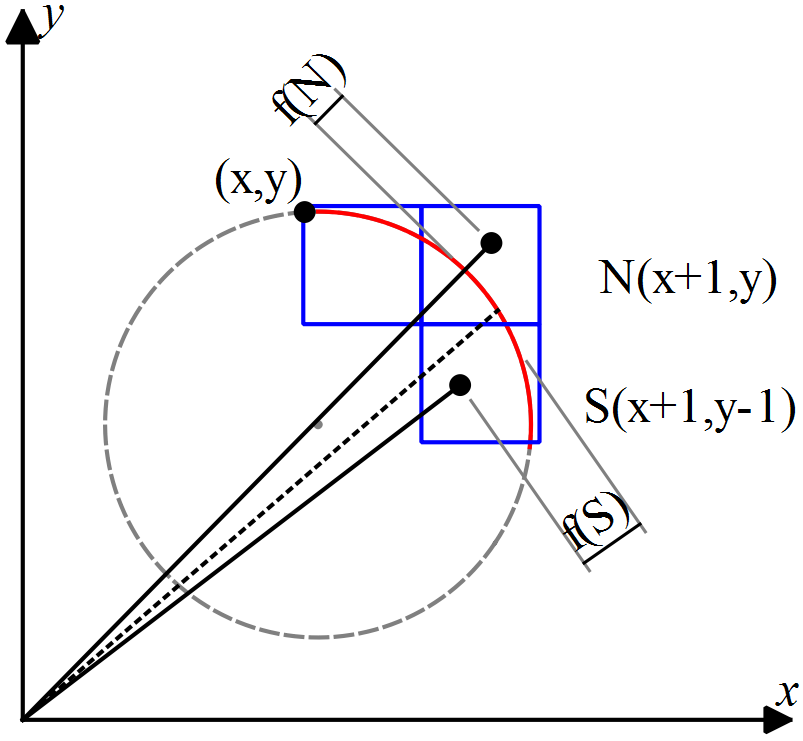
putpixel (𝑥𝑐−𝑥 , 𝑦𝑐+𝑦)

putpixel (𝑥𝑐−y , 𝑦𝑐+x)

putpixel (𝑥𝑐+y , 𝑦𝑐+x)

Trong đó: xc, yc là tâm của đường tròn

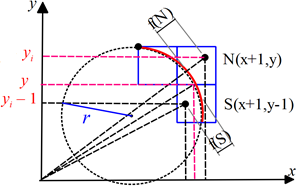
Ta xét các điểm tạo ra từ góc phần tư thứ 2: từ 450 đến 900, thực hiện theo hướng x, y



Áp dụng định lý pithagoras: cho f(N) và f(S)

Cụ thể:

Hay

Hay

Nếu ,

Ngược lại:

Từ

Lập bảng 1/8 đường tròn

Nếu ,

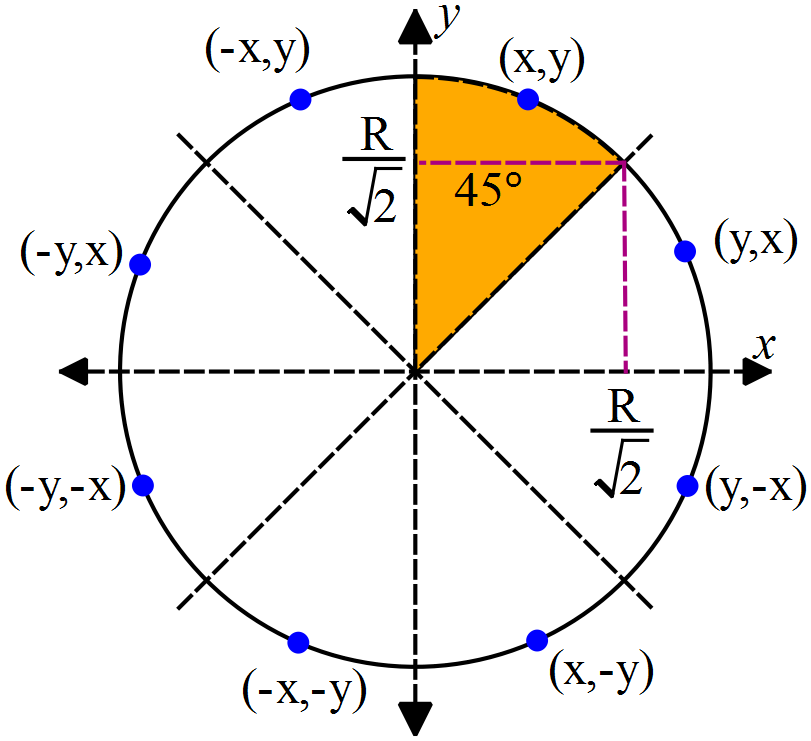
Ngược lại:

|  |  |  |  |
| --- | --- | --- | --- |
| Bước i | xi | yi | d |
| 0 | 0 | 5 | -7 |
| 1 | 1 | 5 | -7+4.0+6 = -1 |
| 2 | 2 | 5 | -1+4.1+6=9 |
| 3 | 3 | 4 | 9+4.2-4.5+10=7 |
| 4 | 4 | 3 | 7+4.3-4.4+10=13 |
| 5 | 5 | 2 | 13+4.4-4.3+10=25 |

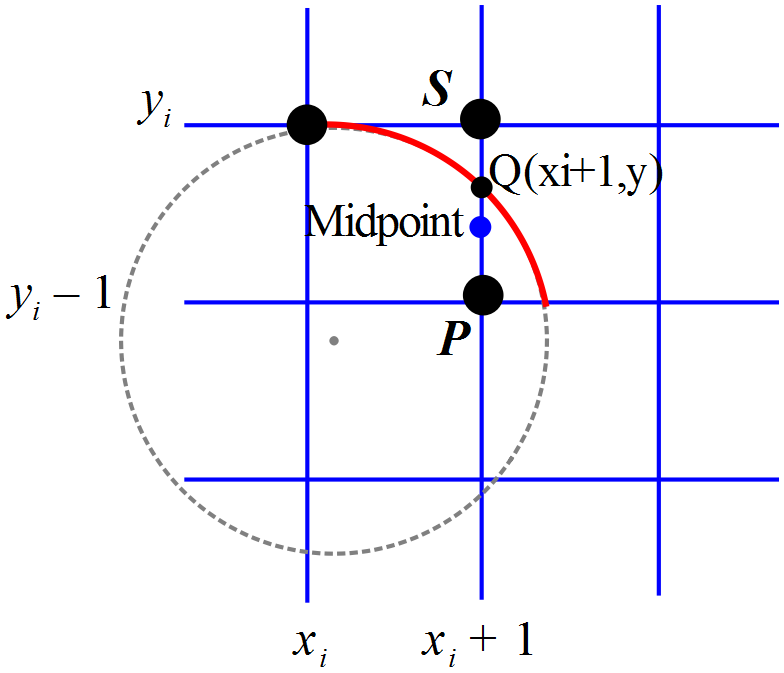
**Câu hỏi. Sử dụng thuật toán Bresenham vẽ đường tròn tâm O(0;0) và R là 5**

**1.1** Trình bày các bước để thực hiện giải thuật trên

**1.2** Lập trình mô phỏng các bước trên với xO, yO, và R là các số nhập từ bàn phím

Ta chia đường tròn là làm 8 phần, do tính chất đối xứng của nó.

Do tính đối xứng của đường tròn (C) nên ta chỉ cần vẽ cung (C1/8) là cung 1/8 đường tròn, sau đó lấy đối xứng.

Cung (C1/8) được mô tả như sau (cung của phần tô màu)

Xét

Nếu *pi* < 0 , điểm MidPoint nằm trong đường tròn.

Lúc này điểm thực Q gần S hơn nên ta chọn S, tức là *yi*+1 = *yi*

Ngược lại, nếu *pi* ≥ 0, điểm MidPoint nằm ngoài đường tròn.

Lúc này điểm thực Q gần P hơn nên ta chọn P, tức là *yi*+1 = *yi* − 1. Mặt khác:

Do

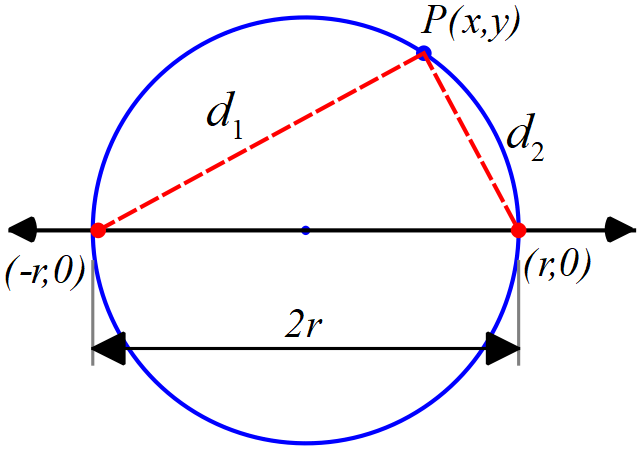
Vậy:

, nếu *pi* < 0 do ta chọn *yi*+1 = *yi*.

