**Trường Đại học Nha Trang**

**Bộ môn Kỹ thuật phần mềm**

**🙡🕮🙣**

**HƯỚNG DẪN GIẢI CHI TIẾT VÀ LẬP TRÌNH**

**MÔN HỌC: KỸ THUẬT ĐỒ HỌA**

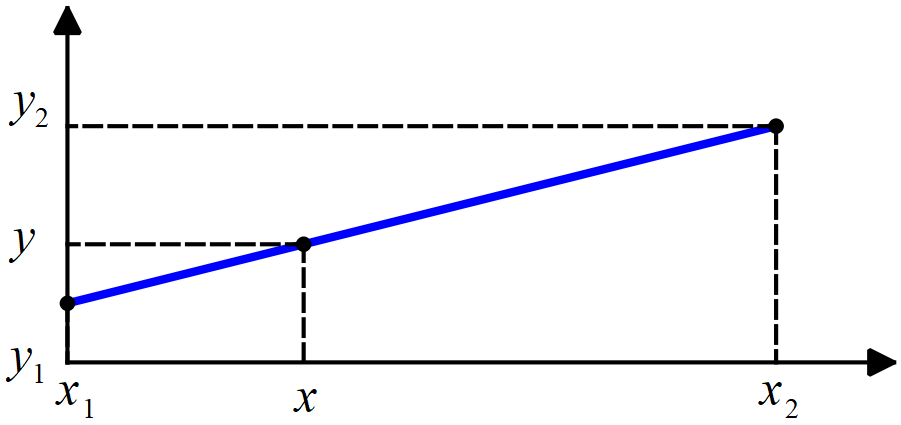
**Biên soạn: ĐOÀN VŨ THỊNH**

**🖂: thinhdv@ntu.edu.vn**

**🕿: 0914949195**

**W:** <https://github.com/thinhdoanvu>

**Nha Trang, 2021**

**Phương trình đường thẳng: y=mx + b**

Chia 2 vế cho và đặt

hay

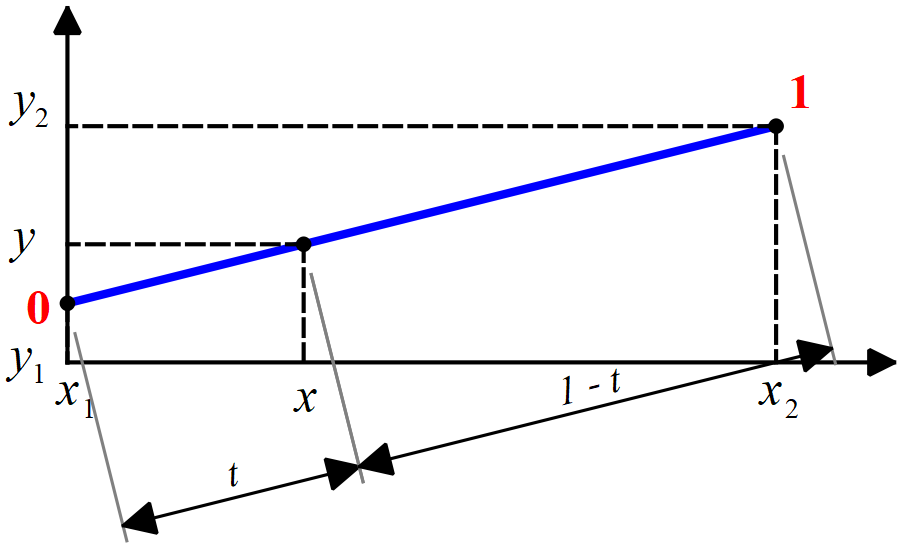
**Phương trình đường thẳng: Ax + By + C**

Từ biểu thức:

Ta có:

Ta được:

**Phương trình đường thẳng:**

Ta có:

Hay

Tương tự:

**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 Bresenham 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 | P0 = A+B/2 = 0  P1 = P0+A+B = -5  P2 = P1+A = 0  P3 = P2+A+B = -5  P4 = P3 + A = 0  P5 = P4 + A + B = -5  P6 = P5 + A = 0  P7 = P6 + A + B = -5  P8 = P7 + A = 0  P9 = P8 + A + B = -5  P10 = P9 + A = 0 | y0= 3  y1 = y0 + 1 = 4  y2 = y1 = 4  y3 = y2 + 1 = 5  y4 = y3 = 5  y5 = y4 + 1 = 6  y6 = y5 = 6  y7 = y6 + 1 = 7  y8 = y7 = 7  y9 = y8 + 1 = 8  y9 = y8 = 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 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  0  -5  0  -5  0  -5  0  -5  0 | 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 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 |

**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 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 |

**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 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 |

**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 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 |

**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 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 |

**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 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 |

**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 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)

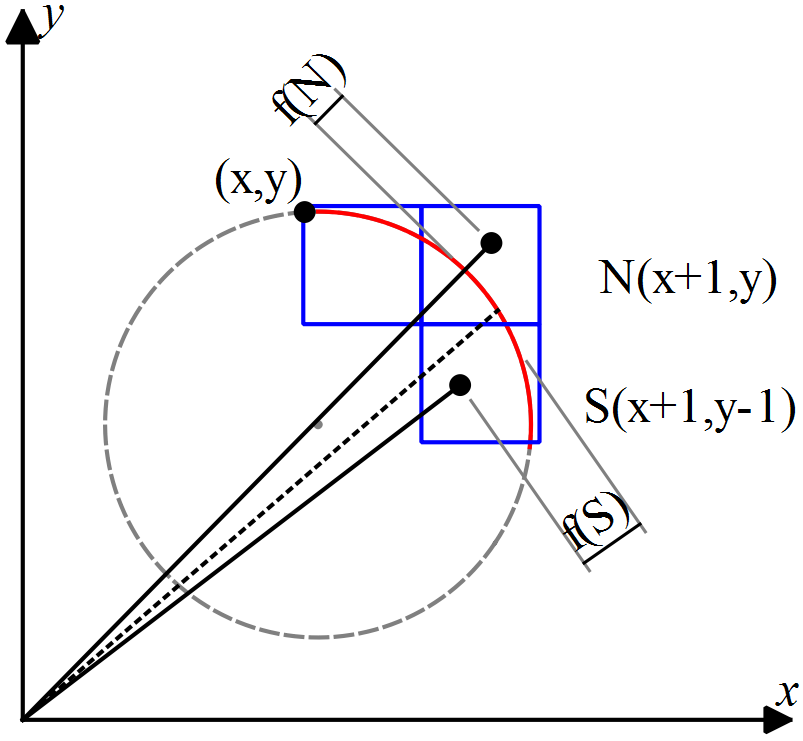
putpixel (𝑥𝑐−𝑥 , 𝑦𝑐+𝑦)

putpixel (𝑥𝑐−y , 𝑦𝑐+x)

putpixel (𝑥𝑐+y , 𝑦𝑐+x)

Trong đó: xc, yc là tâm của đường tròn

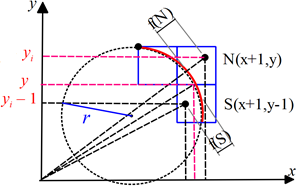
Ta xét các điểm tạo ra từ góc phần tư thứ 2: từ 450 đến 900, thực hiện theo hướng x, y



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 |

**1.2. Lập trình mô phỏng**

//khai bao thu vien

#include<stdio.h>

#include <graphics.h>

//khai bao bien

int xc;

int yc;

int x;

int y;

int r;

int d;

//chuong trinh con

void nhapxy()

{

printf("Nhap tam duong tron, xc= ");

scanf("%d",&xc);

printf("Nhap tam duong tron, yc= ");

scanf("%d",&yc);

printf("Nhap ban kinh duong tron, r= ");

scanf("%d",&r);

}

void drawcircle()

{

putpixel(xc+x,yc+y,255);

putpixel(xc-x,yc+y,255);

putpixel(xc+x,yc-y,255);

putpixel(xc-x,yc-y,255);

putpixel(xc+y,yc+x,255);

putpixel(xc-y,yc+x,255);

putpixel(xc+y,yc-x,255);

putpixel(xc-y,yc-x,255);

}

void Bresenham()

{

initwindow(400,400);

x=0;

y=r;

d=3-2\*r;

while(x<y)

{

drawcircle();

if(d<=0)

{

d=d+4\*x+6;

}

else

{

y--;

d=d+4\*x-4\*y+10;

}

drawcircle();

x++;

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

}

}

//chuong trinh chinh

main()

{ //khoi tao window

nhapxy();

Bresenham();

//closegraph();

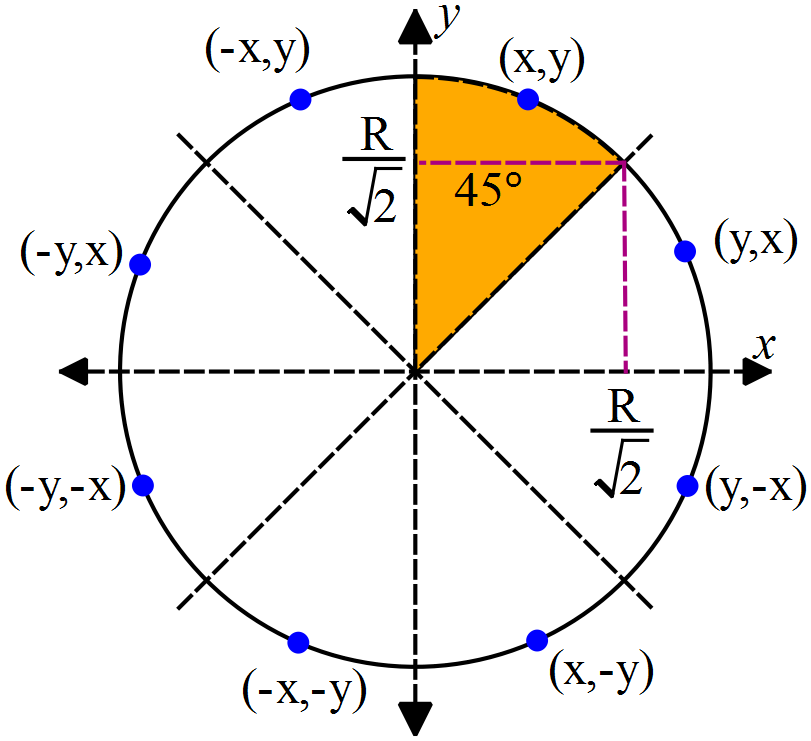
getch();

}

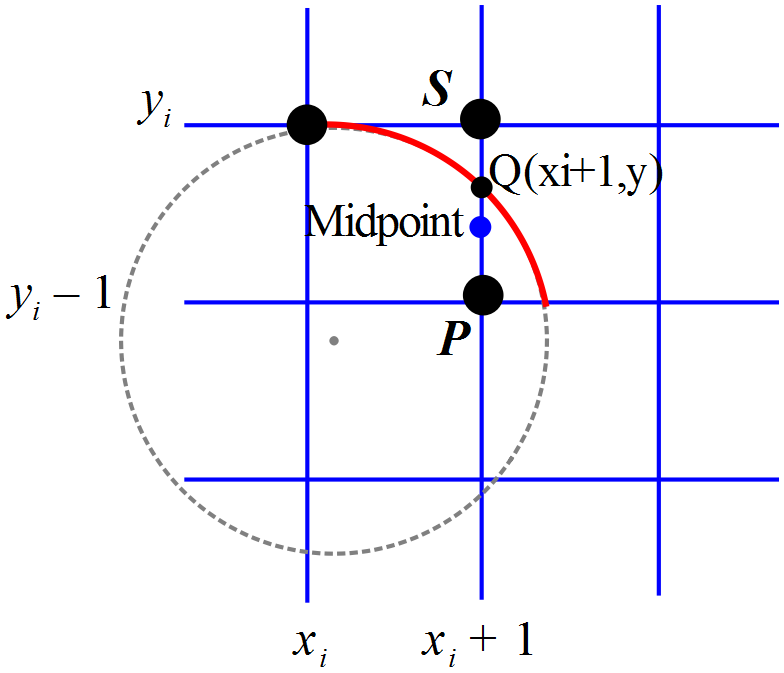
**Câu hỏi. Sử dụng thuật toán Bresenham vẽ đường tròn tâm O(0;0) và R là 5**

**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 |

**1.2. Lập trình mô phỏng**

//khai bao thu vien

#include <graphics.h>

#include <stdio.h>

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

//khai bao bien

int xc;

int yc;

int x;

int y;

int r;

int p;

//chuong trinh con

void nhapxy()

{

printf("Nhap tam duong tron, xc= ");

scanf("%d",&xc);

printf("Nhap tam duong tron, yc= ");

scanf("%d",&yc);

printf("Nhap ban kinh duong tron, r= ");

scanf("%d",&r);

}

void drawcircle()

{

putpixel(xc+x,yc+y,255);

putpixel(xc-x,yc+y,255);

putpixel(xc+x,yc-y,255);

putpixel(xc-x,yc-y,255);

putpixel(xc+y,yc+x,255);

putpixel(xc-y,yc+x,255);

putpixel(xc+y,yc-x,255);

putpixel(xc-y,yc-x,255);

}

void Midpoint()

{

initwindow(400,400);

x=0;

y=r;

p=ROUND(5/4-r);

while(x<y)

{

drawcircle();

if(p<=0)

{

p=p+2\*x+3;

}

else

{

y--;

p=p+2\*x-2\*y+5;

}

drawcircle();

x++;

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

}

}

//chuong trinh chinh

main()

{ //khoi tao window

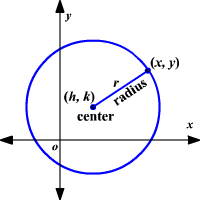
nhapxy();

Midpoint();

//closegraph();

getch();

}

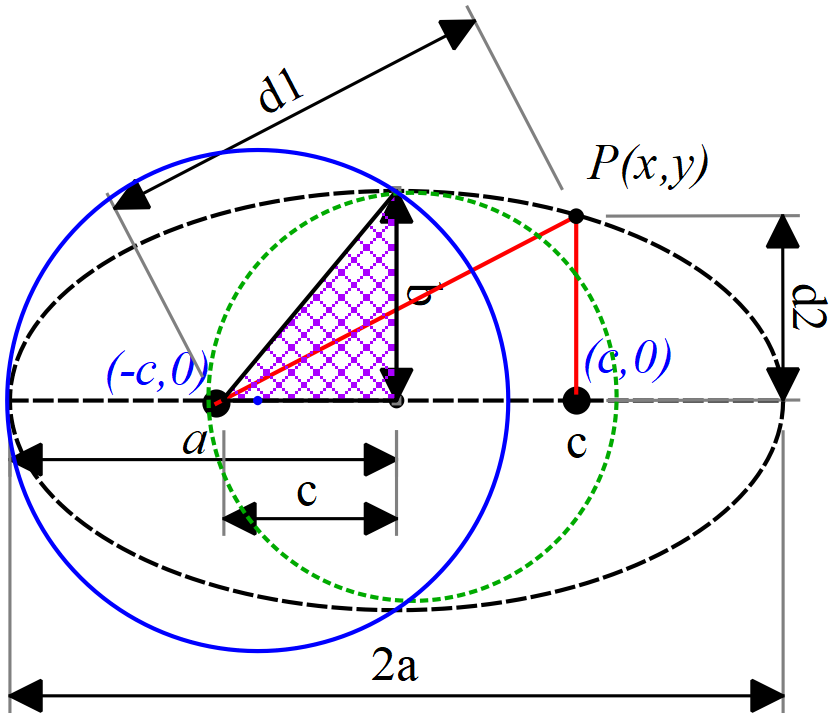
Tìm công thức của phương trình đường tròn:

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:

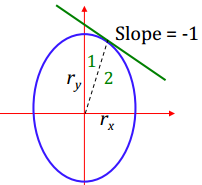
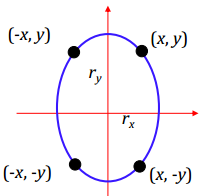
Dựa vào hình tam giác (màu tím), áp dụng định lý pitago:

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:

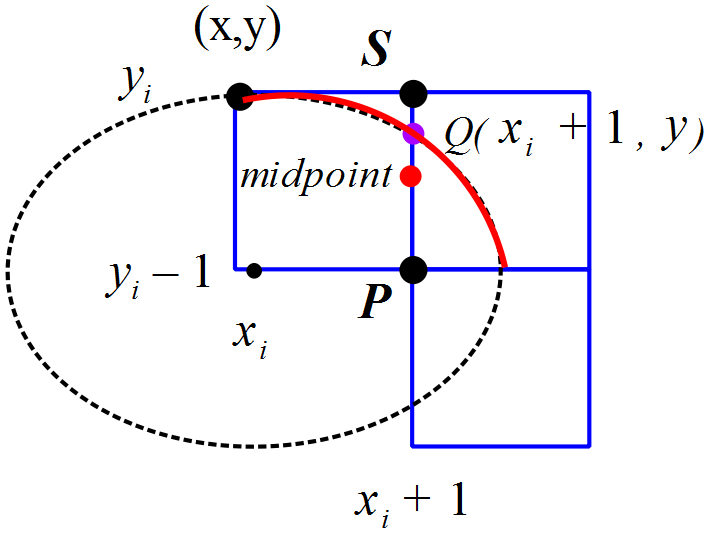
Từ pt Ellipse ở trên, ta có hệ số góc:

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

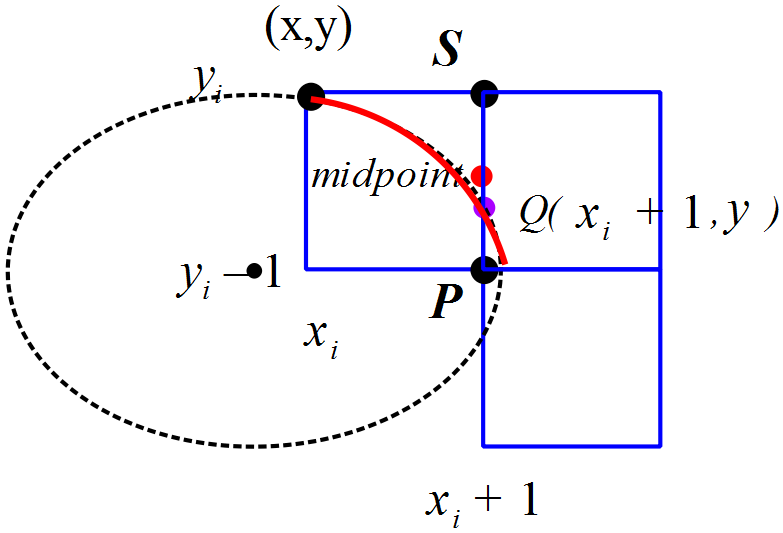
Tại vùng (2) ứng với hệ số góc m>1, ta có: y=y-1 và x=f(y)

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.