, nếu *pi* ≥ 0 do ta chọn *yi*+1 = *yi*−1

Ta tính giá trị *p*0 ứng với điểm ban đầu (x0,*y*0 ) = (0, *r*).

|  |  |  |  |
| --- | --- | --- | --- |
| Bước i | xi | yi | p |
| 0 | 0 | 5 | 5/4-5=-15/4 |
| 1 | 1 | 5 | -15/4+(2.0+3) = -3/4 |
| 2 | 2 | 5 | -3/4+(2.1+3)=17/4 |
| 3 | 3 | 4 | 7/4+2.2-2.5+5=3/4 |
| 4 | 4 | 3 | 3/4+2.3-2.4+5=15/4 |
| 5 | 5 | 2 | 15/4+2.4-2.3+5=21/2 |

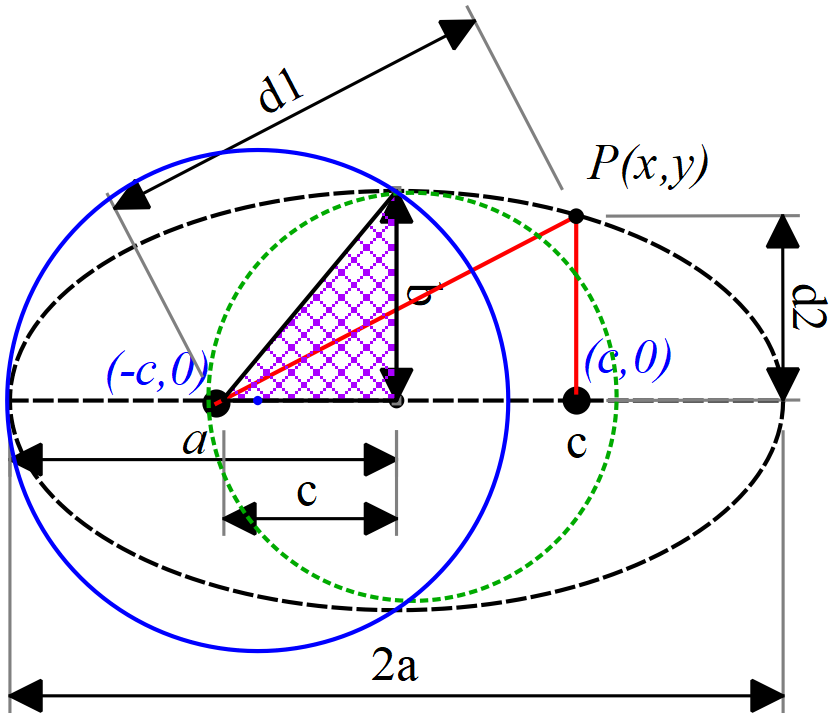
Tìm công thức của phương trình đường tròn:

Trong không gian 2D, khoảng cách d được quy định:

Ta có:

Bình phương 2 vế:

Hay:

**Chứng minh phương trình Ellipse:**

Ta có: Khoảng cách

Trong không gian 2D, khoảng cách d được tính:

Bình phương 2 vế:

Bình phương 2 vế:

Chia 2 vế cho

Chia 2 vế cho , ta có:

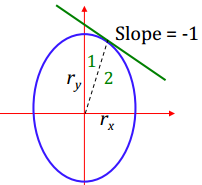
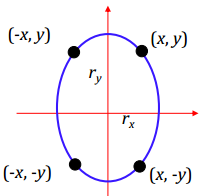
Dựa vào hình tam giác (màu tím), áp dụng định lý pitago:

Ta được:

**Câu hỏi. Sử dụng thuật toán Midpoint vẽ Ellipse tâm O(0;0), ry=2 và rx=4**

**1.1** Trình bày các bước để thực hiện giải thuật trên

Các tính chất của Ellipse:



1. Đối xứng qua các góc phần tư
2. Không đối xứng qua các quãng 8 của các góc phần tư

Do đó, chúng ta sẽ tính toán các pixel của Ellipse thông qua 1 góc phần tư thứ nhất. Sau đó, các điểm còn lại của các góc phần tư khác sẽ thực hiện thông qua các phép đối xứng

Phương trình Ellipse:

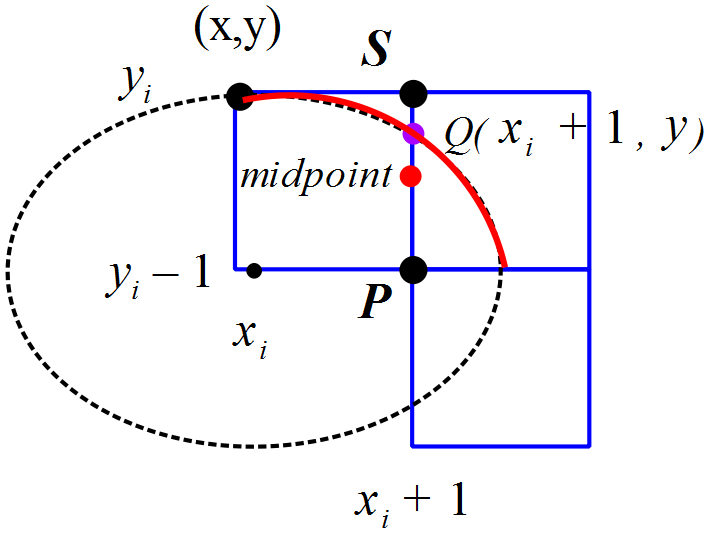
Từ pt Ellipse ở trên, ta có hệ số góc:

Tại vùng (1) ứng với hệ số góc m<1, ta có: x=x+1 và y=f(x)

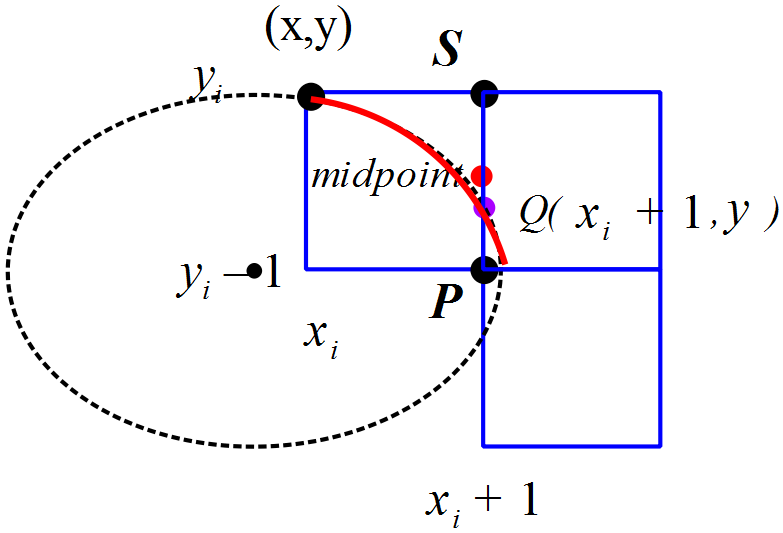
Tại vùng (2) ứng với hệ số góc m>1, ta có: y=y-1 và x=f(y)

Bắt đầu tại tâm (0,r), thực hiện dịch chuyển x tăng dần cho đến khi chạm biên của vùng 1 và vùng 2. Sau đó, dịch chuyển y tăng dần từ biên vùng 2 cho đến hết góc phần tư thứ nhất.

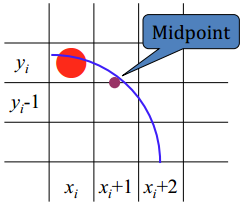
**Trường hợp 1 (vùng 1):**

Ta có:

* Nếu , điểm midpoint nằm bên trong Ellipse. Hay S là điểm kế tiếp.



* Ngược lại, điểm midpoint nằm ngoài Ellipse. Hay P là điểm kế tiếp.