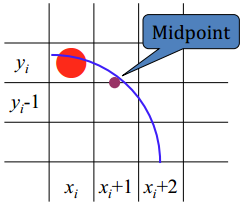
**Trường hợp 1 (vùng 1):**

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**

**1.2. Lập trình mô phỏng**

//khai bao thu vien

#include <graphics.h>

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

#include<stdio.h>

//khai bao bien

int xc;

int yc;

int rx;

int ry;

int x;

int y;

//chuong trinh con

void nhapxy()

{

printf("Nhap tam ellipse, xc= ");

scanf("%d",&xc);

printf("Nhap tam ellipse, yc= ");

scanf("%d",&yc);

printf("Nhap ban kinh truc ngang, rx= ");

scanf("%d",&rx);

printf("Nhap ban kinh truc doc, ry= ");

scanf("%d",&ry);

}

void drawellipse()

{

putpixel(xc+x,yc+y,255);

putpixel(xc-x,yc+y,255);

putpixel(xc+x,yc-y,255);

putpixel(xc-x,yc-y,255);

}

void Ellipse()

{

int p1;

int px;

int py;

int p2;

initwindow(400,400);

x=0;

y=ry;

p1=ROUND(ry\*ry-rx\*rx\*ry+rx\*rx/4);

px=0;

py=2\*rx\*rx\*y;

drawellipse();

while(px<py)

{

x++;

px=px+2\*ry\*ry;

if(p1<=0)

{

p1=p1+ry\*ry+px;

}

else

{

y--;

py=py-2\*rx\*rx;

p1=p1+ry\*ry+px-py;

}

drawellipse();

}

//Region 2

p2=ROUND(ry\*ry\*(x+0.5)\*(x+0.5)+rx\*rx\*(y-1)\*(y-1)-rx\*rx\*ry\*ry);

while (y>0)

{

y--;

py=py-2\*rx\*rx;

if (p2>0)

{

p2=p2+rx\*rx - py;

}

else

{

x++;

px = px+2\*ry\*ry;

p2= p2+rx\*rx - py + px;

}

drawellipse();

}

}

//chuong trinh chinh

main()

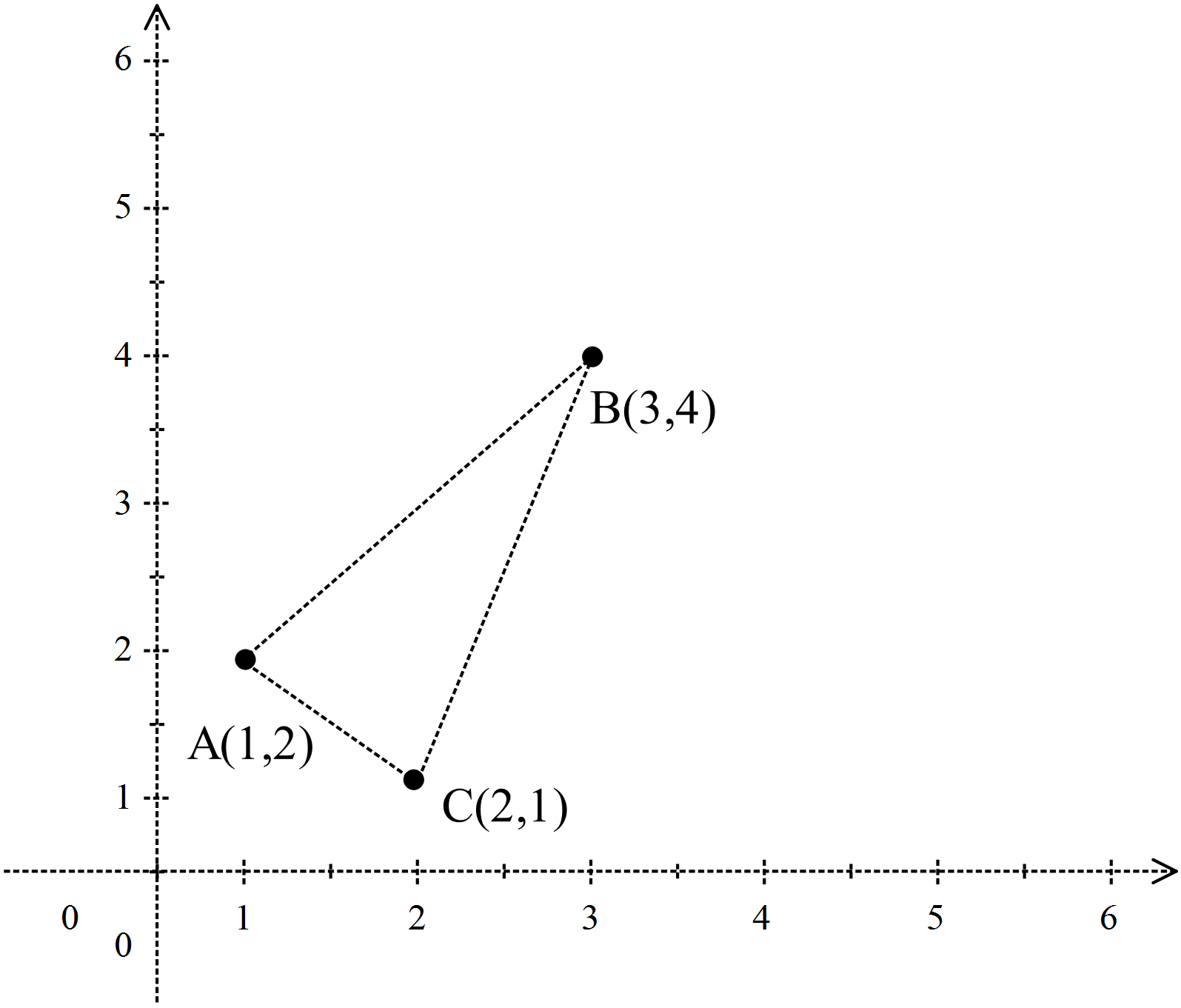
{ //khoi tao window

nhapxy();

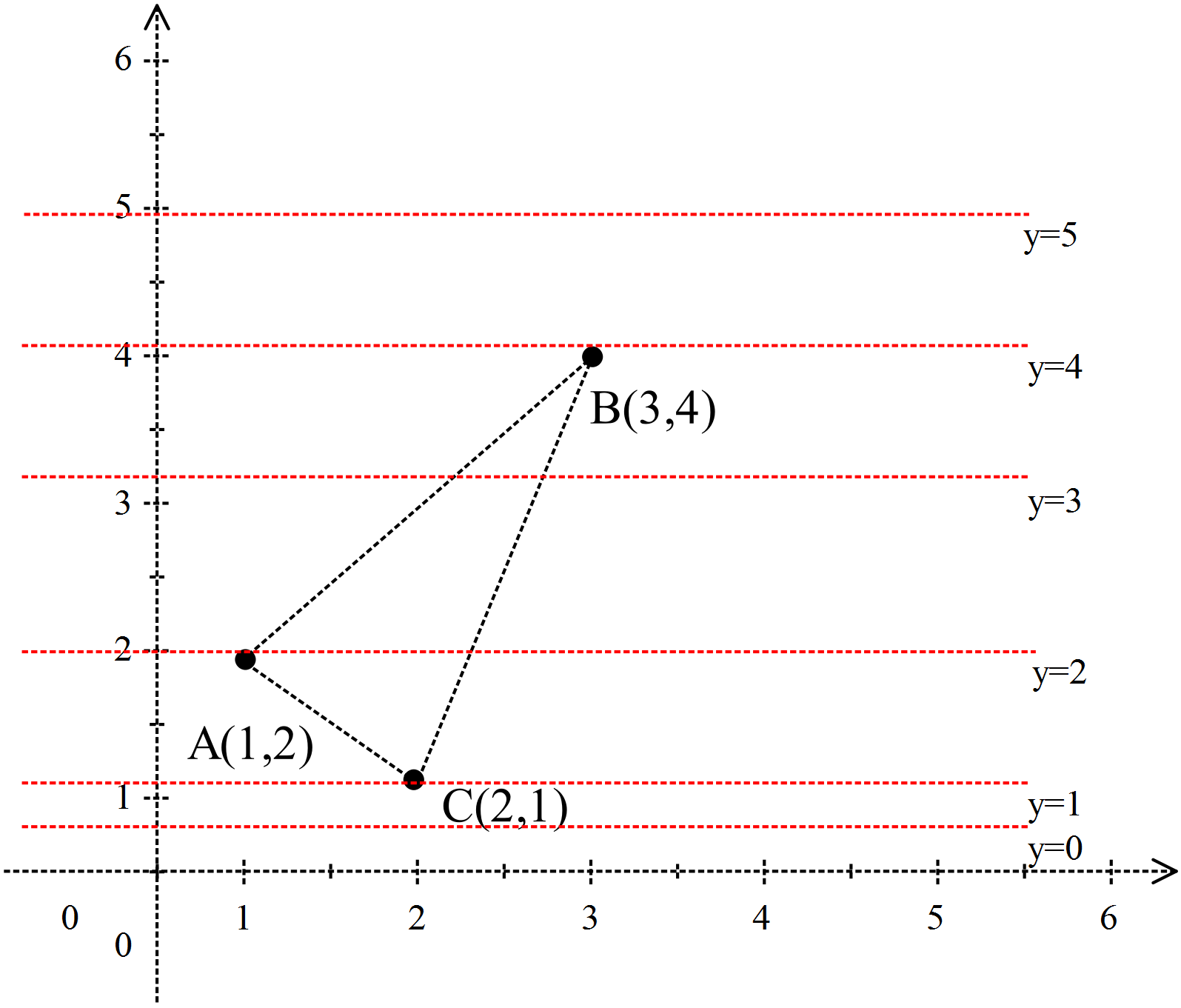
Ellipse();

//closegraph();

getch();

**Tô màu đa giác có tọa độ như sau**

Ma trận các đỉnh của tam giác

Xác định giá trị dx và dy

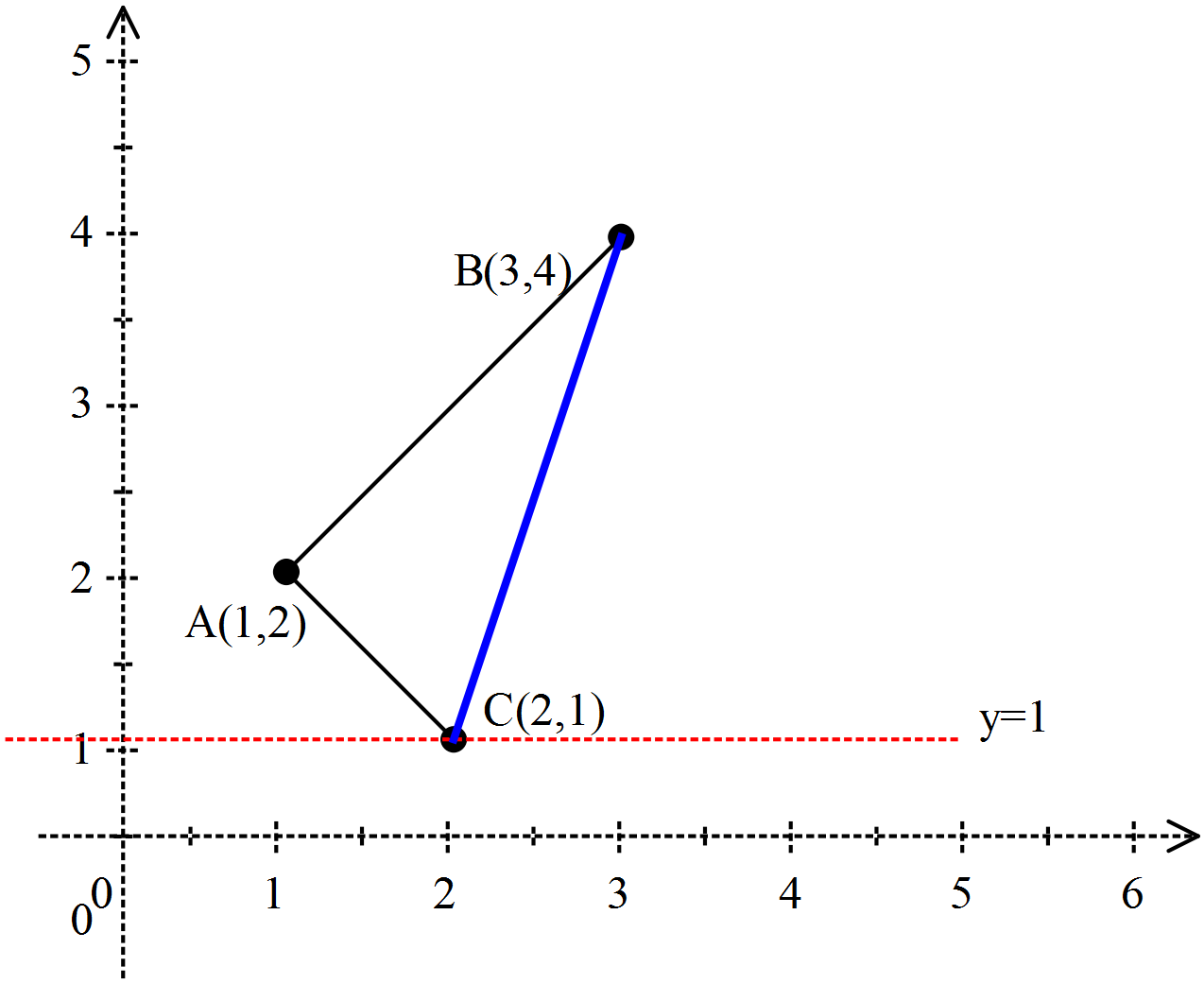
Hệ số góc dy/dx

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***

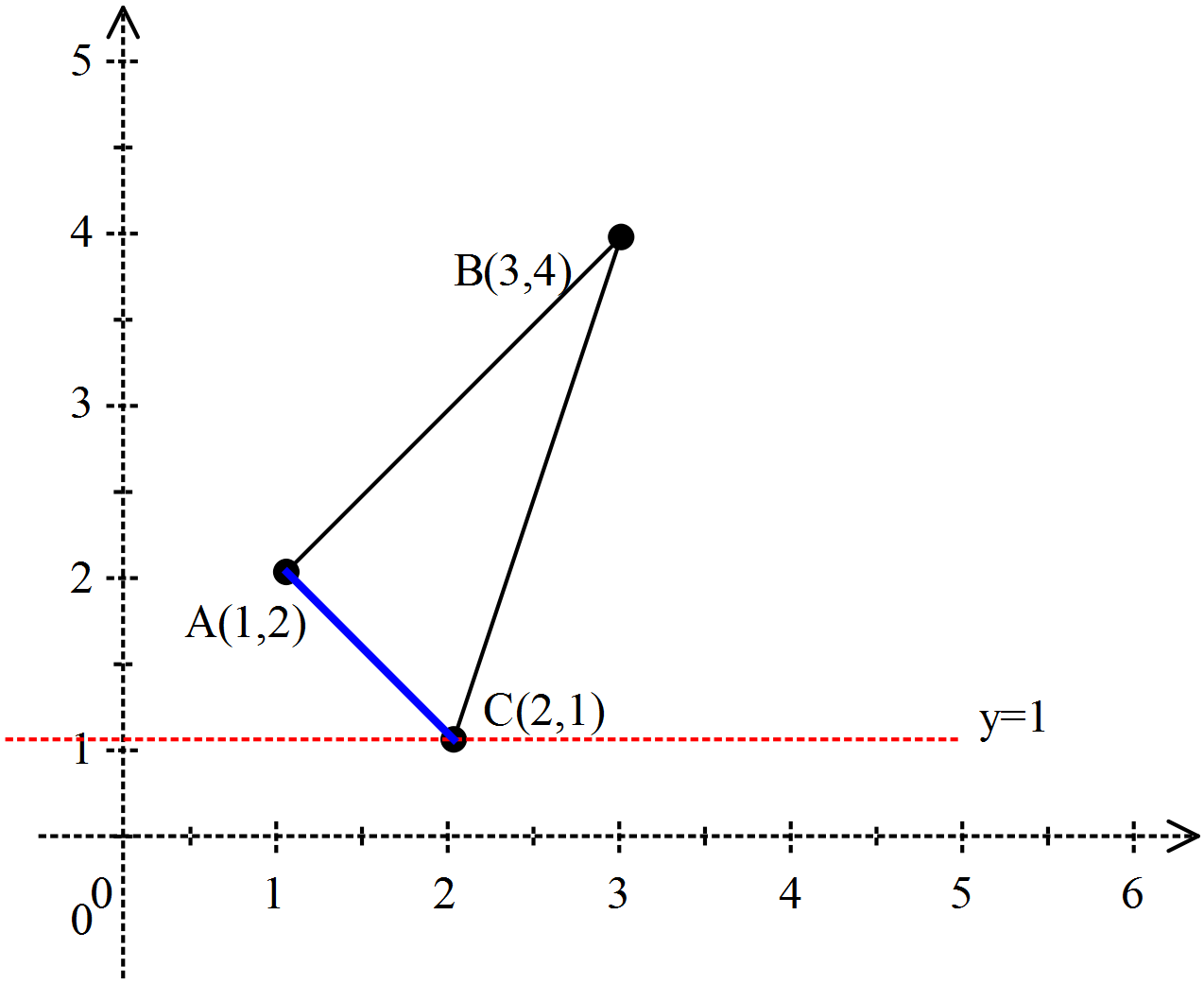
* + Xét cạnh AB, rõ ràng cả 2 đỉnh A và B đều không thỏa mãn biểu thức để chứng tỏ đường thẳng y=1 cắt ngang cạnh AB (. Do đó không cần xác định tọa độ điểm giao nhau giữa y=1 và đoạn AB.
  + Xét cạnh BC

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

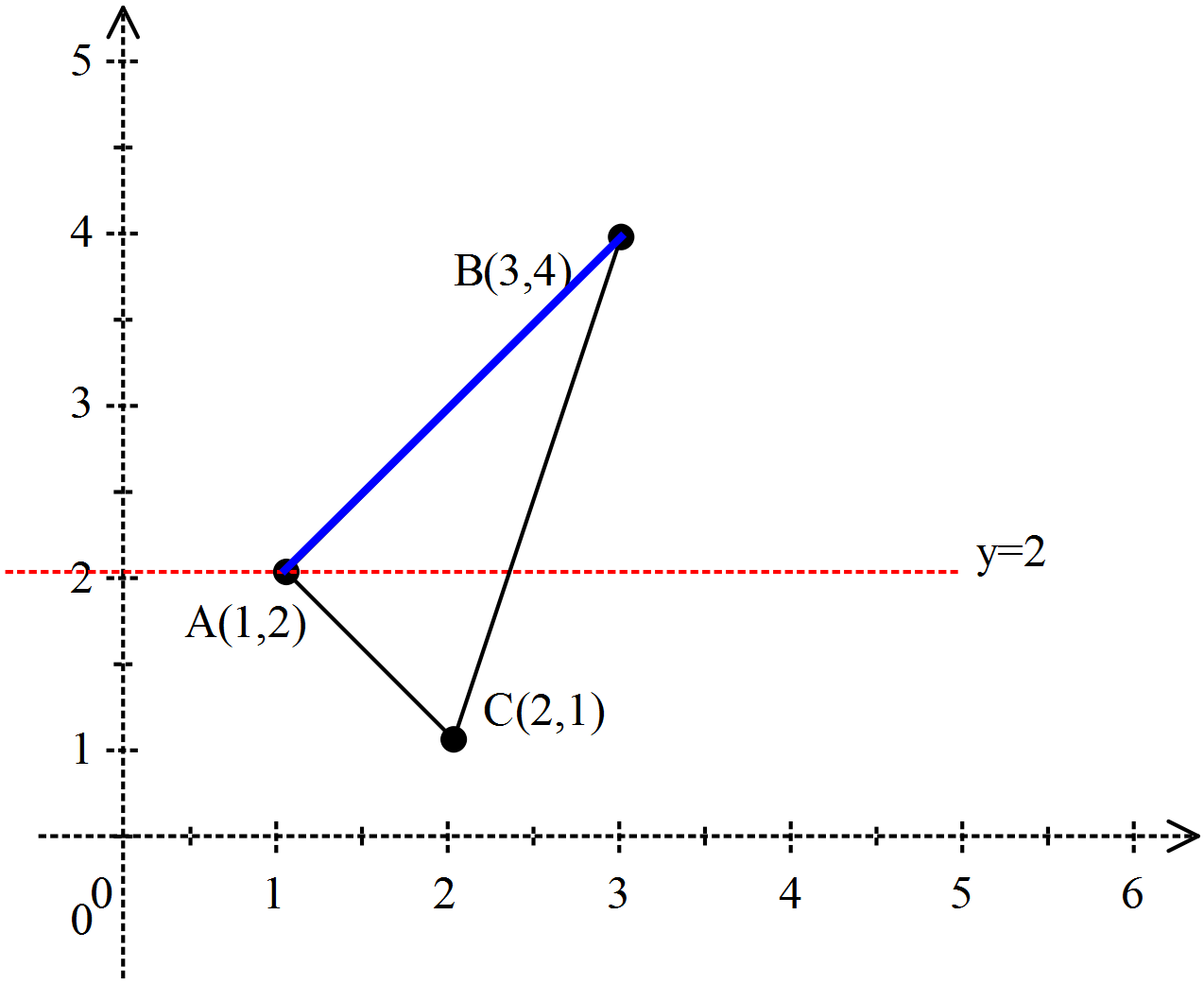
* + Xét cạnh CA

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

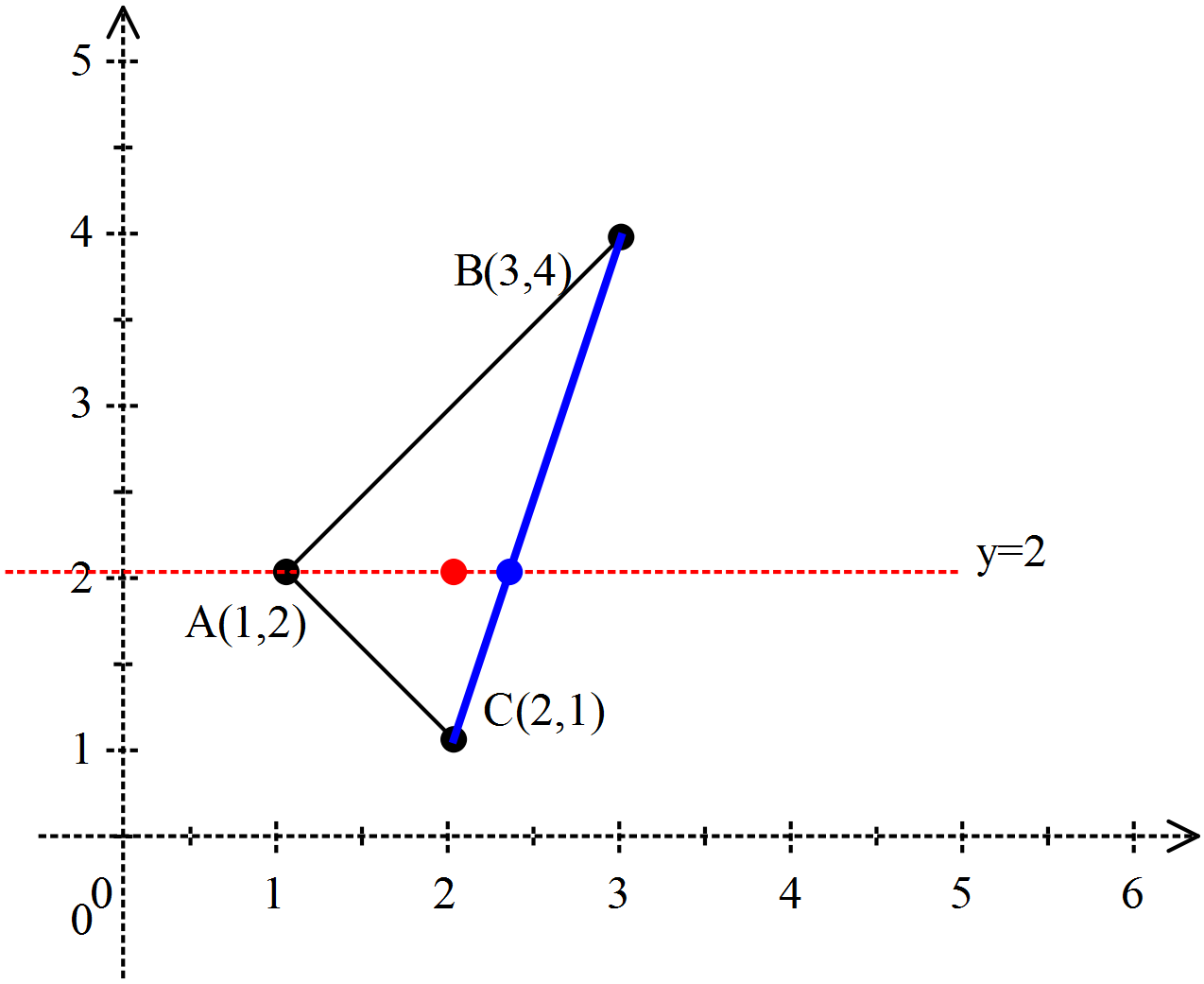
***Y=2***

* + 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

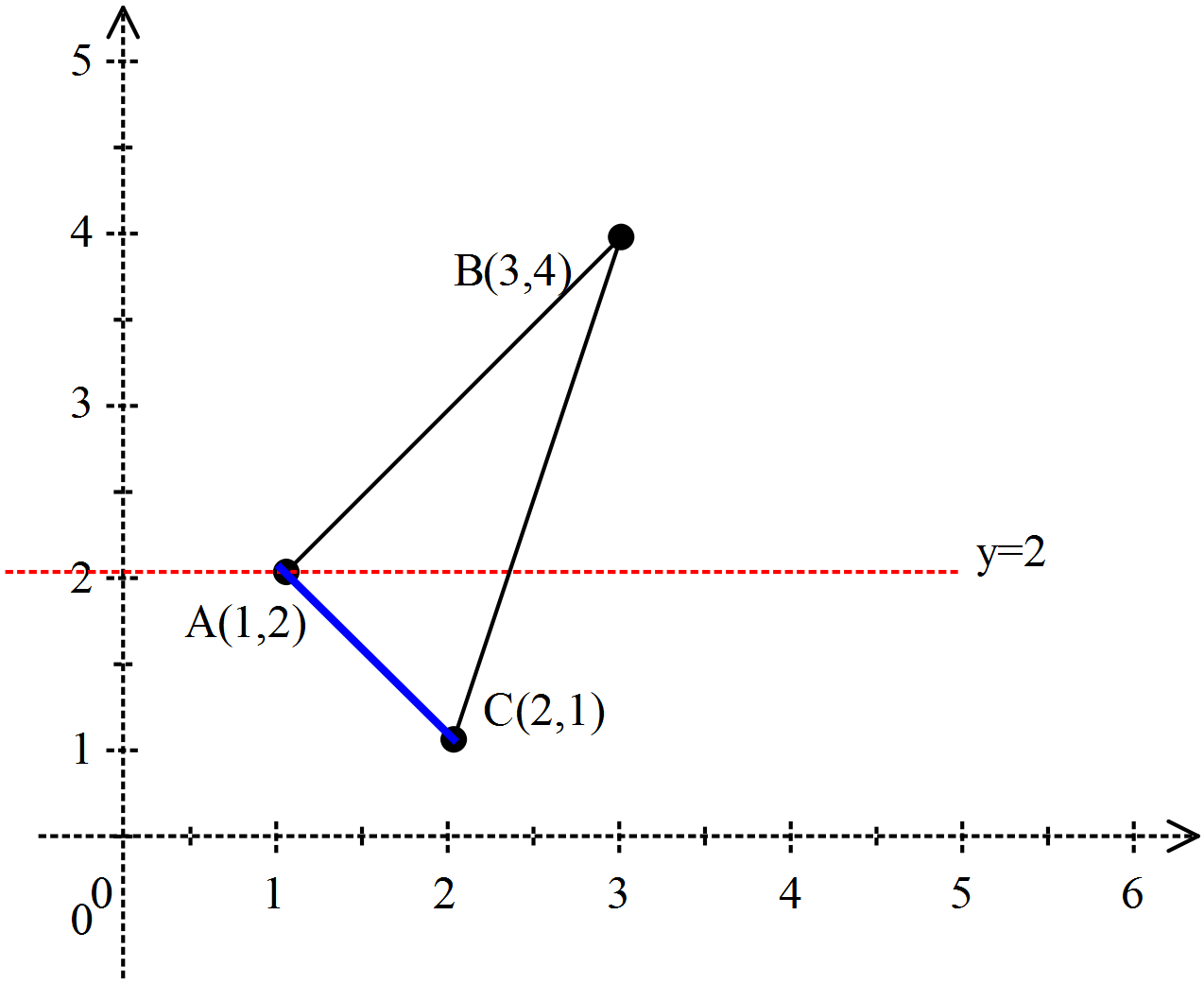
* + Xét cạnh BC

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

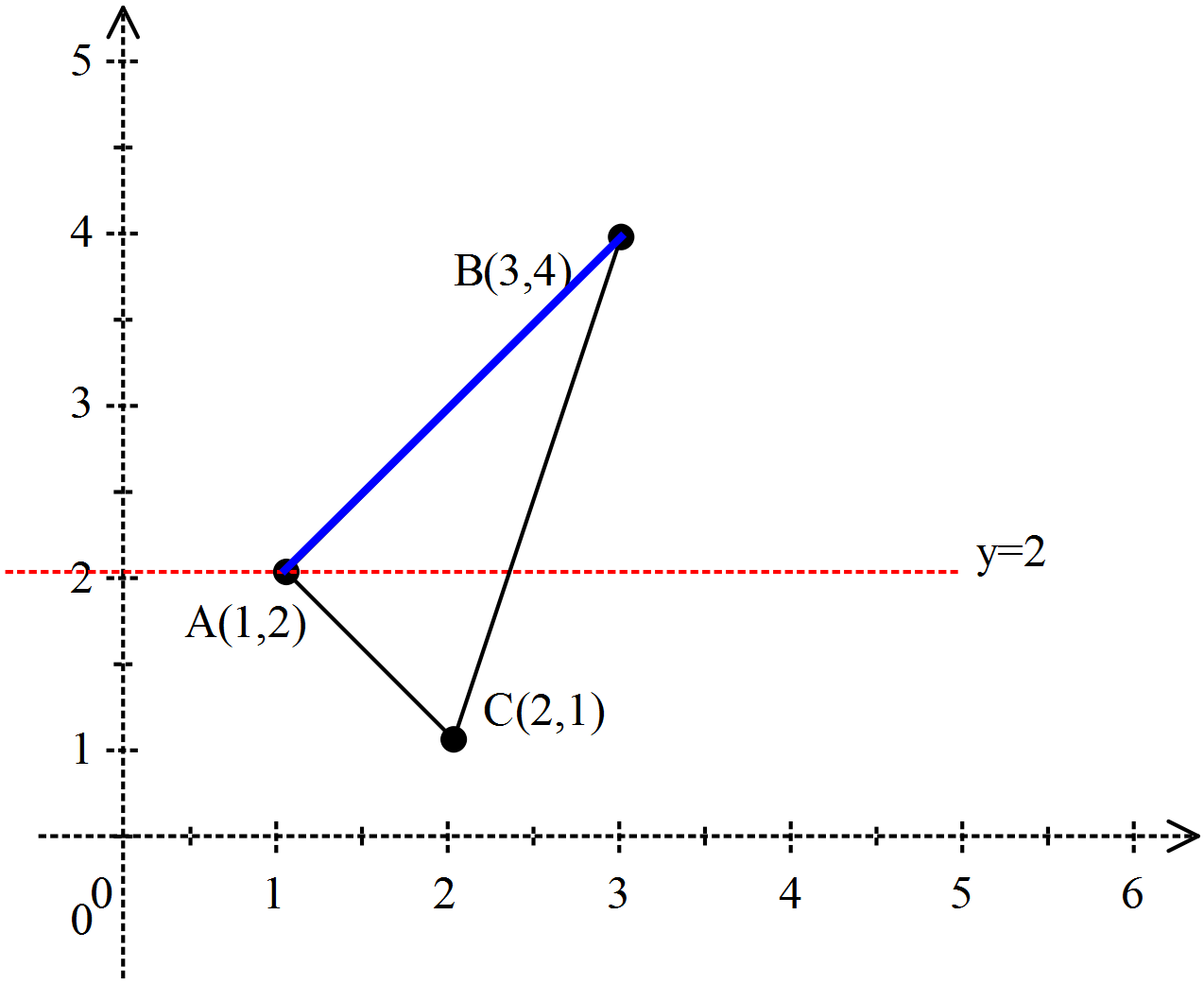
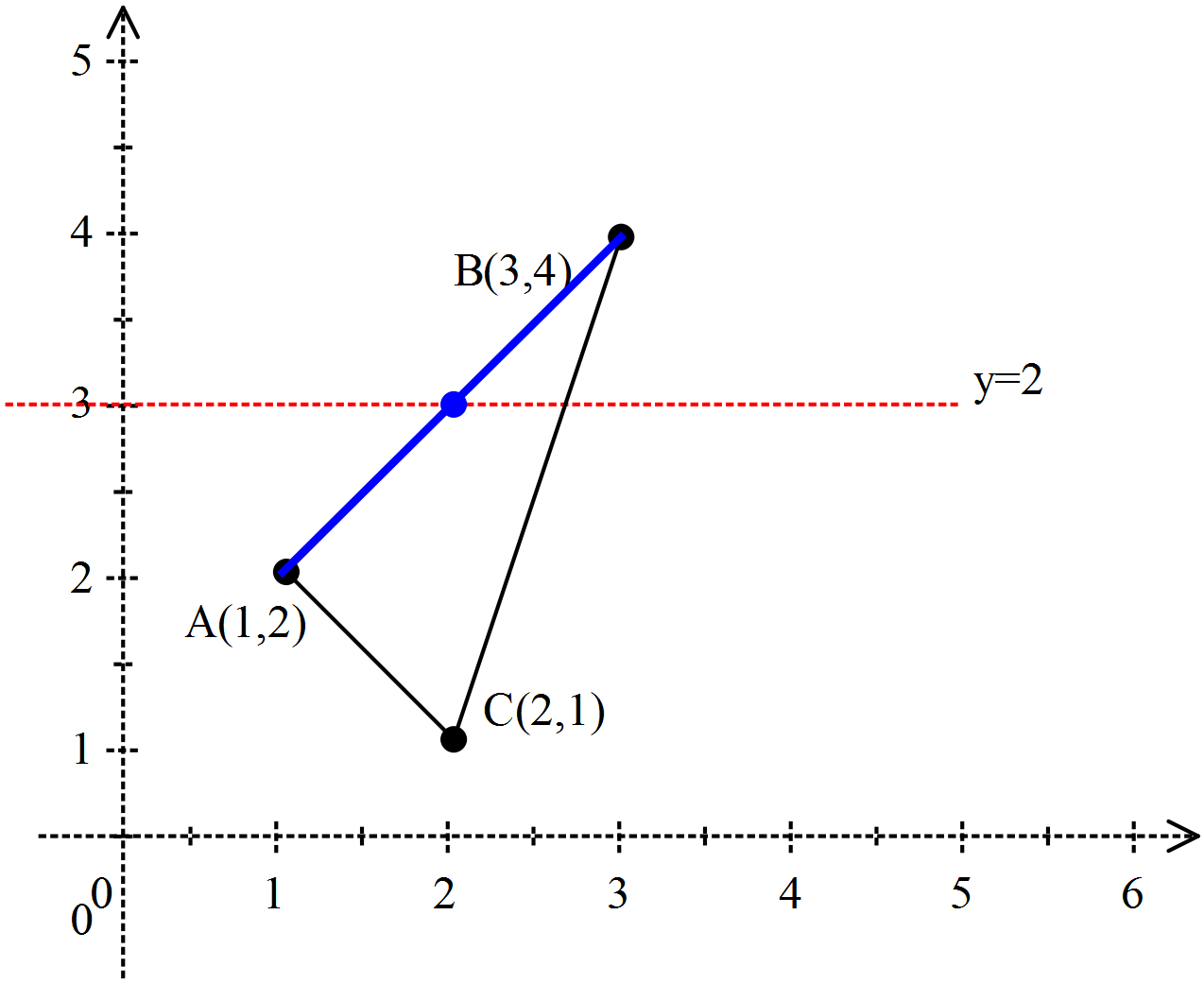
* + Xét cạnh CA

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***

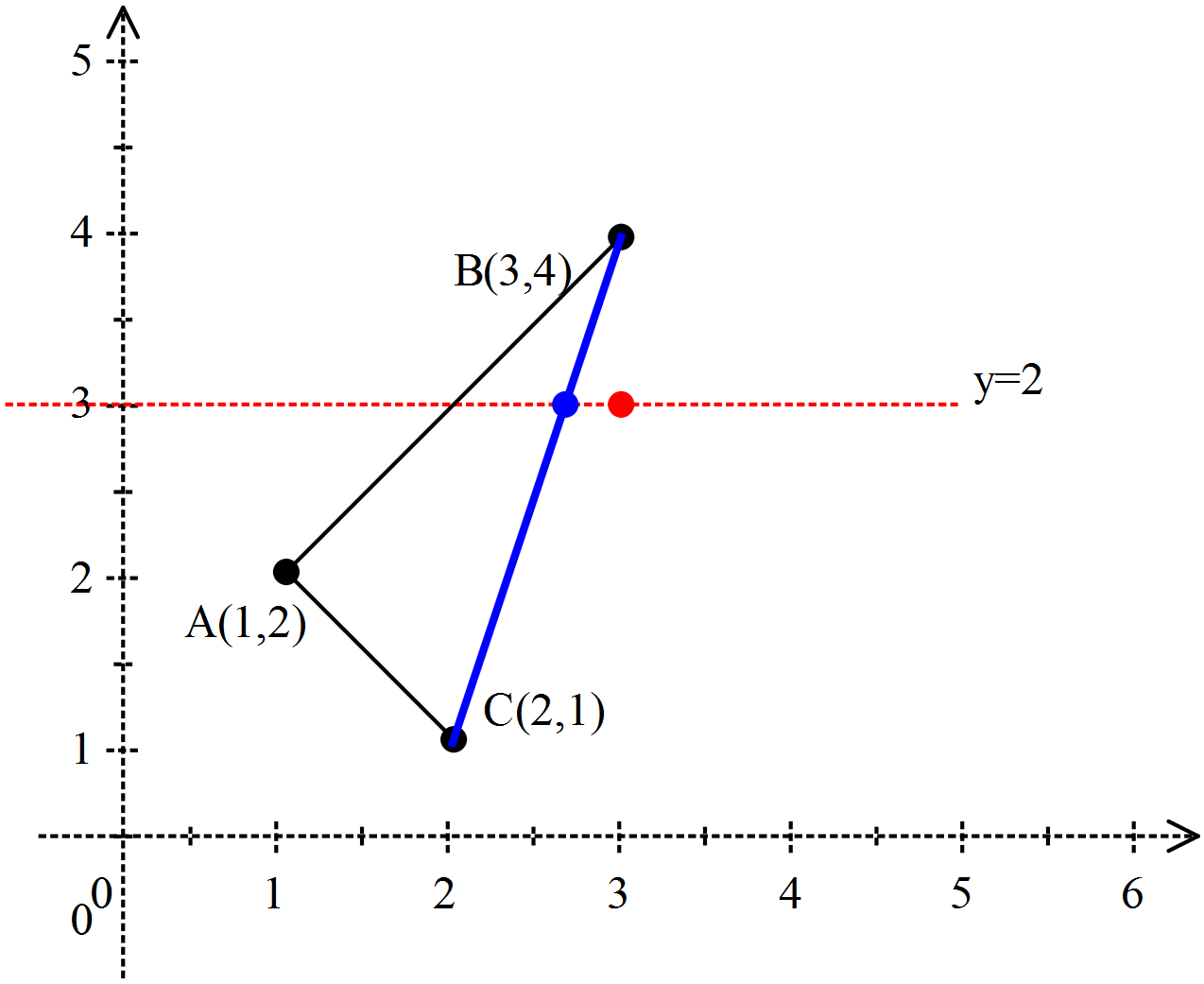
* + Xét cạnh AB

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

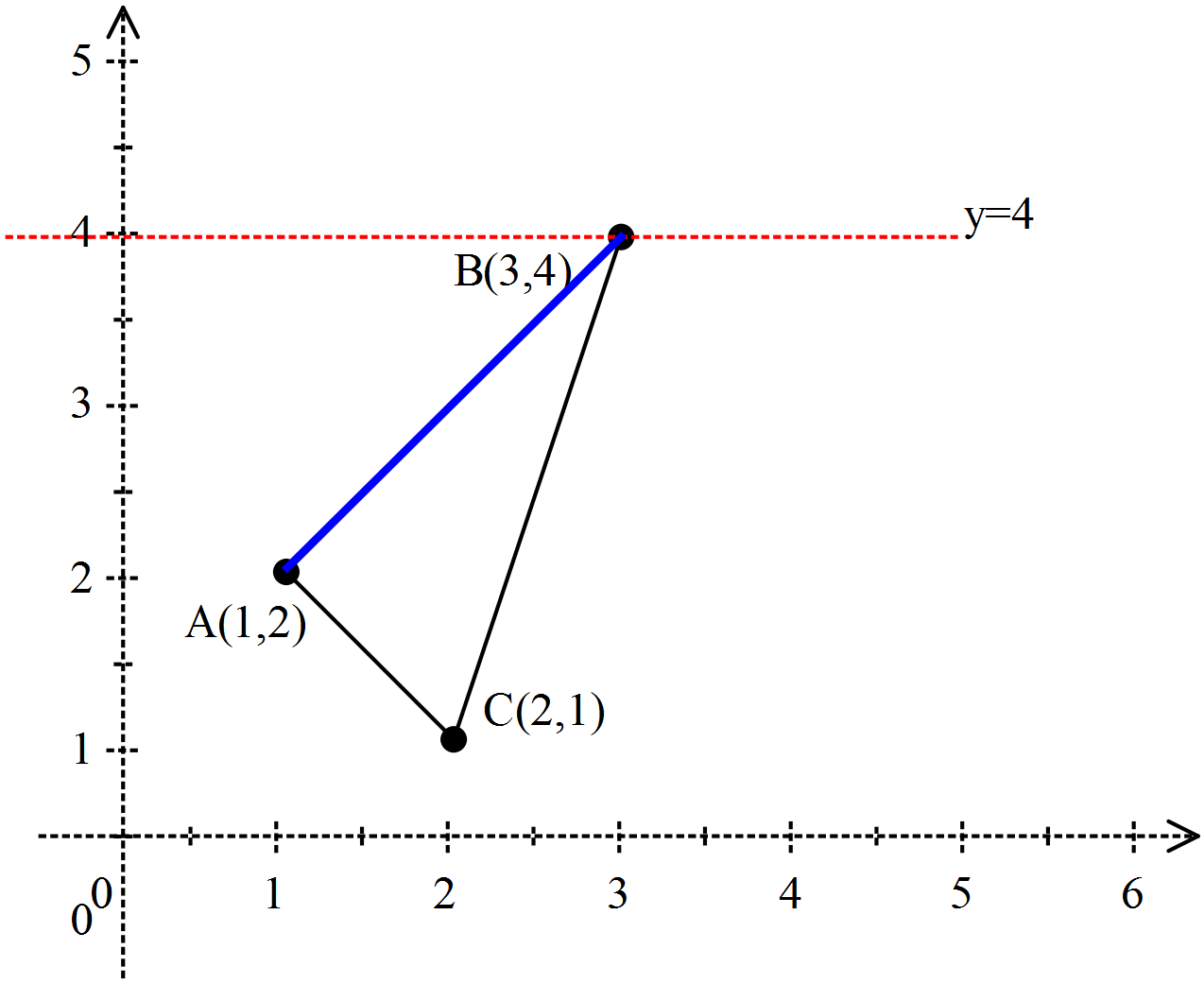
xgđ[3] = xB + m(y-yB) hoặc xgđ[3] = xC + m(y-yC)

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

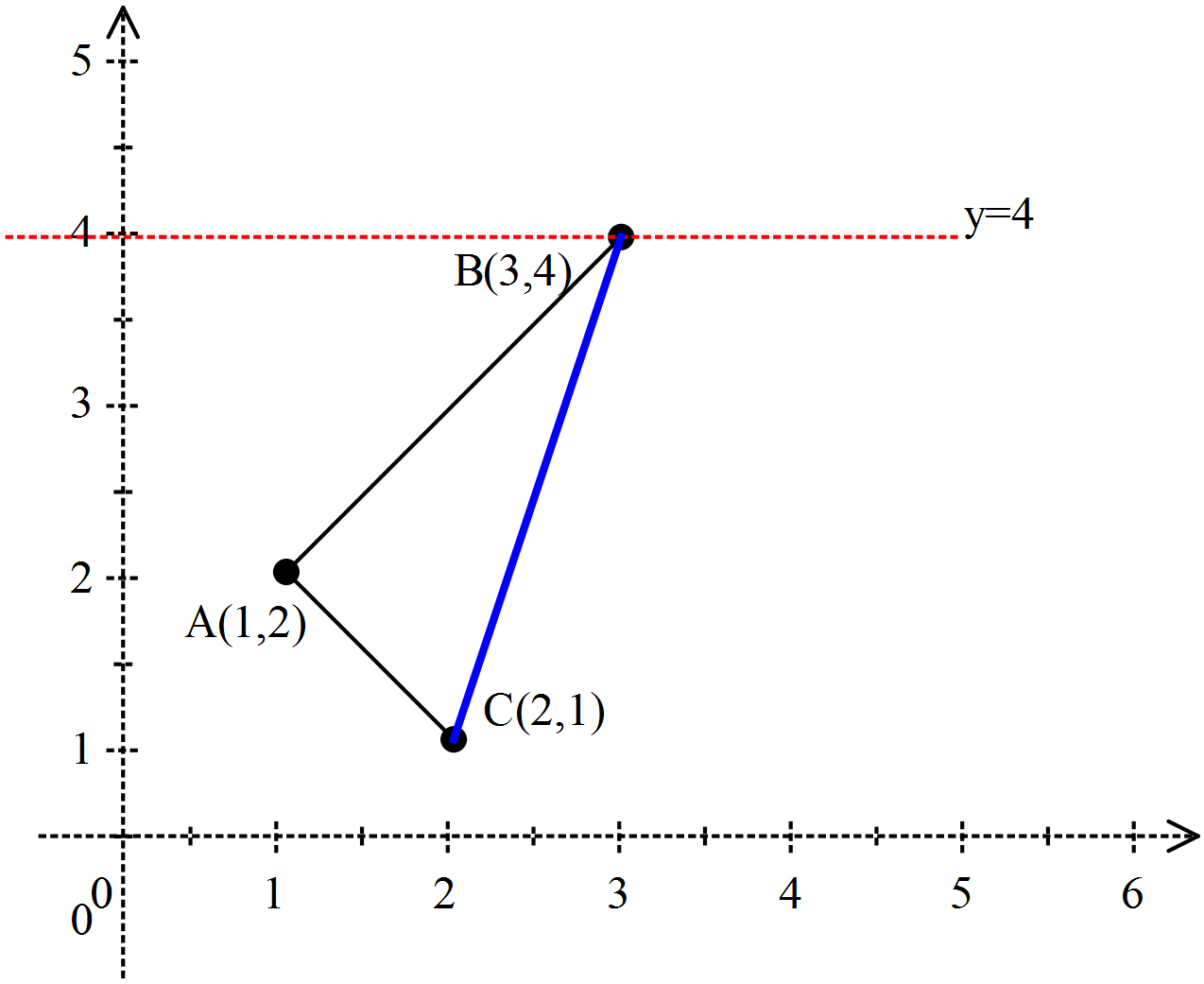
***Y=4***

* + 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