Ta có:

TH1. , nếu , hay

TH2. , nếu , hay

**Xác định**

**Trường hợp 2 (vùng 2):**

Từ pt:

Nếu , điểm midpoint nằm bên ngoài ellipse, gần đường biên hơn

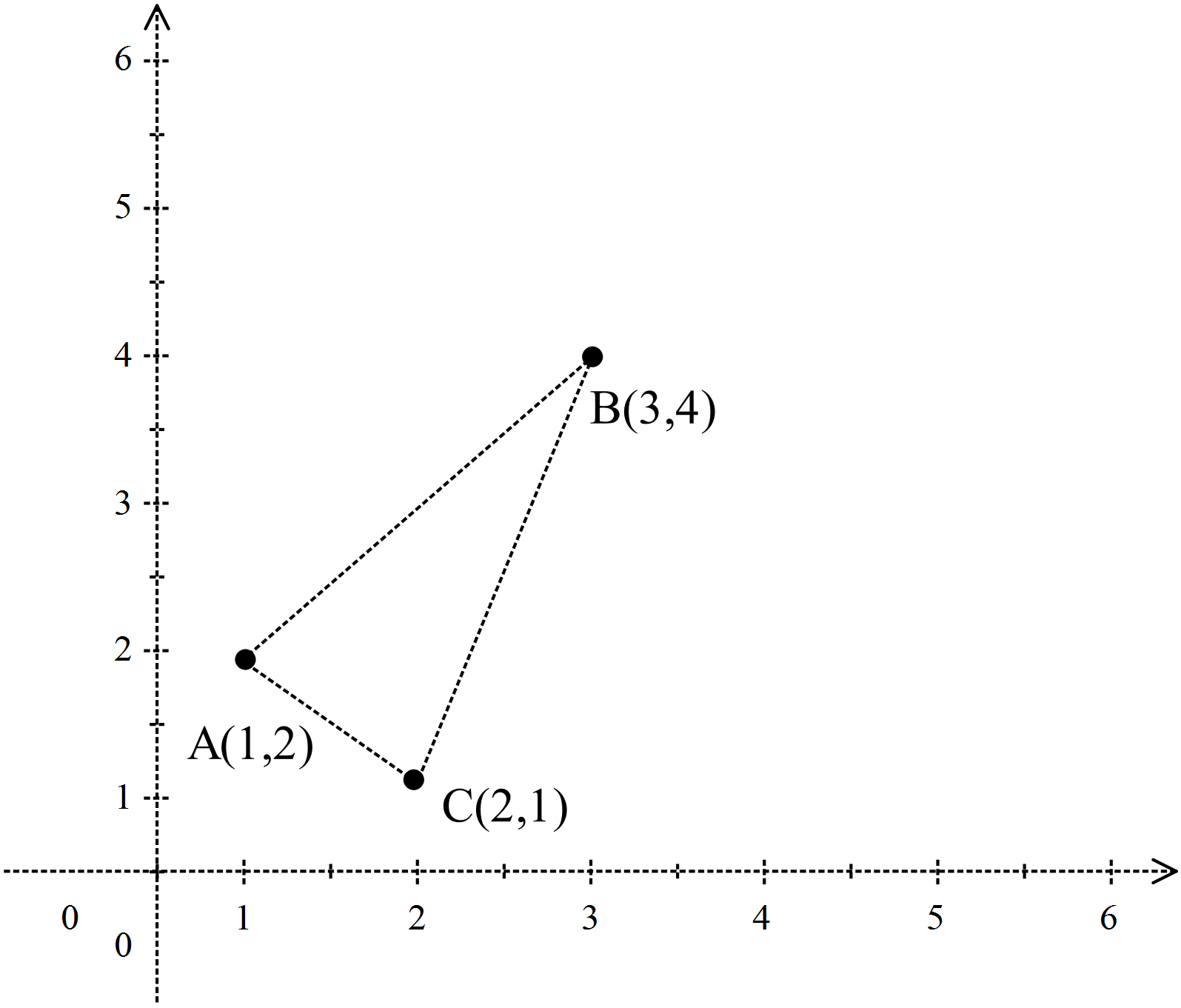
*A picture containing text

Description automatically generated*Ngược lại, , điểm midpoint nằm bên trong ellipse, gần đường biên hơn