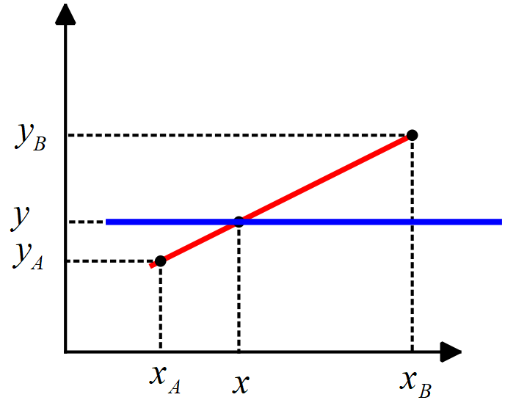
* + Xét cạnh CA

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

**Tìm công thức hoành độ giao điểm:**

1. Ta có tỷ lệ các đoạn thẳng theo công thức sau:

Trong đó, y là giá trị của dòng quét (biết trước)

1. Hay có thể được chứng minh qua công thức sau:

Từ hệ số góc

hoặc

**1.2. Lập trình mô phỏng**

#include<stdio.h>

#include<graphics.h>

#include<math.h>

#define maxdinh 20

int toadodinh[maxdinh][2];

float hesogoc[maxdinh];

int xgd[maxdinh];

int dx;

int dy;

int sodinh;

int ymin;

int yMAX;

void nhaptoado()

{

//nhap so dinh

do

{

printf("Nhap so dinh cua da giac: ");

scanf("%d",&sodinh);

}

while(sodinh<3 || sodinh >maxdinh);

//nhap to do lan luot cho cac dinh

int i;

for(i=0;i<sodinh;i++)

{

printf("X[%d] = ",i);

scanf("%d",&toadodinh[i][0]);

printf("Y[%d] = ",i);

scanf("%d",&toadodinh[i][1]);