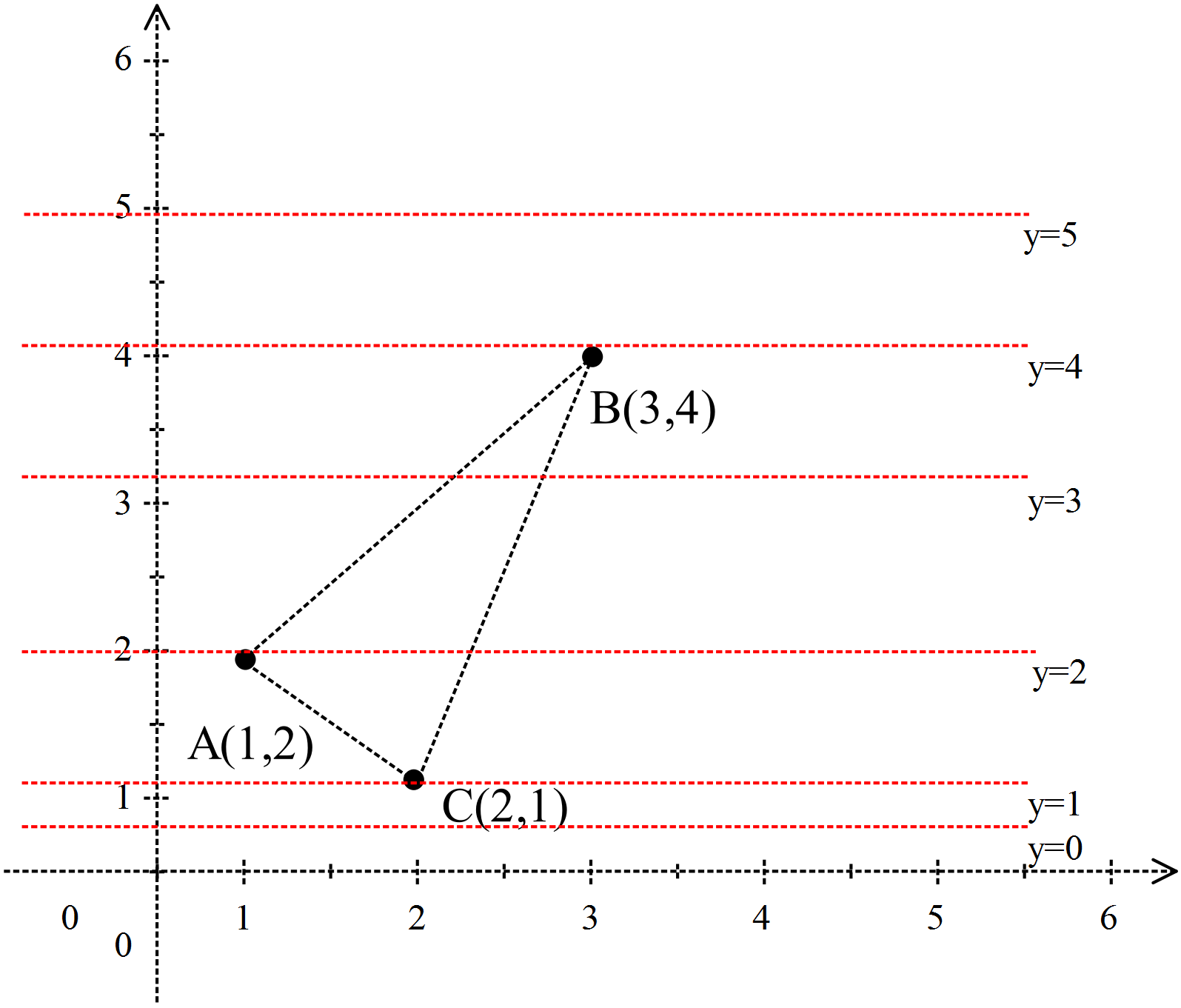
TH1. *, nếu >0, hay*

*TH2.* *, nếu <=0, hay +1*

**Xác định**

Tô màu đa giác có tọa độ như sau

Ma trận các đỉnh của tam giác

Xác định giá trị dx và dy

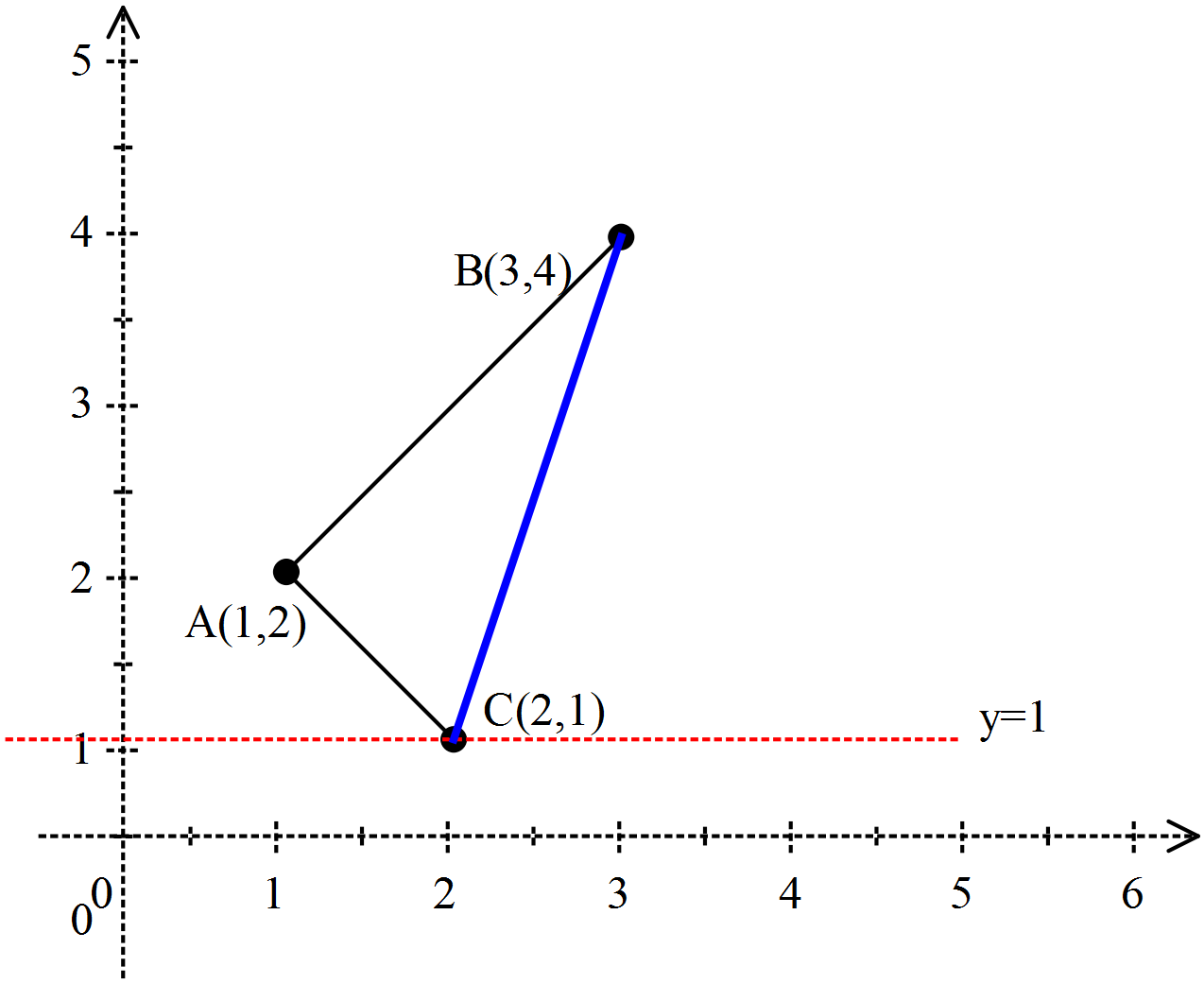
Hệ số góc dy/dy

Hệ số góc dx/dy

 Tìm các điểm giao cắt các cạnh của tam giác với các dòng quét y=1( đến y=4 (. Nếu tồn tại giao điểm, xác định tọa độ tại điểm cắt với giá trị (x*giao điểm*, y)

* Nếu scanline y bất kỳ cắt qua cạnh XY nào đó khi và chỉ khi:
* Tọa đọa điểm cắt được xác định thông qua phương trình sau:
  + hay

***Y=1***

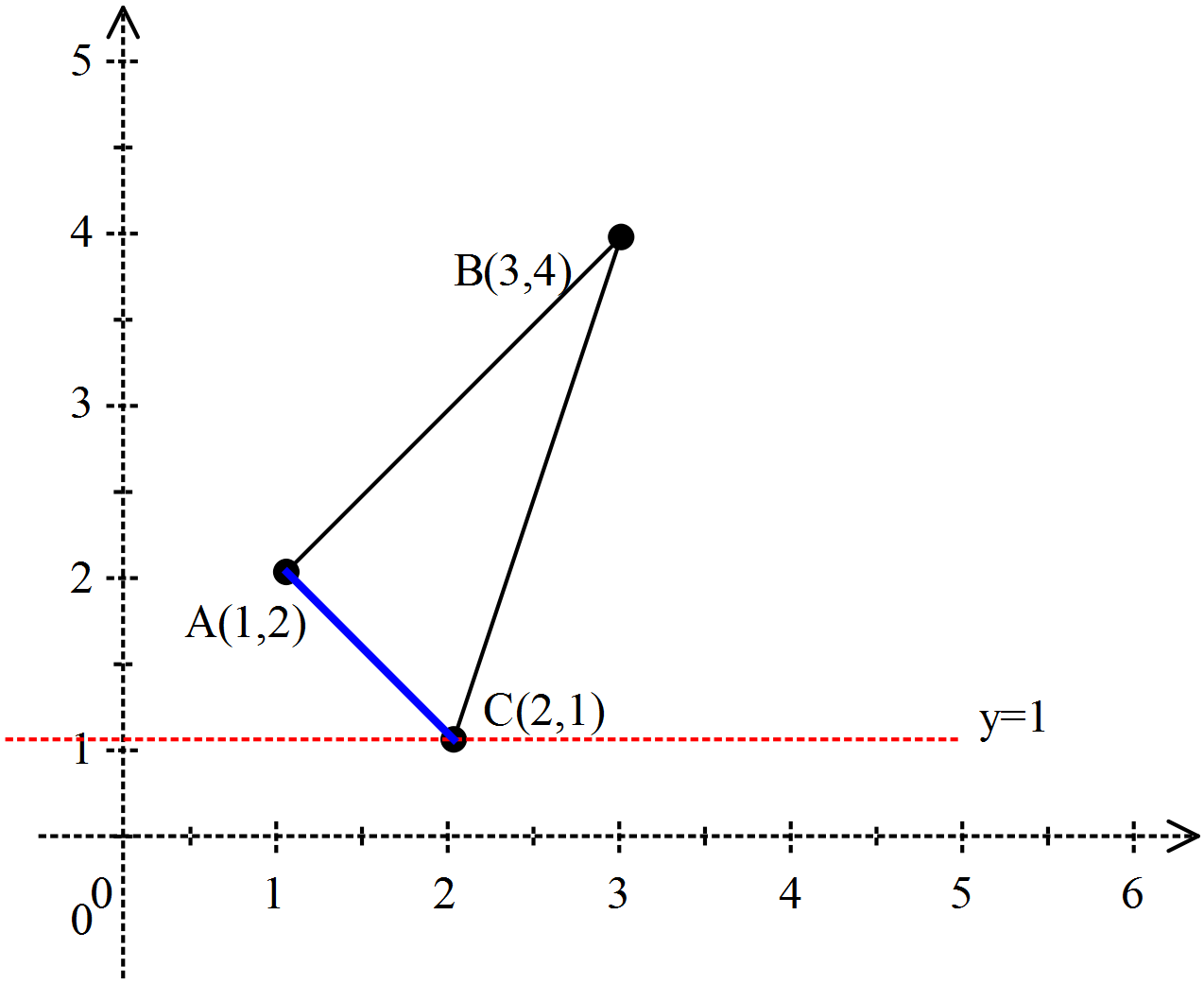
* + Xét cạnh AB, rõ ràng cả 2 đỉnh A và B đều không thỏa mãn biểu thức để chứng tỏ đường thẳng y=1 cắt ngang cạnh AB (. Do đó không cần xác định tọa độ điểm giao nhau giữa y=1 và đoạn AB.
  + Xét cạnh BC

yB = 4 > 1 và yC = 1 <=1. Nên hoành độ giao điểm của BC

xgđ[0] = xB + m(y-yB) hoặc xgđ[0] = xC + m(y-yC)

xgđ[0] = 3 + 1/3(1-4) hoặc xgđ[0] = 2 + 1/3(1-1)