}

//gan to do dinh cuoi ve dinh dau

toadodinh[sodinh][0]=toadodinh[0][0];

toadodinh[sodinh][1]=toadodinh[0][1];

//xac dinh he so goc cac canh

for(i=0;i<sodinh;i++)

{

dx=toadodinh[i+1][0] - toadodinh[i][0];

dy=toadodinh[i+1][1] - toadodinh[i][1];

if(dx==0)

{

hesogoc[i]=1.0;

}

else

{

if(dy==0)

{

hesogoc[i]=0.0;

}

else

{

hesogoc[i]=(float)dy/dx;

}

}

}

//in ra danh sach cac thong so

printf("So dinh: %d\n",sodinh);

printf("Toa do cac dinh: \n");

for(i=0;i<sodinh;i++)

{

printf("(%d,%d)\t",toadodinh[i][0],toadodinh[i][1]);

}

printf("\nHe so goc cua cac canh la:\n");

for(i=0;i<sodinh;i++)

{

printf("%f\t",hesogoc[i]);

}

}

//Tim min max

void y\_minmax()

{

int i;

//tim ymin

ymin=toadodinh[0][1];

for(i=0;i<sodinh;i++)

{

if(ymin>toadodinh[i][1])

{

ymin=toadodinh[i][1];

}

}

//tim yMAX

yMAX=toadodinh[0][1];

for(i=0;i<sodinh;i++)

{

if(yMAX<toadodinh[i][1])

{

yMAX=toadodinh[i][1];

}

}

//in ra scanline tu ymin den yMAX

printf("\ny\_min = %d \t y\_MAX =%d",ymin,yMAX);

}

void vedagiac()

{

int i;

setcolor(BLUE);

for(i=0;i<sodinh;i++)

{

line(toadodinh[i][0],toadodinh[i][1],toadodinh[i+1][0],toadodinh[i+1][1]);

}

}

void scanline()

{

int y;

int i;

int k;

y=ymin;

while(y<=yMAX)

{

//tim giao diem cac canh

k=0;

for(i=0;i<sodinh;i++)

{

//xet xem dong quet y co cat canh dang xet hay khong?

if((toadodinh[i][1]<y && toadodinh[i+1][1]<y)||(toadodinh[i][1]>y && toadodinh[i+1][1]>y))

{

//khong co giao diem;

}

else

{

//Tim hoanh do giao diem xgd[k]=round(toadodinh[i][0]+(y-toadodinh[i][1])/hesogoc[i]);

k++;

}

}

//sap xep cac hoanh do giao diem va in ra

int j;

int t;

int tam;

for(j=0;j<k-1;j++)

{

for(t=j+1;t<k;t++)

{

if(xgd[j]>xgd[t])

{

//doi cho

tam=xgd[j];

xgd[j]=xgd[t];

xgd[t]=tam;

}

}

}

//in ra cac giao diem sau khi da sap xep

for(j=0;j<k;j++)

{

printf("\n(%d,%d)",xgd[j],y);

}

//Ve duong thang noi cac giao diem

setcolor(RED);

for(j=0;j<k-1;j=j+2)

{

line(xgd[j],y,xgd[j+1],y);

}

//Tang dong quet len 1

y++;

}

}

int main()

{

nhaptoado();

y\_minmax();

initwindow(400,400);

vedagiac();

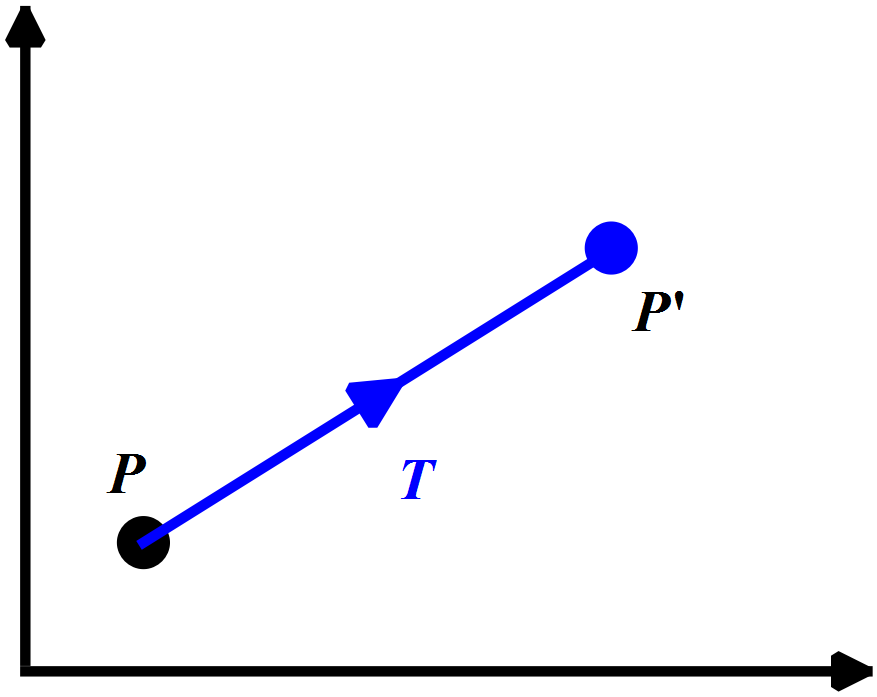
scanline();

getch();

}

**Bài 1.** Thực hiện tịnh tiến đối tượng có tọa độ (0,0) (1,3) (-1,2) với dx=2 và dy=1.5

Ta có: P(x,y) là điểm ban đầu, P’(x’,y’) là điểm tại vị trí mới

x’ = x + tx và y’ = y + ty

Cặp (tx, ty) gọi là vector dịch chuyển

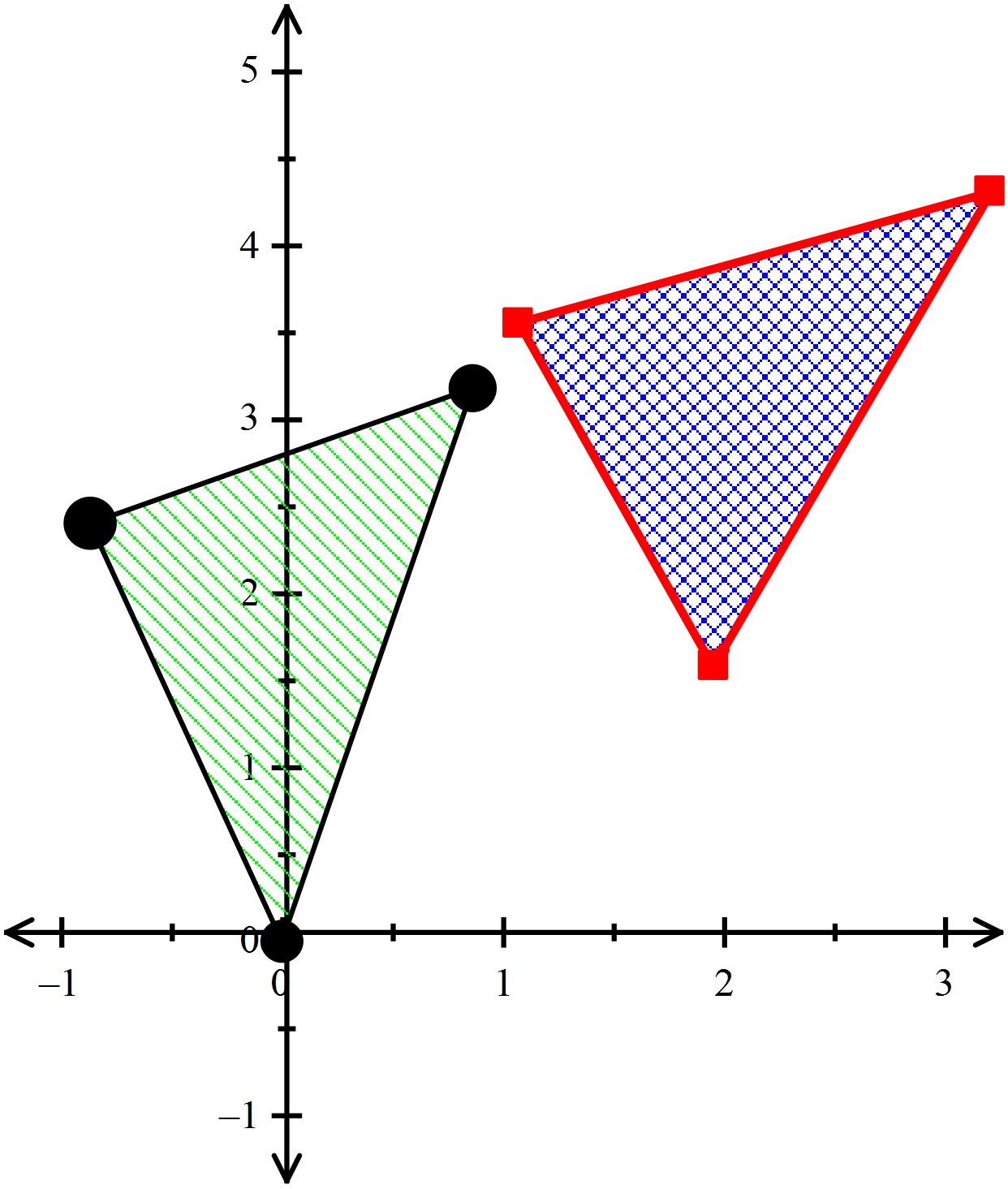
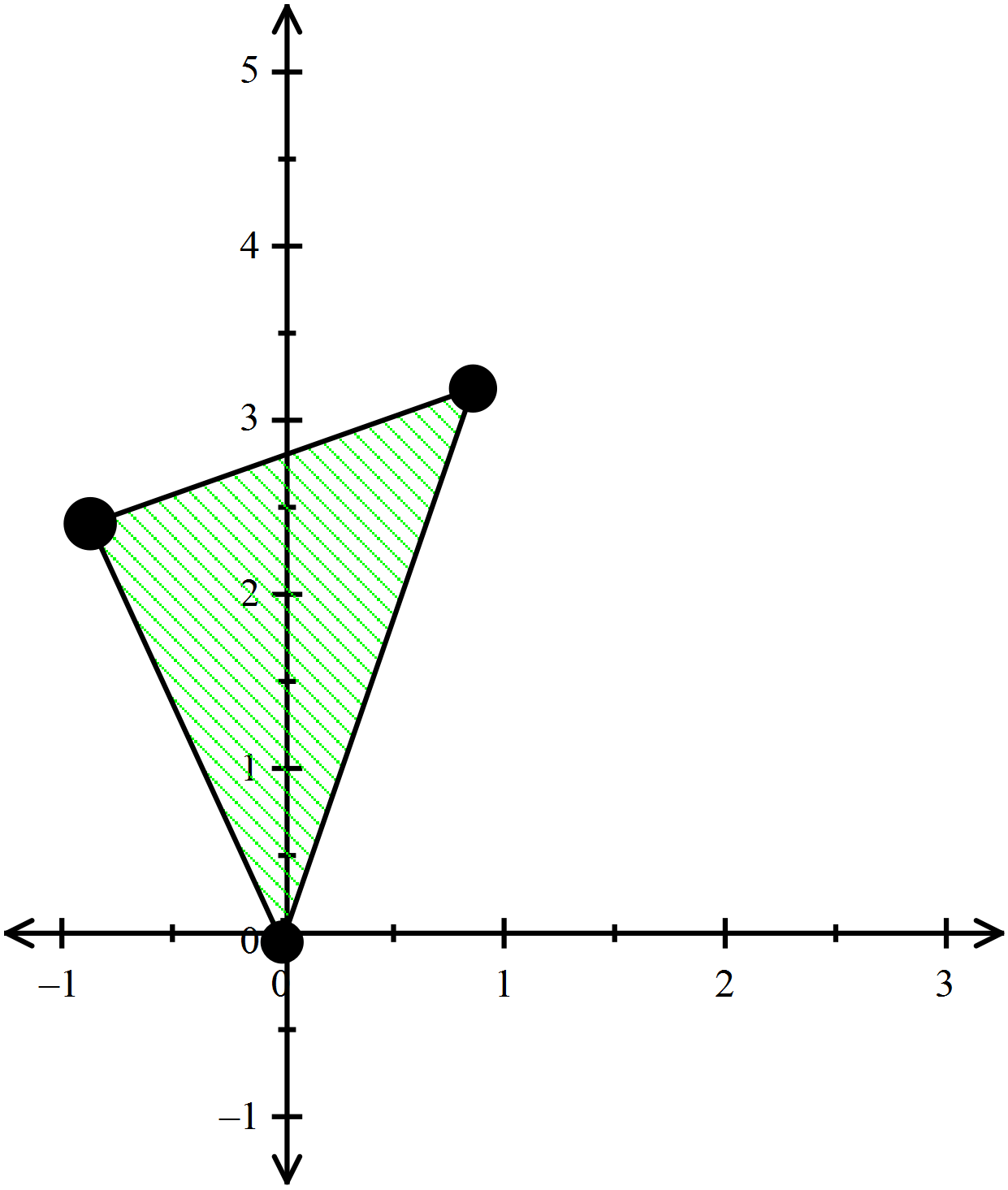
Ta có: P =

P’ =

T =

Từ đó: P’ = P + T

**Áp dụng:**



Ta có phương trình biến đổi:

P’ = P + T Với T =

Hay

Ta có, tại tọa độ (0,0)

Ta có, tại tọa độ (1,3)

Ta có, tại tọa độ (-1,2)

**1.2. Lập trình mô phỏng**

#include <graphics.h>

//khai bao bien

int xshift;

int yshift;

int n;

int xc[100];

int yc[100];

//chuong trinh con

void draw\_object()

{

int i;

for(i=0;i<n;i++)

{

line(xc[i],yc[i],xc[(i+1)%n],yc[(i+1)%n]);

delay(1000);

}

}

//nhap thong so

void nhapdulieu()

{

int i;

printf("Nhap so canh cua da giac = ");

scanf("%d",&n);

for (i=0;i<n;i++)

{

printf("Toa do x cho canh %d = ",i);

scanf("%d",&xc[i]);

printf("Toa do y cho canh %d = ",i);

scanf("%d",&yc[i]);

}

printf("Khoang cach dich chuyen theo truc x = ");

scanf("%d",&xshift);

printf("Khoang cach dich chuyen theo truc y = ");

scanf("%d",&yshift);

}

//dich chuyen doi tuong

void translate()

{

int i;

for(i=0;i<n;i++)

{

xc[i]=xc[i]+xshift;

yc[i]=yc[i]+yshift;

}

}

//chuong trinh chinh

int main()

{

nhapdulieu();

initwindow(800,800);

draw\_object();

delay(1000);

translate();

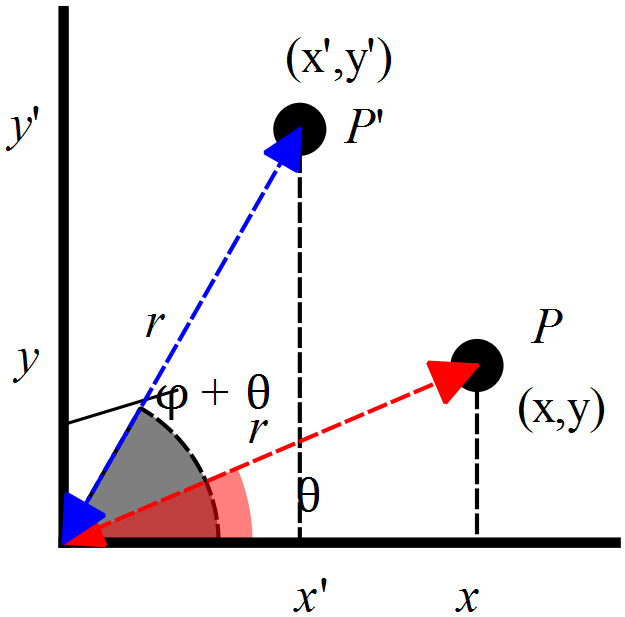
draw\_object();

getch();

return 0;

}

**Bài 2.** Thực hiện phép quay dương cho đối tượng có tọa độ (0,0) (2,4) (-3,4) với 1 góc

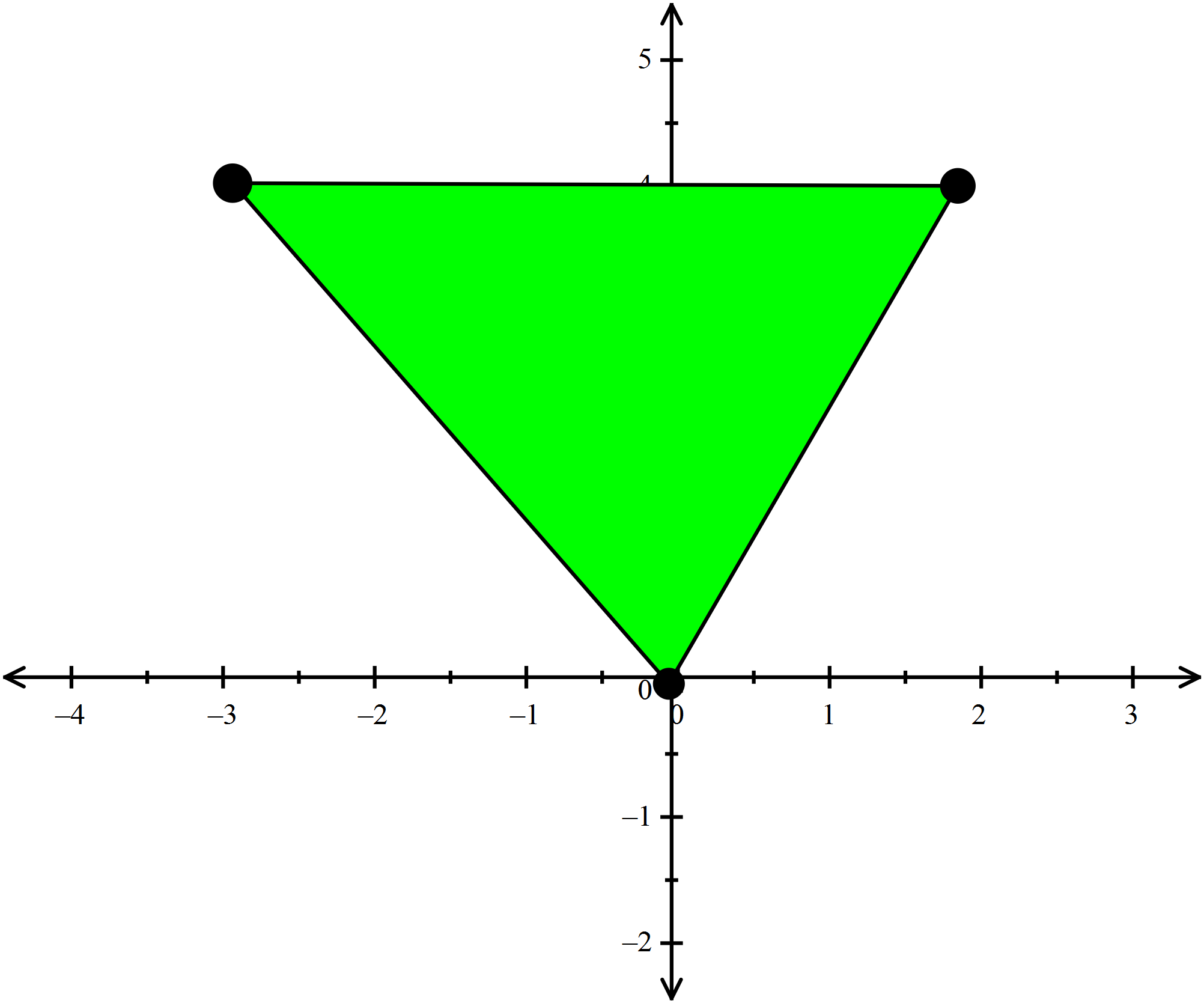
Phương trình biến đổi để xoay điểm P khi tâm ở gốc tọa độ và mối quan với hệ góc quay

Trong đó: góc tại vị trí ban đầu

Các tọa độ ban đầu của điểm trong tọa độ cực là

Thay vào phương trình trước ta có:

Từ ma trận biểu diễn P và P’: P = P’ =

**Ta có: P’ = P . R. Với R =

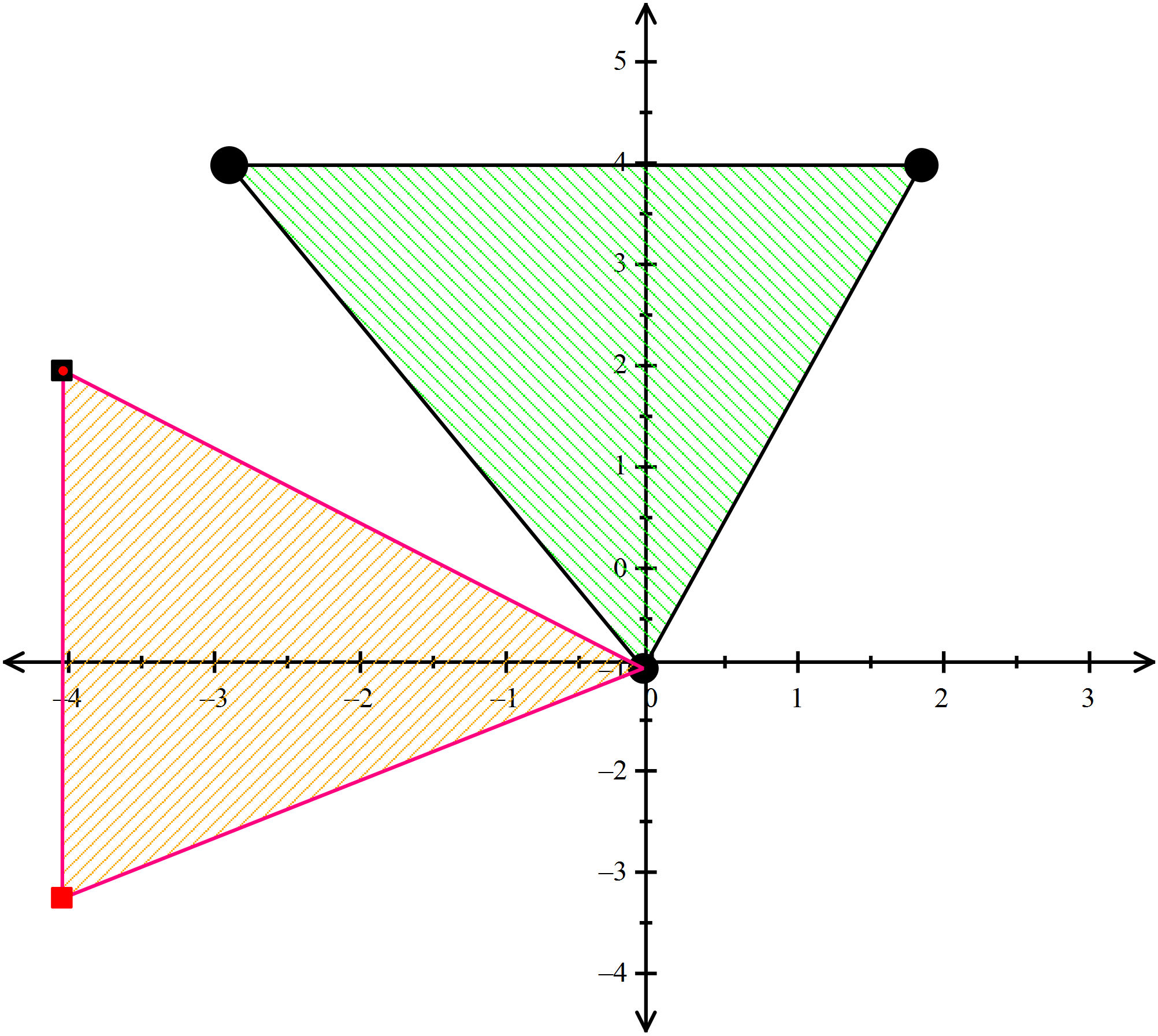
Áp dụng:

Ta có phương trình biến đổi:

P’ = P . R. Với R =

(sin)=1; (cos)=0

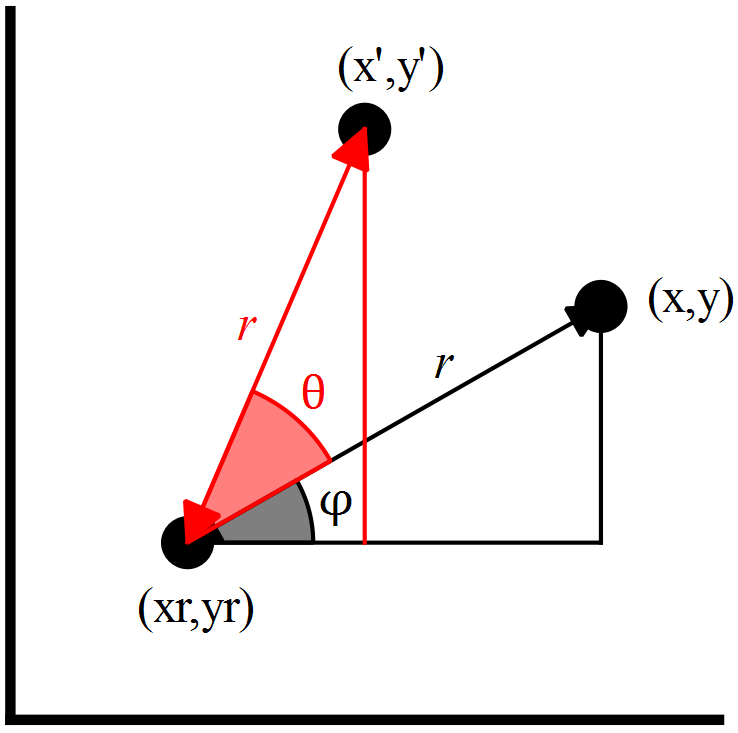
Hay

**Ta có, tại tọa độ (0,0)

Ta có, tại tọa độ (2,4)

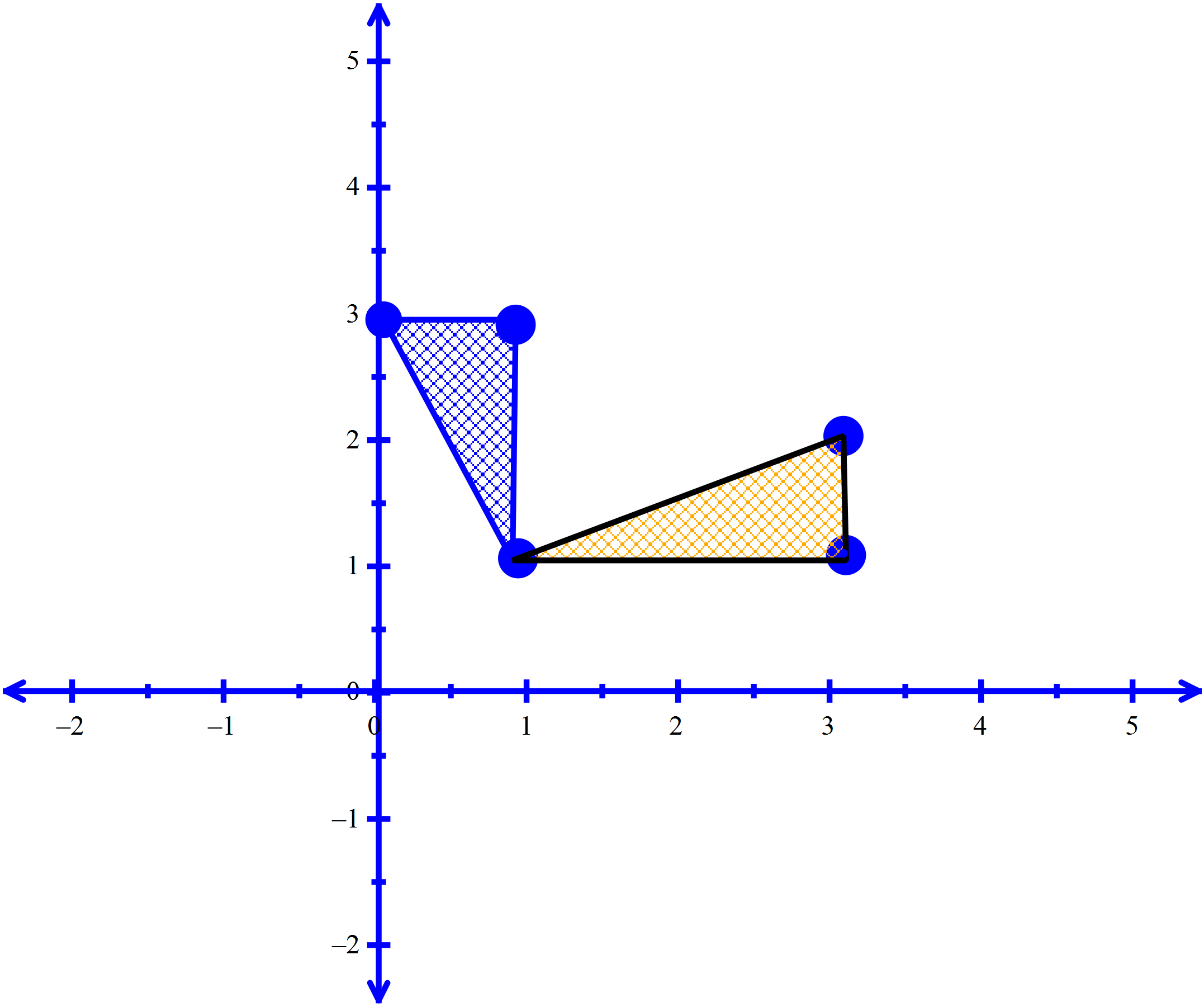
Ta có, tại tọa độ (-3,4)

* *Góc quay dương ngược chiều kim đồng hồ và ngược lại*

**Bài 3.** Thực hiện phép quay dương cho đối tượng có tọa độ (1,1) (3,1) 3,2) tại tâm có tọa độ (1,1) với 1 góc

Quay 1 điểm với tâm tùy ý (khác với gốc tọa độ)

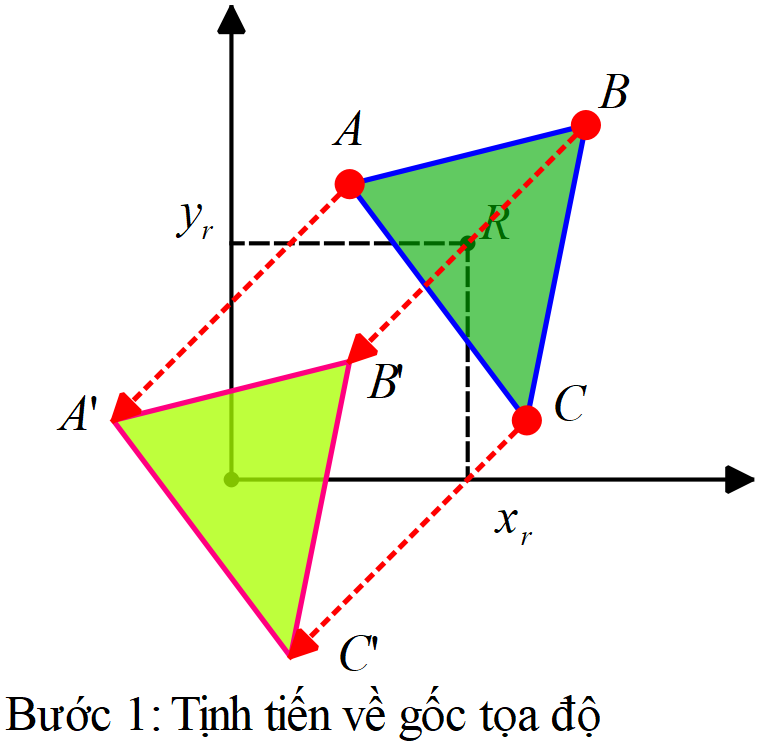
Áp dụng công thức trên ta có:

*Ta có, tại tọa độ (1,1)*

*Ta có, tại tọa độ (3,1)*

*Ta có, tại tọa độ (3,2)*

*0*



**Chứng minh biểu thức:**

Bước 1. Tịnh tiến đối tượng về gốc tọa độ từ vị trí (*xr,yr*)

Hay

Bước 2. Quay tại gốc tọa độ với góc quay

Từ phương trình quay đối tượng: và

Ta thế: và vào phương trình quay đối tượng, ta có:

Bước 3. Tịnh tiến đối tượng về vị trí tâm (xr, yr) ban đầu:

Từ pt: , ta lấy x’ và y’ cộng với gia trị lần lượt xr và yr

**1.2. Lập trình mô phỏng**

#include <stdio.h>

#include <graphics.h>

#include <math.h>

//khai bao bien

int xr;

int yr;

int n;

int xc[100];

int yc[100];

int degree;

float radian;

//chuong trinh con

void draw\_object()

{

int i;

for(i=0;i<n;i++)

{

line(xc[i],yc[i],xc[(i+1)%n],yc[(i+1)%n]);

delay(1000);

}

}

//nhap thong so

void nhapdulieu()

{

int i;

printf("Nhap so canh cua da giac = ");

scanf("%d",&n);

for (i=0;i<n;i++)

{

printf("Toa do x cho canh %d = ",i);

scanf("%d",&xc[i]);

printf("Toa do y cho canh %d = ",i);

scanf("%d",&yc[i]);

}

printf("Toa do x cua doi tuong tai vi tri moi = ");

scanf("%d",&xr);

printf("Toa do y cua doi tuong tai vi tri moi y = ");

scanf("%d",&yr);

printf("He so goc quay (dgree) = ");

scanf("%d",&degree);

}

void convert\_degree\_2\_radian()

{

radian=(float) degree/180\*3.14;

}

//dich chuyen doi tuong

void rotation()

{

int i;

float dx;

float dy;

for(i=0;i<n;i++)

{

dx=float(xc[i]-xr);

dy=float(yc[i]-yr);

xc[i]=floor(xr+dx\*cos(radian)-dy\*sin(radian)); yc[i]=floor(yr+dx\*sin(radian)+dy\*cos(radian));

}

}

//ve hinh vuong quay

void five\_square()

{

int i;

float dx;

float dy;

int num\_square;

for(num\_square=0; num\_square<5; num\_square++)