xgđ[0] = 2 hoặc xgđ[0] = 2

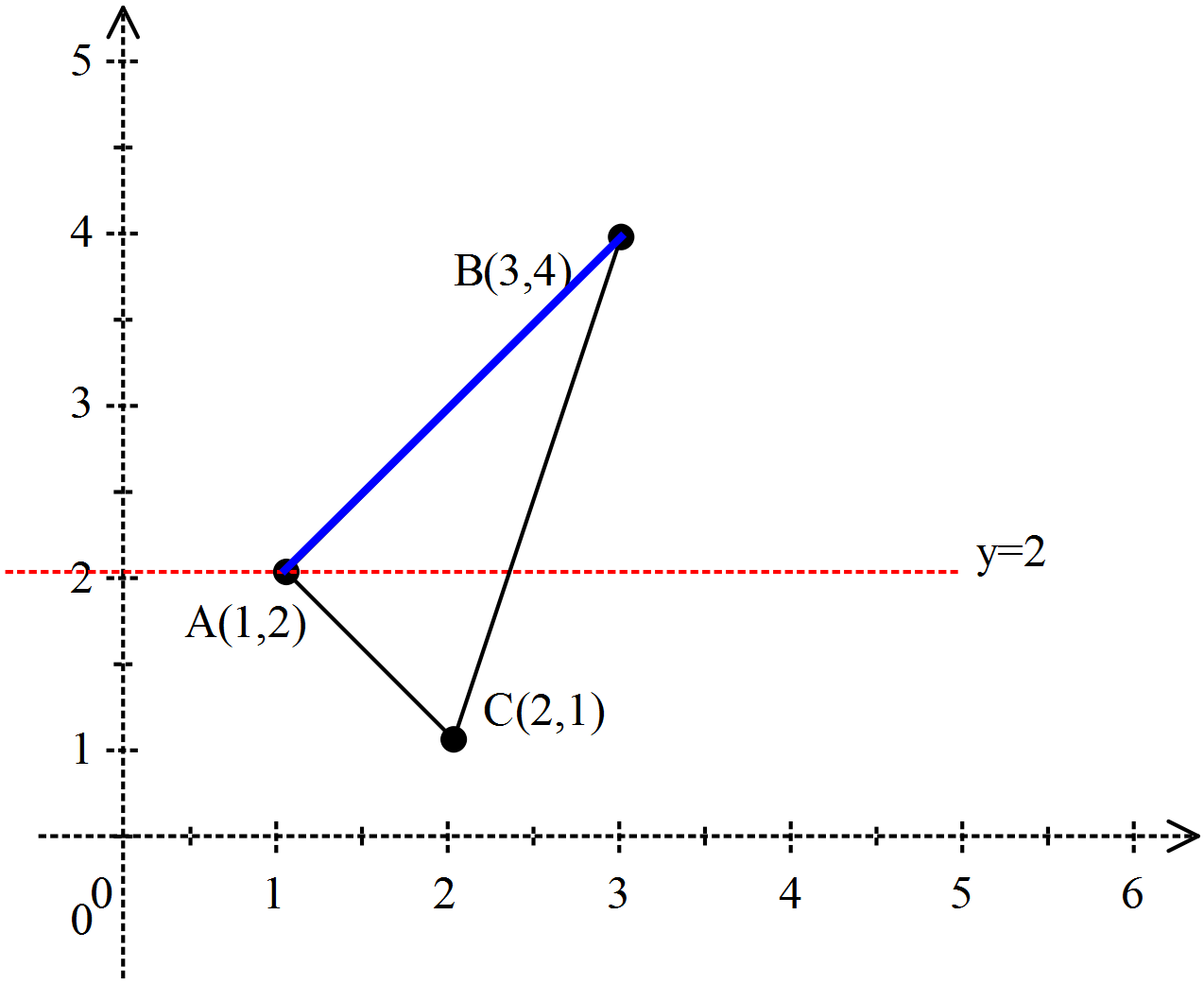
* + Xét cạnh CA

yC = 1 <= 1 và yA = 2 >1

xgđ[1] = xA + m(y-yA) hoặc xgđ[1] = xC + m(y-yC)

xgđ[1] = 1 - 1(1-2) hoặc xgđ[1] = 2 - 1(1-1)

xgđ[1] = 2 hoặc xgđ[1] = 2

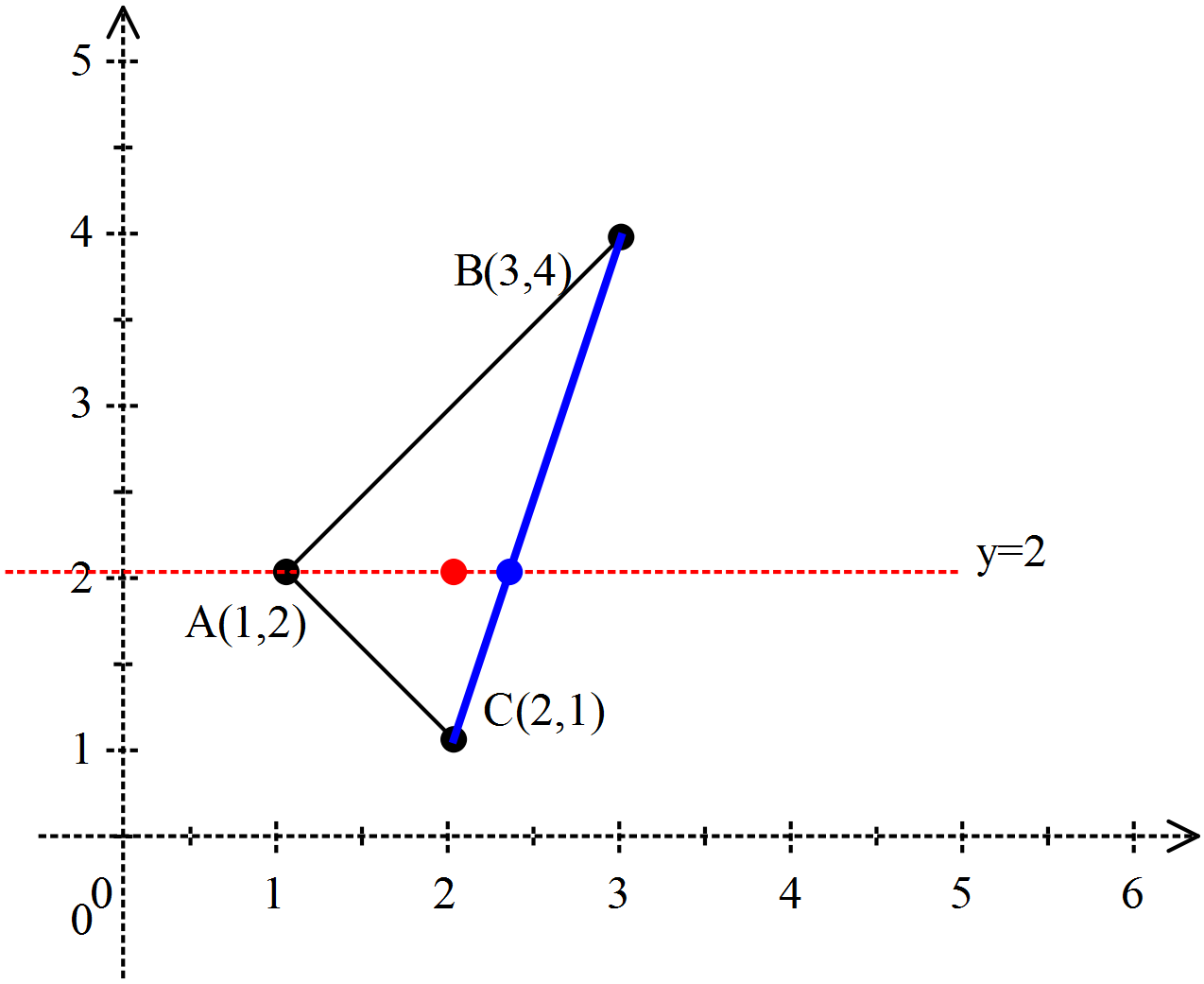
***Y=2***

* + Xét cạnh AB

yA = 2 <=2 và yB = 4 > 2

xgđ[2] = xA + m(y-yA) hoặc xgđ[2] = xB + m(y-yB)

xgđ[2] = 1 + 1(2-2) hoặc xgđ[2] = 3 + 1(2-4)

xgđ[2] = 1 hoặc xgđ[2] = 1

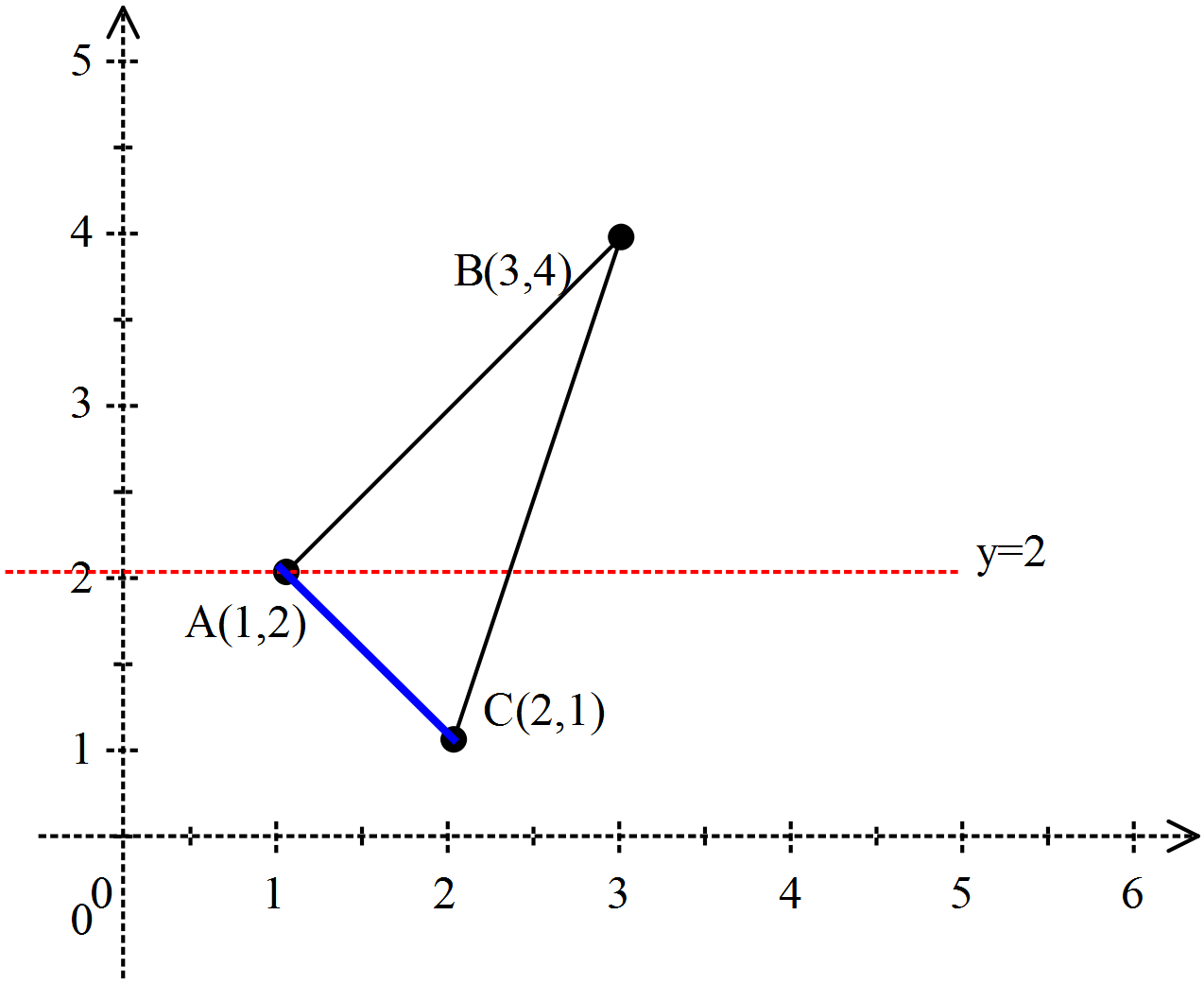
* + Xét cạnh BC

yC = 1 <= 2 và yB = 4 >2

xgđ[3] = xB + m(y-yB) hoặc xgđ[3] = xC + m(y-yC)

xgđ[3] = 3 + 1/3(2-4) hoặc xgđ[3] = 2 + 1/3(2-1)

xgđ[3] = 2 hoặc xgđ[3] = 2

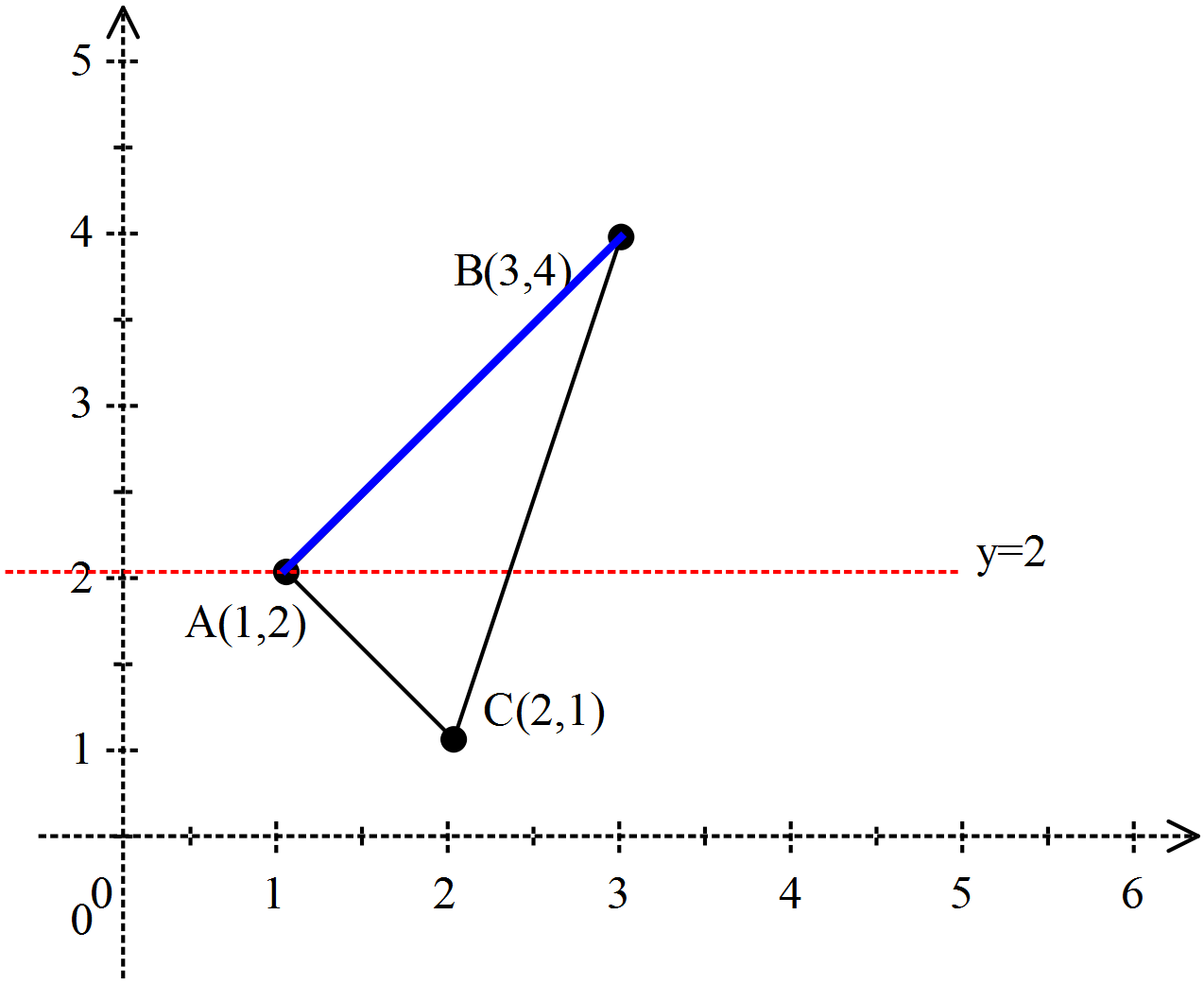
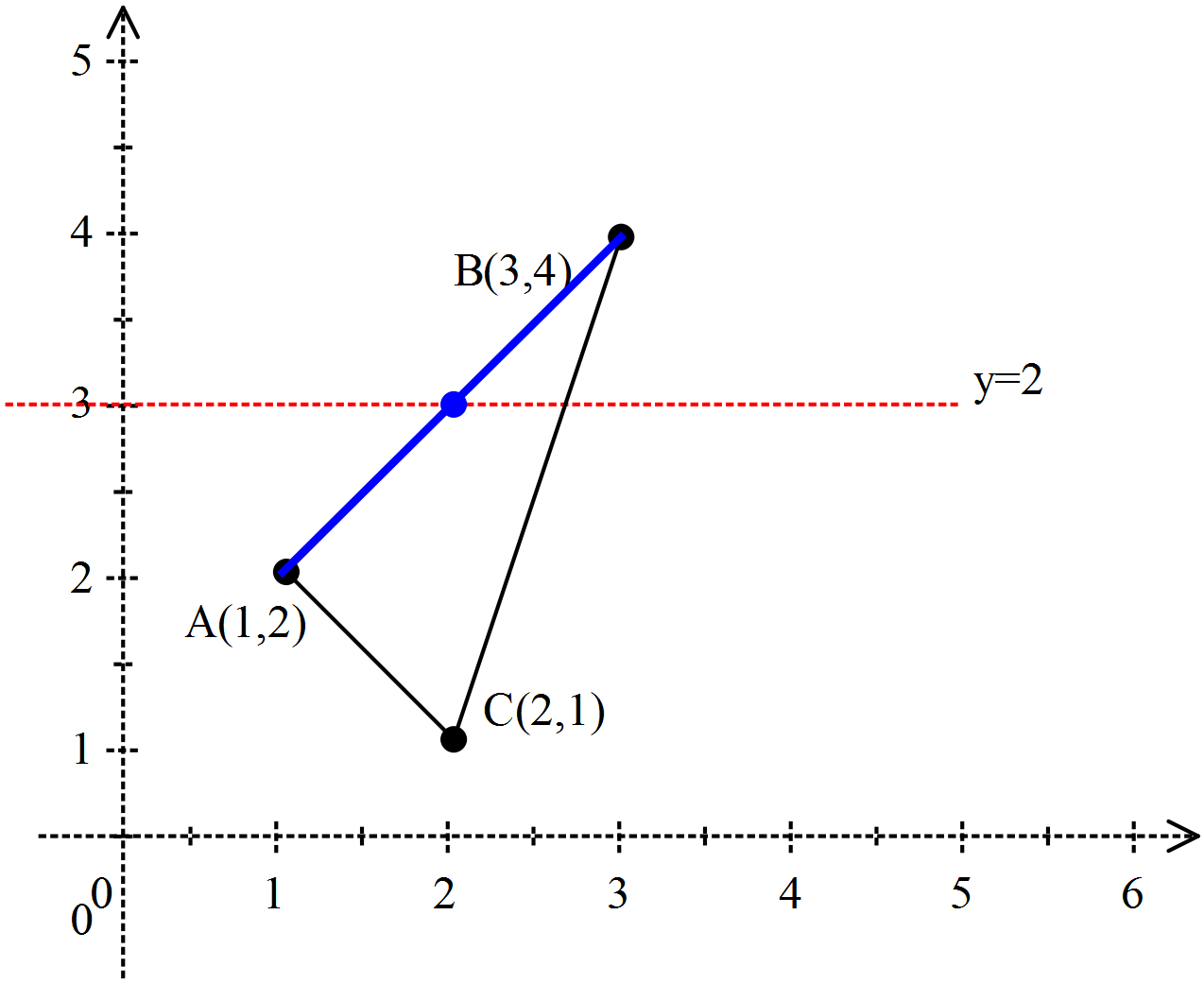
* + Xét cạnh CA

yC = 1 < 2 và yA = 2 >=2

xgđ[4] = xA + m(y-yA) hoặc xgđ[4] = xC + m(y-yC)

xgđ[4] = 1 - 1(2-2) hoặc xgđ[4] = 2 - 1(2-1)

xgđ[4] = 1 hoặc xgđ[4] = 1

***Y=3***

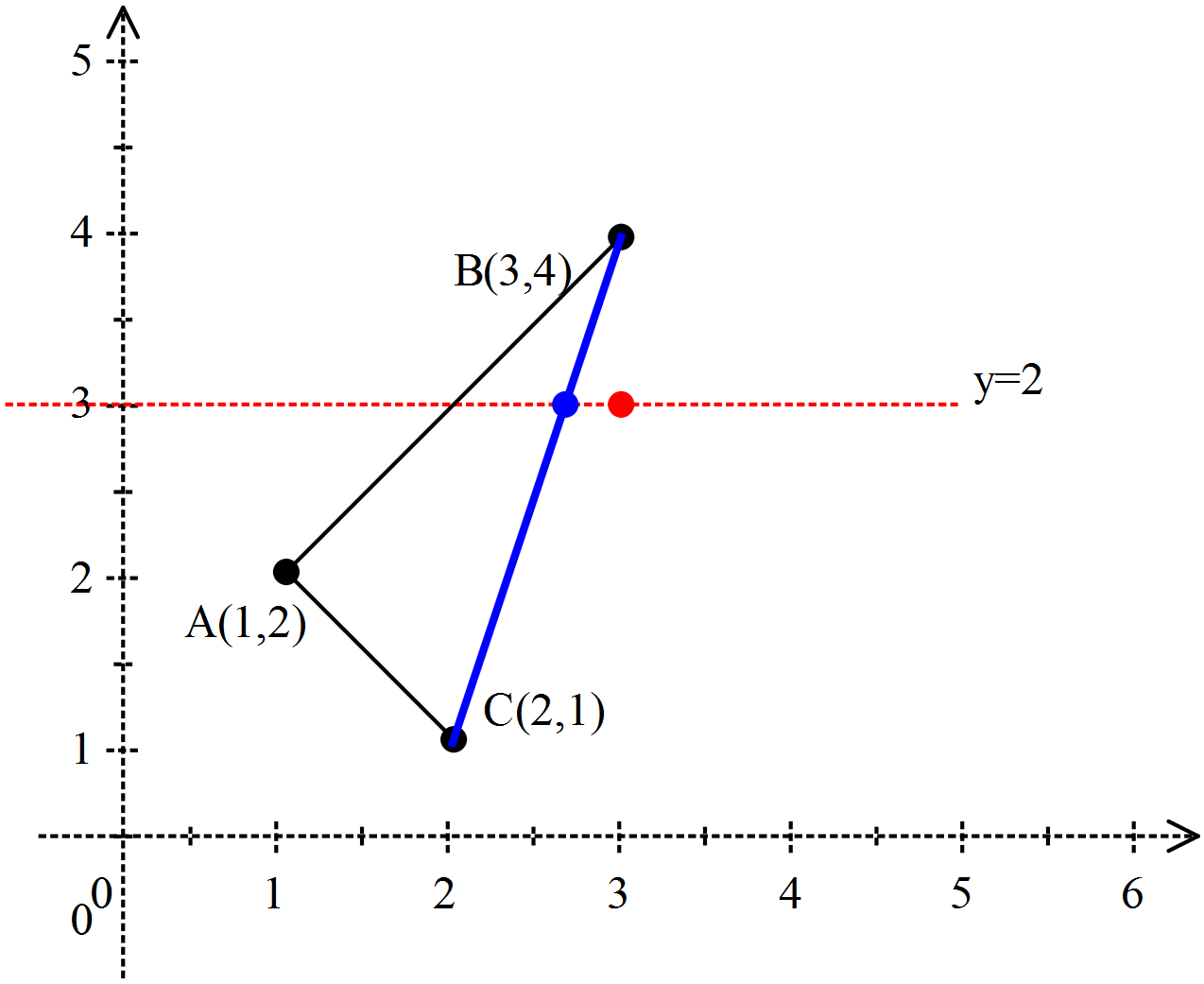
* + Xét cạnh AB

yA = 2 <=3 và yB = 4 > 3

xgđ[5] = xA + m(y-yA) hoặc xgđ[5] = xB + m(y-yB)

xgđ[5] = 1 + 1(3-2) hoặc xgđ[5] = 3 + 1(3-4)

xgđ[5] = 2 hoặc xgđ[5] = 2

* + Xét cạnh BC

yC = 1 <= 2 và yB = 4 >2

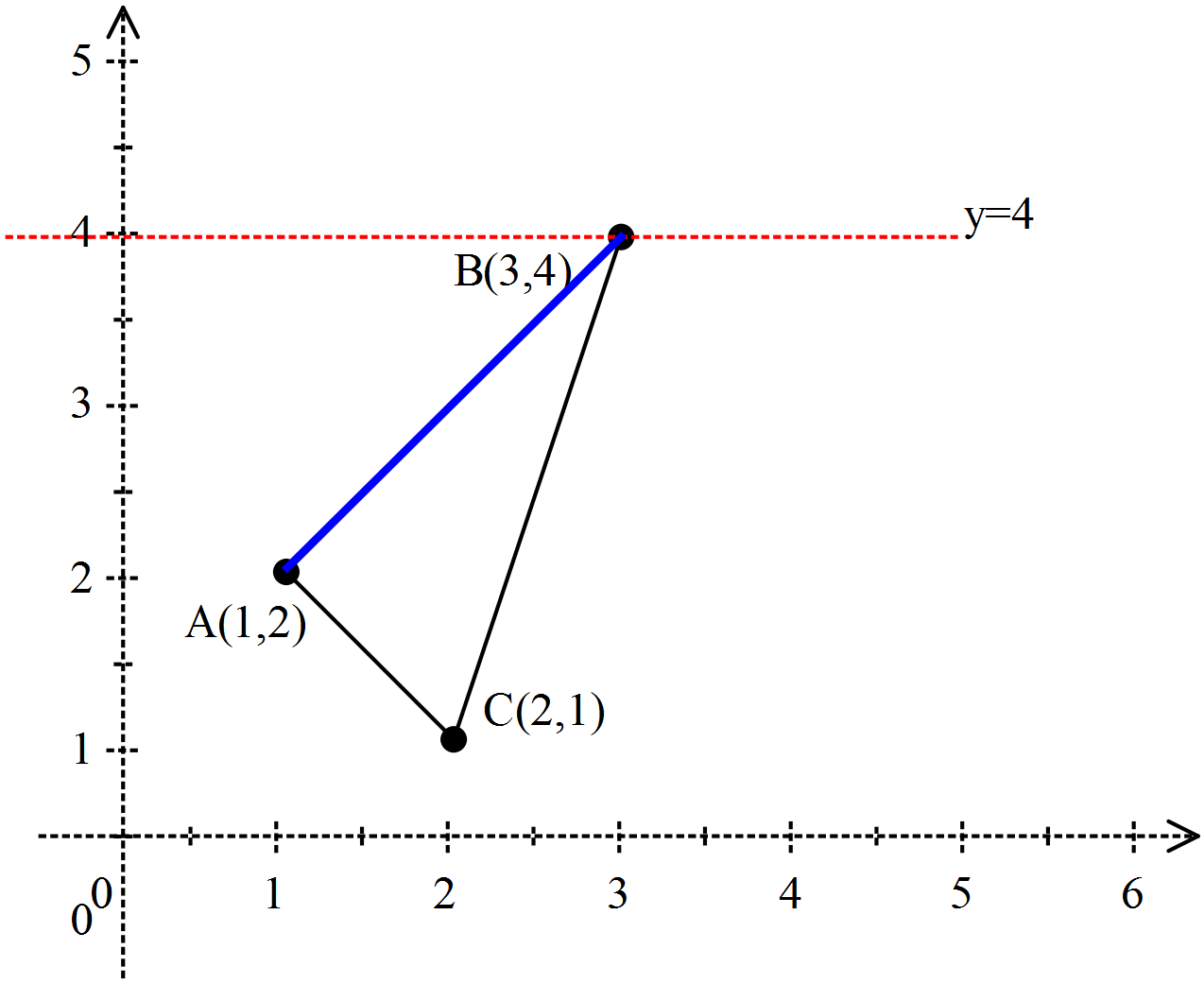
xgđ[3] = xB + m(y-yB) hoặc xgđ[3] = xC + m(y-yC)

xgđ[3] = 3 + 1/3(2-4) hoặc xgđ[3] = 2 + 1/3(2-1)

xgđ[3] = 2 hoặc xgđ[3] = 2

* + Xét cạnh CA

Cả 2 đỉnh A và C đều không thỏa mãn biểu thức để chứng tỏ đường thẳng y=3 cắt ngang cạnh AC (.. Do đó không cần xác định tọa độ điểm giao nhau giữa y=3 và đoạn AC

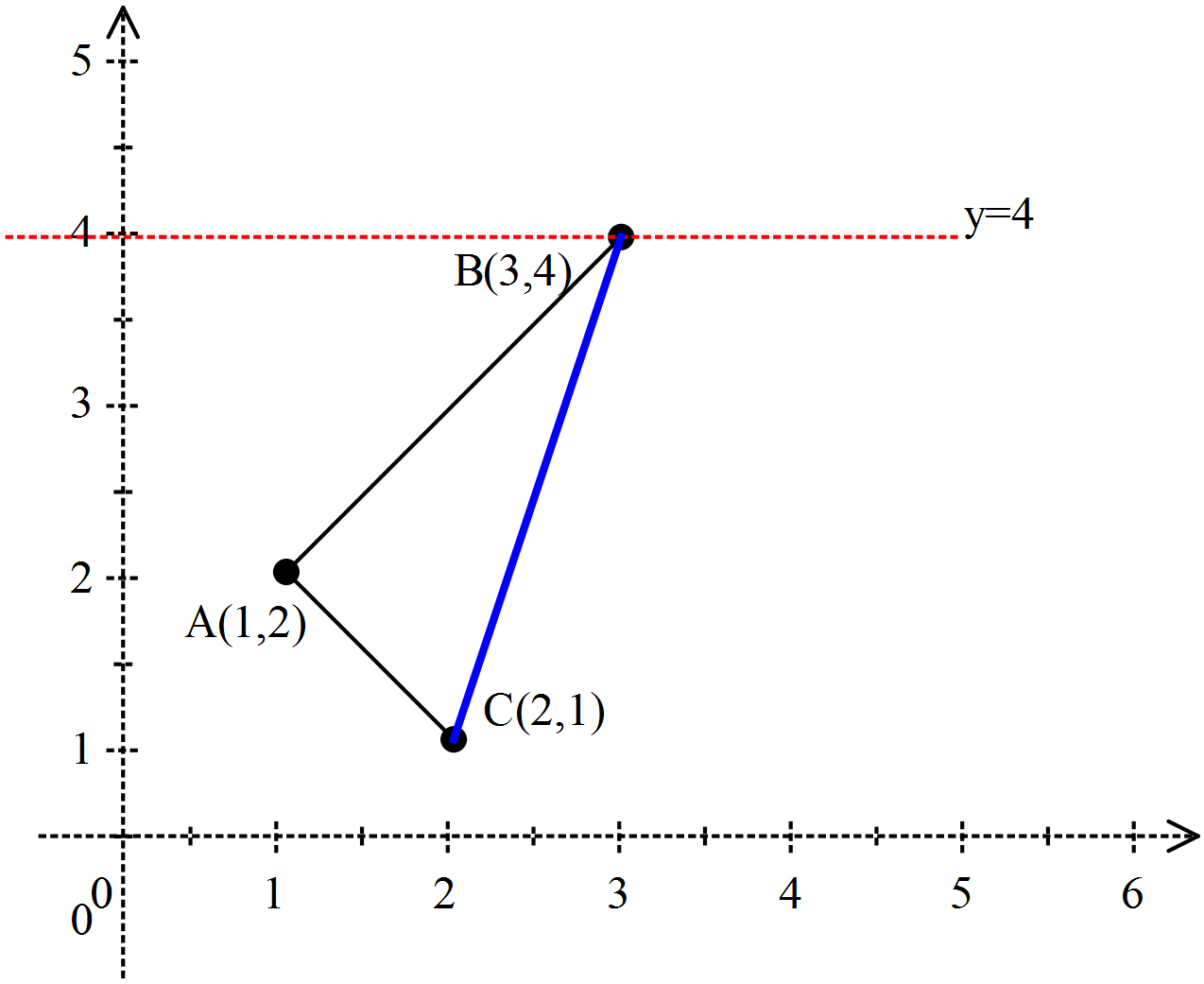
***Y=4***

* + Xét cạnh AB

yA = 2 <4 và yB = 4 >=4

xgđ[7] = xA + m(y-yA) hoặc xgđ[7] = xB + m(y-yB)

xgđ[7] = 1 + 1(4-2) hoặc xgđ[7] = 3 + 1(4-4)

xgđ[7] = 3 hoặc xgđ[7] = 3

* + Xét cạnh BC

yC = 1 < 4 và yB = 4 >= 4

xgđ[8] = xB + m(y-yB) hoặc xgđ[8] = xC + m(y-yC)

xgđ[8] = 3 + 1/3(4-4) hoặc xgđ[8] = 2 + 1/3(4-1)

xgđ[8] = 3 hoặc xgđ[8] = 3

* + Xét cạnh CA

Cả 2 đỉnh A và C đều không thỏa mãn biểu thức để chứng tỏ đường thẳng y=4 cắt ngang cạnh AC (.. Do đó không cần xác định tọa độ điểm giao nhau giữa y=3 và đoạn AC

Tập hợp các giao điểm:

Tọa độ các đoạn thẳng (x*gđ*,y) là:

Tiến hành tô màu: thực hiện vẽ các đường thẳng nối các tọa độ ở trên