{

for(i=0;i<n;i++)

{

dx=float(xc[i]-xr);

dy=float(yc[i]-yr);

xc[i]=floor(xr+dx\*cos(radian)-dy\*sin(radian));

yc[i]=floor(yr+dx\*sin(radian)+dy\*cos(radian));

}

draw\_object();

}

}

//chuong trinh chinh

int main()

{

nhapdulieu();

convert\_degree\_2\_radian();

initwindow(800,800);

draw\_object();

delay(1000);

rotation();

draw\_object();

//five\_square();

getch();

}

**Bài tập 4.** Thực hiện phép biến đổi tỷ lệ cho đối tượng có tọa độ (-2,1) (-1,4) (1,2) với hệ số S(2,3)

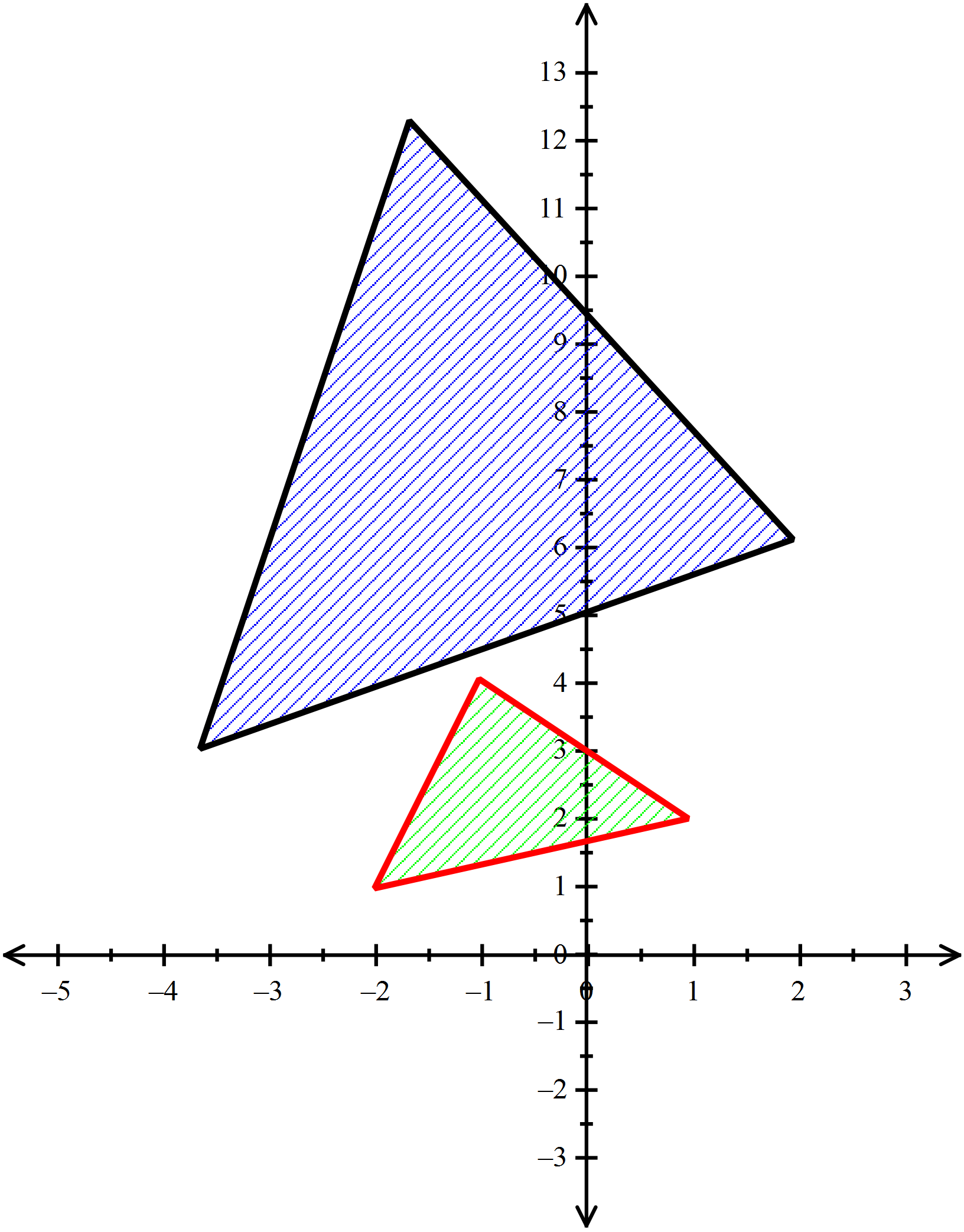
Một phép biến đổi tỷ lệ làm thay đổi kích thước của một đối tượng. Thao tác này có thể được thực hiện cho đa giác bằng cách nhân các giá trị tọa độ (x, y) của từng đỉnh với các hệ số tỷ lệ sx và sy, để tạo ra tọa độ biến đổi (x', y’): x’=x.sx y’=y.sy

hay P’=S.P

Trong đó sx, sy là các số nguyên dương. Nếu giá trị của sx và sy bé hơn 1 được phép thu nhỏ, ngược lại là phép phóng to.

Nếu cả sx và sy đều là 1, đối tượng không có sự biến đổi

Nếu sx và sy có cùng 1 giá trị, phép biến đổi được gọi là phép biến đổi đồng nhất

**Áp dụng:**

Ta có phương trình biến đổi:

Với sx = 2 và sy = 3

Hay

Ta có, tại tọa độ (-2,1)

Ta có, tại tọa độ (-1,4)

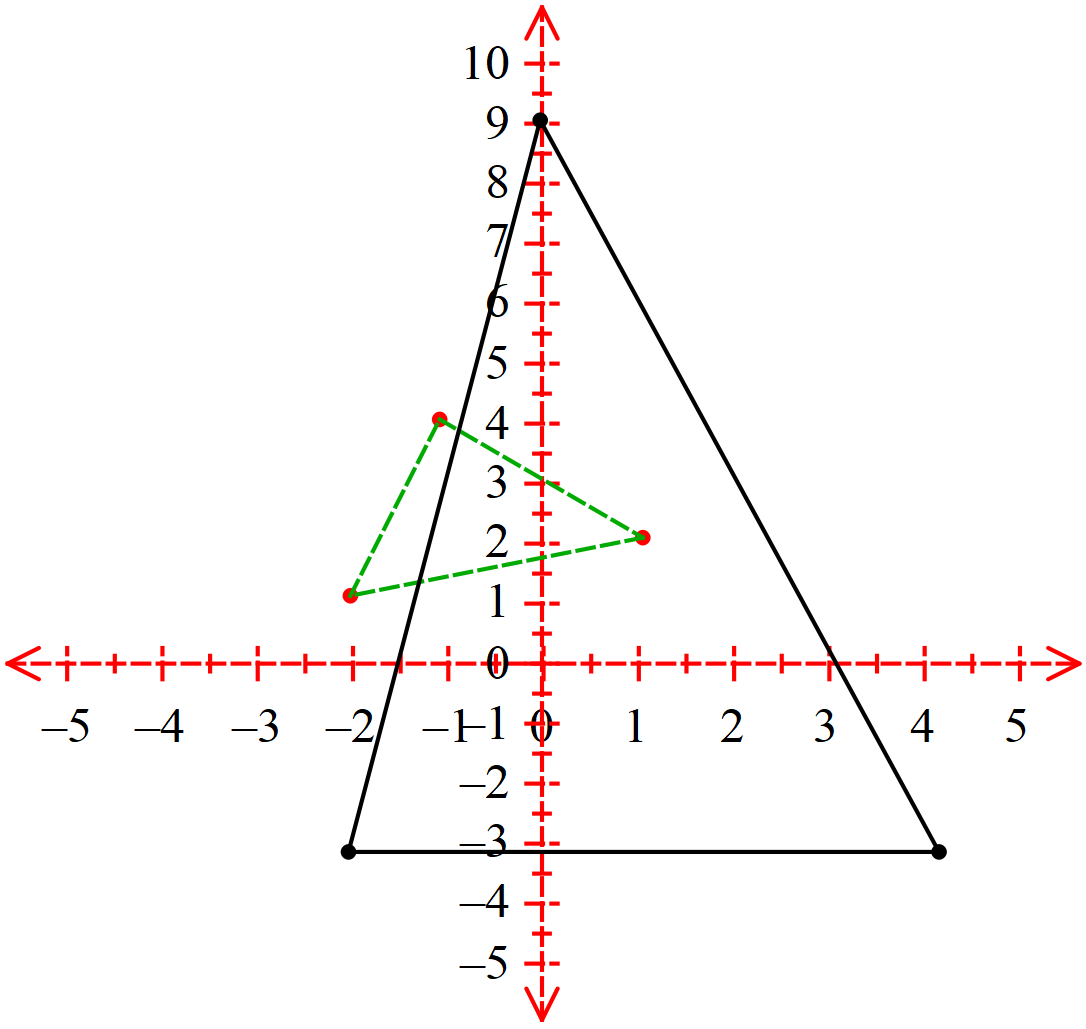
Ta có, tại tọa độ (1,2)

**Bài tập 5.** Thực hiện phép biến đổi tỷ lệ cho đối tượng có tọa độ (-2,1) (-1,4) (1,2) với hệ số S(2,3) tại tâm có tọa độ (1,3).

Ta có thể kiểm soát vị trí của một đối tượng được chia tỷ lệ bằng cách chọn một vị trí, được gọi là điểm cố định, nghĩa là không thay đổi sau khi chuyển đổi tỷ lệ. Điểm cố định (xf,yf) có thể được chọn làm một trong các đỉnh, tâm đối tượng hoặc bất kỳ vị trí nào khác. Một đa giác sau khi được chia tỷ lệ so với điểm cố định bằng cách chia tỷ lệ khoảng cách từ mỗi đỉnh đến điểm cố định.

x’ = xf + (x - xf).sx y’ = yf + (y - xf).sy

**hay x’ = x.sx + (1 - sx).xf y’ = y.sy + (1 - sy).yf**

Áp dụng:

Ta có, tại tọa độ (-2,1)

Ta có, tại tọa độ (-1,4)

Ta có, tại tọa độ (1,2)

**1.2. Lập trình mô phỏng**

#include <graphics.h>

#include <math.h>

//khai bao bien

int n;

int xc[100];

int yc[100];

int xs;

int ys;

int degree;

float scalex;

float scaley;

//chuong trinh con

void draw\_object()

{

int i;

for(i=0;i<n;i++)

{

line(xc[i],yc[i],xc[(i+1)%n],yc[(i+1)%n]);

delay(1000);

}

}

//nhap thong so

void nhapdulieu()

{

int i;

printf("Nhap so canh cua da giac = ");

scanf("%d",&n);

for (i=0;i<n;i++)

{

printf("Toa do x cho canh %d = ",i);

scanf("%d",&xc[i]);

printf("Toa do y cho canh %d = ",i);

scanf("%d",&yc[i]);

}

printf("Ty le bien dang theo truc x= ");

scanf("%f",&scalex);

printf("Ty le bien dang theo truc y= ");

scanf("%f",&scaley);

printf("Toa do moi cua doi tuong tren truc x = ");

scanf("%d",&xs);

printf("Toa do moi cua doi tuong tren truc y = ");

scanf("%d",&ys);

}

//bien doi ty le

void scaling()

{

int i;

int dx;

int dy;

for(i=0;i<n;i++)

{

//chon tam cua doi tuong la diem x0,y0

xc[i]=xs+(int)((float)(xc[i]-xs)\*scalex);

yc[i]=ys+(int)((float)(yc[i]-ys)\*scaley);

printf("%d,%d\n",xc[i],yc[i]);

}

}

//bien doi chong cheo

void fivesquare()

{

int i;

int num\_square;

int dx;

int dy;

for(num\_square=0; num\_square<5; num\_square++)

{

for(i=0;i<n;i++)

{

//chon tam cua doi tuong la diem xs,ys

xc[i]=xs+(int)((float)(xc[i]-xs)\*scalex);

yc[i]=ys+(int)((float)(yc[i]-ys)\*scaley);

printf("%d,%d\n",xc[i],yc[i]);

}

draw\_object();

}

}

//chuong trinh chinh

int main()

{

nhapdulieu();

initwindow(800,800);

draw\_object();

delay(1000);

//scaling();

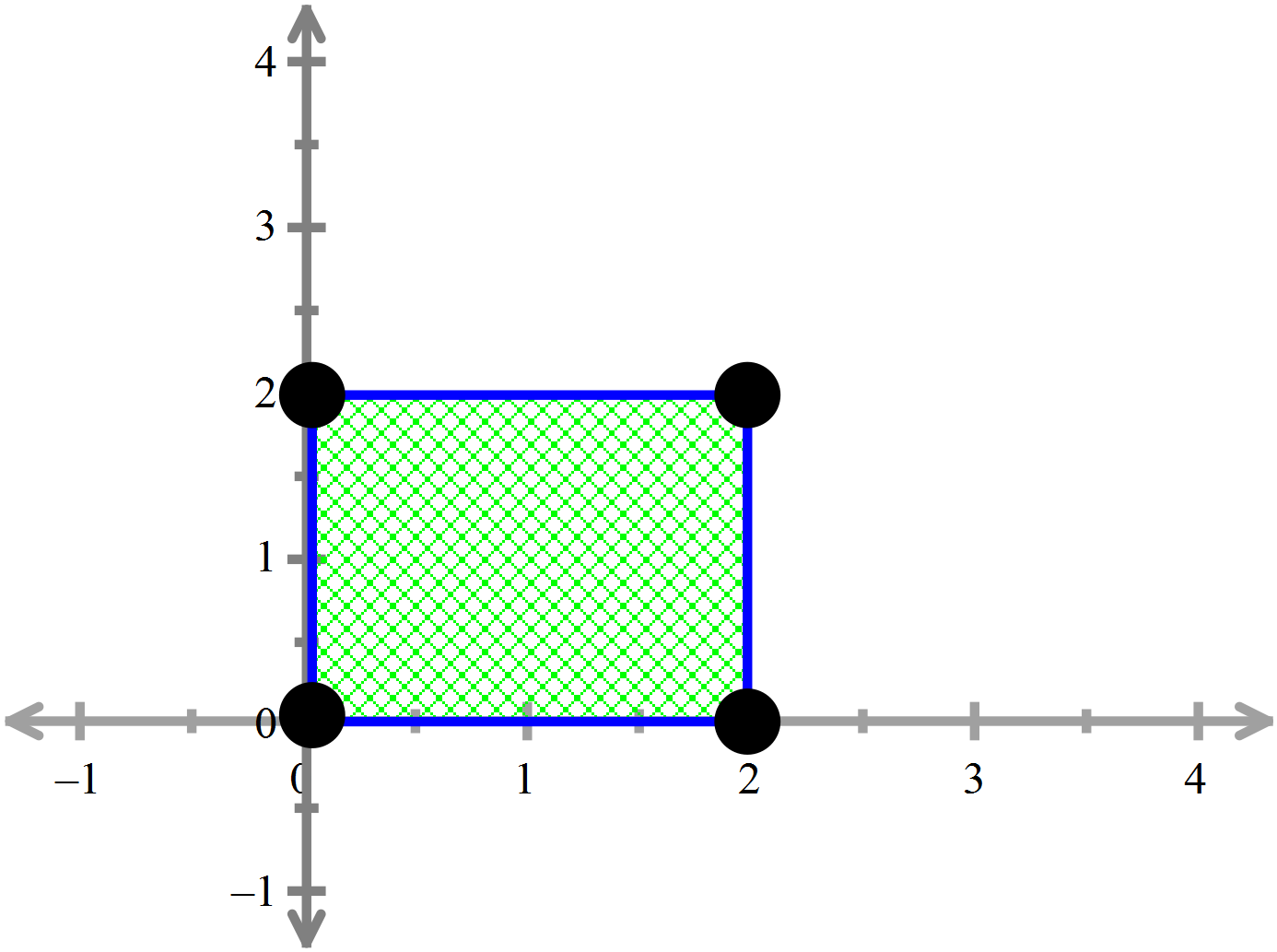
//draw\_object();

fivesquare();

getch();

return 0;

}

**Bài tập 6.** Thực hiện phép biến dạng cho hình sau (0,0) (2,0) (2,2) (0,2) theo trục x với shx=2, tương tự cho shy=2

Ta có phương trình biến đổi theo x:

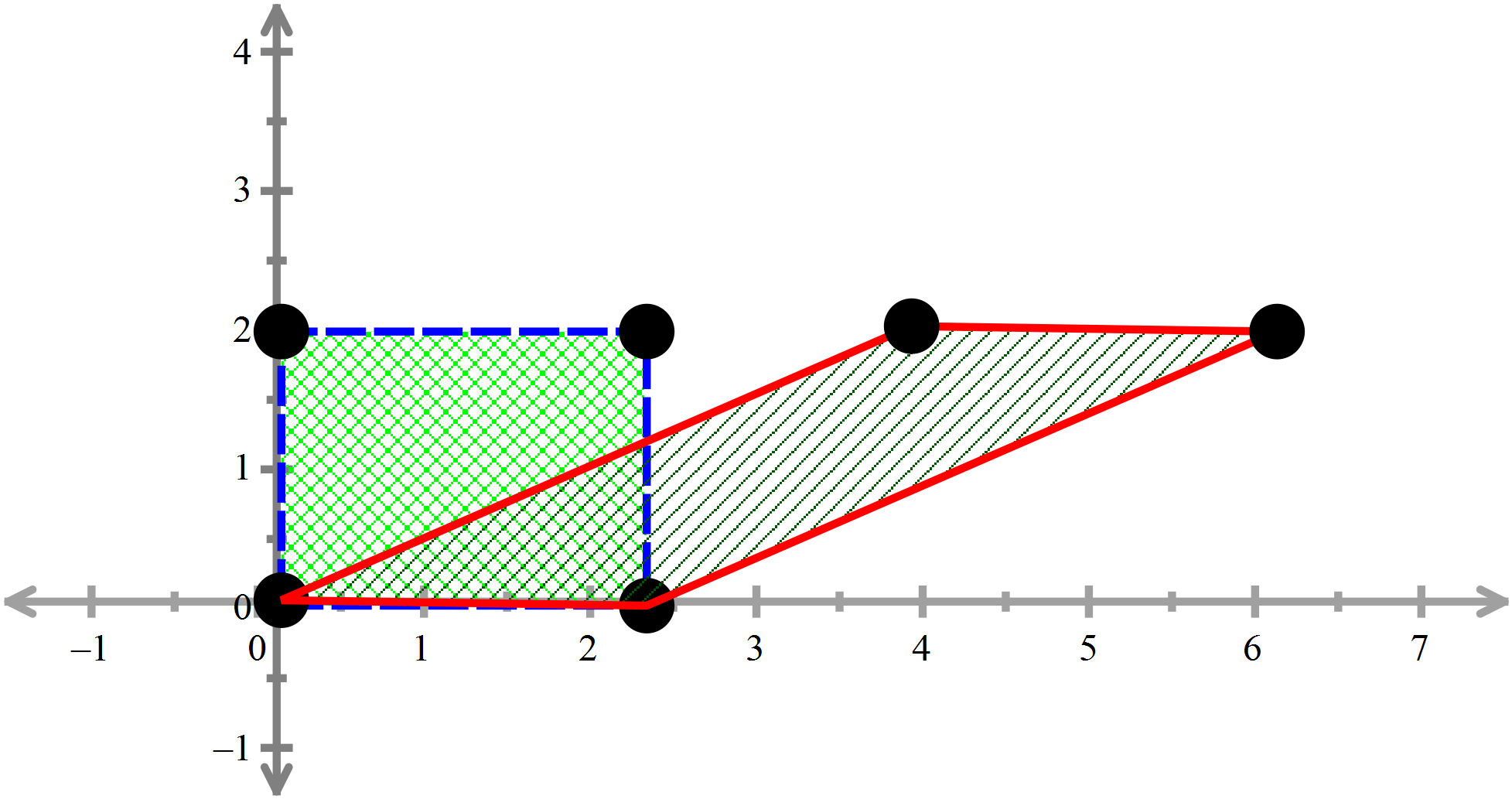
* y’ = y x’ = x+ shx . y

Ta có phương trình biến đổi theo y:

* x’ = x y’ = y+ shy . x

***Áp dụng:***

Thực hiện phép biến dạng cho hình sau (0,0) (2,0) (2,2) (0,2) theo trục x với shx=2

Ta có phương trình biến đổi (0,0):

y’ = 0 x’ = 0+ 2. 0 = 0

Ta có phương trình biến đổi (2,0):

y’ = 0 x’ = 2+ 2. 0 = 2

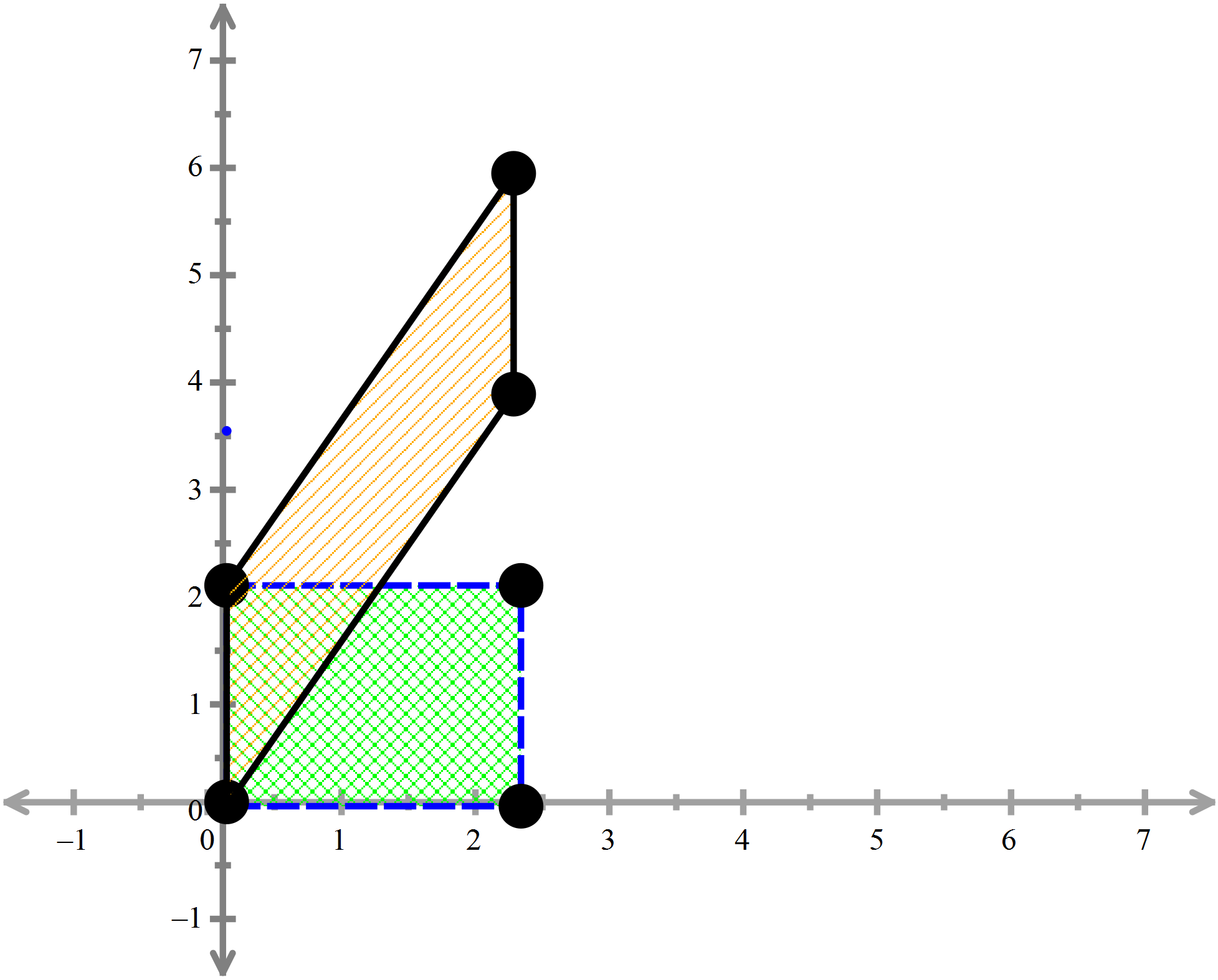
Ta có phương trình biến đổi (2,2):

y’ = 2 x’ = 2+ 2. 2 = 6

Ta có phương trình biến đổi (0,2):

y’ = 2 x’ = 0+ 2. 2 = 4

Thực hiện phép biến dạng cho hình sau (0,0) (2,0) (2,2) (0,2) theo trục y với shy=2

Ta có phương trình biến đổi (0,0):

x’ = 0 y’ = 0+ 2. 0 = 0

Ta có phương trình biến đổi (2,0):

x’ = 2 y’ = 2+ 2. 2 = 4

Ta có phương trình biến đổi (2,2):

x’ = 2 y’ = 2+ 2. 2 = 6

Ta có phương trình biến đổi (0,2):

x’ = 0 y’ = 2+ 2. 0 = 2

**1.2. Lập trình mô phỏng**

#include <graphics.h>

#include <math.h>

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

//khai bao bien

int n;

int xc[100];

int yc[100];

int shx;

int shy;

//chuong trinh con

void draw\_object()

{

int i;

for(i=0;i<n;i++)

{

line(xc[i],yc[i],xc[(i+1)%n],yc[(i+1)%n]);

delay(1000);

}

}

//nhap thong so

void nhapdulieu()

{

int i;

printf("Nhap so canh cua da giac = ");

scanf("%d",&n);

for (i=0;i<n;i++)

{

printf("Toa do x cho canh %d = ",i);

scanf("%d",&xc[i]);

printf("Toa do y cho canh %d = ",i);

scanf("%d",&yc[i]);

}

printf("He so bien dang theo truc x = ");

scanf("%d",&shx);

printf("He so bien dang theo truc y = ");

scanf("%d",&shy);

}

//Bien dang theo truc X

void shearingX()

{

int i;

for(i=0;i<n;i++)

{

xc[i]=xc[i]+shx\*yc[i];

printf("(%d,%d)\t",xc[i],yc[i]);

}

}

//Bien dang theo truc Y

void shearingY()

{

int i;

for(i=0;i<n;i++)

{

yc[i]=yc[i]+shy\*xc[i];

printf("(%d,%d)\t",xc[i],yc[i]);

}

}

//chuong trinh chinh

int main()

{

nhapdulieu();

initwindow(800,800);

draw\_object();

delay(1000);

shearingX();

draw\_object();

shearingY();

draw\_object();

getch();

return 0;

}

**Lập trình mô phỏng cho phép lật đối tượng (đối xứng qua các trục tọa độ)**

1. Lật qua đường chéo chính

#include <graphics.h>

#include <math.h>

//khai bao bien

int n;

int xc[100];

int yc[100];

int xs;

int ys;

int degree;

float scalex;

float scaley;

//chuong trinh con

void draw\_object()

{

int i;

for(i=0;i<n;i++)

{

line(xc[i],yc[i],xc[(i+1)%n],yc[(i+1)%n]);

delay(1000);

}

}

//nhap thong so

void nhapdulieu()

{

int i;

printf("Nhap so canh cua da giac = ");

scanf("%d",&n);

for (i=0;i<n;i++)

{

printf("Toa do x cho canh %d = ",i);

scanf("%d",&xc[i]);

printf("Toa do y cho canh %d = ",i);

scanf("%d",&yc[i]);

}

}

//Lat theo goc toa do la giao nhau y=x va x=y

void FlipDiagonal()

{

int tempCord;

int i;

int tempX;

int tempY;

tempY=getmaxx()/2;

tempX=getmaxy()/2;

for(i=0;i<n;i++)

{

xc[i]=tempX+(tempY-xc[i]);

yc[i]=tempY+(tempX-yc[i]);

printf("(%d,%d,%d,%d)\t",xc[i],yc[i],getmaxx(),getmaxy());

}

//Ve duong phan cach theo duong cheo chinh

line(0,getmaxy(),getmaxx(),0);

}

//chuong trinh chinh

int main()

{

nhapdulieu();

initwindow(800,800);

draw\_object();

delay(1000);

FlipDiagonal();

draw\_object();

getch();

return 0;

}

**2. Lật qua đường chéo phụ**

#include <graphics.h>

#include <math.h>

//khai bao bien

int n;

int xc[100];

int yc[100];

int xs;

int ys;

int degree;

float scalex;

float scaley;

//chuong trinh con

void draw\_object()

{

int i;

for(i=0;i<n;i++)

{

line(xc[i],yc[i],xc[(i+1)%n],yc[(i+1)%n]);

delay(1000);

}

}

//nhap thong so

void nhapdulieu()

{

int i;

printf("Nhap so canh cua da giac = ");

scanf("%d",&n);

for (i=0;i<n;i++)

{

printf("Toa do x cho canh %d = ",i);

scanf("%d",&xc[i]);

printf("Toa do y cho canh %d = ",i);

scanf("%d",&yc[i]);

}

}

//Lat theo goc toa do la giao nhau y=x va x=y

void FlipDiagonalSubmain()

{

int tempCord;

int i;

int tempX;

int tempY;

tempY=getmaxx()/2;

tempX=getmaxy()/2;

for(i=0;i<n;i++)

{

xc[i]=tempY+(tempY-xc[i]);

yc[i]=getmaxy()-yc[i];

printf("(%d,%d,%d,%d)\t",xc[i],yc[i],getmaxx(),getmaxy());

}

//Ve duong phan cach theo duong cheo chinh

line(0,0,getmaxx(),getmaxy());

}

//chuong trinh chinh

int main()

{

nhapdulieu();

initwindow(800,800);

draw\_object();

delay(1000);

FlipDiagonalSubmain();

draw\_object();

getch();

return 0;

}

**3. Lật qua gốc tọa độ**

#include <graphics.h>

#include <math.h>

//khai bao bien

int n;

int xc[100];

int yc[100];

int xs;

int ys;

int degree;

float scalex;

float scaley;

//chuong trinh con

void draw\_object()

{

int i;

for(i=0;i<n;i++)

{

line(xc[i],yc[i],xc[(i+1)%n],yc[(i+1)%n]);

delay(1000);

}

}

//nhap thong so

void nhapdulieu()

{

int i;

printf("Nhap so canh cua da giac = ");

scanf("%d",&n);

for (i=0;i<n;i++)

{

printf("Toa do x cho canh %d = ",i);

scanf("%d",&xc[i]);

printf("Toa do y cho canh %d = ",i);

scanf("%d",&yc[i]);

}

}

//Lat theo goc toa do la giao nhau y=x va x=y

void FlipCordinate()

{

int tempCord;

int i;

int tempX;

int tempY;

tempY=getmaxx()/2;

tempX=getmaxy()/2;

for(i=0;i<n;i++)

{

xc[i]=tempY+(tempY-xc[i]);

yc[i]=tempX+(tempX-yc[i]);

}

//Ve duong phan cach theo chieu doc

for(i=0;i<getmaxy();i++)

{

putpixel(tempY,i,GREEN);

putpixel(i,tempX,RED);

}

}

//chuong trinh chinh

int main()

{

nhapdulieu();

initwindow(800,800);

draw\_object();

delay(1000);

FlipCordinate();

draw\_object();

getch();

return 0;

}

**4. Lật qua 2 trục x và y**

#include <graphics.h>

#include <math.h>

#include <stdio.h>

//khai bao bien

int n;

int xc[100];

int yc[100];

int xs;

int ys;

int degree;

float scalex;

float scaley;

//chuong trinh con

void draw\_object()

{

int i;

for(i=0;i<n;i++)

{

line(xc[i],yc[i],xc[(i+1)%n],yc[(i+1)%n]);

delay(1000);

}

}

//nhap thong so

void nhapdulieu()

{

int i;

printf("Nhap so canh cua da giac = ");

scanf("%d",&n);

for (i=0;i<n;i++)

{

printf("Toa do x cho canh %d = ",i);

scanf("%d",&xc[i]);

printf("Toa do y cho canh %d = ",i);

scanf("%d",&yc[i]);

}

}

//Lat theo truc y qua duong thang y=x

void FlipV()

{

int tempY;

int i;

tempY=getmaxx()/2;

for(i=0;i<n;i++)

{

xc[i]=tempY+(tempY-xc[i]);

}

//Ve duong phan cach theo chieu doc

for(i=0;i<getmaxy();i++)

{

putpixel(tempY,i,GREEN);

}

}

//Lat theo truc x=y

void FlipH()

{

int tempX;

int i;

tempX=getmaxy()/2;

for(i=0;i<n;i++)

{

yc[i]=tempX+(tempX-yc[i]);

}

//Ve duong phan cach theo chieu doc

for(i=0;i<getmaxx();i++)

{

putpixel(i,tempX,RED);

}

}

//chuong trinh chinh

int main()

{

nhapdulieu();

initwindow(800,800);

draw\_object();

delay(1000);

FlipV();

draw\_object();

FlipH();

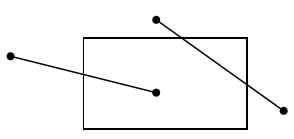
draw\_object();

getch();

return 0;

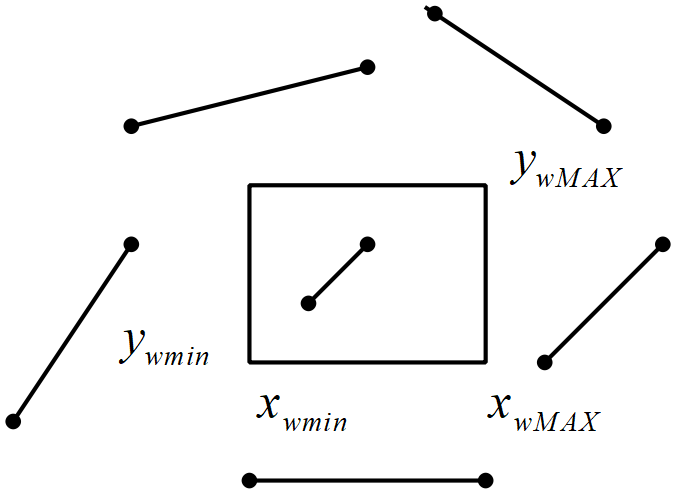
}

**Giải thuật Cohen-Sutherland**

Giả sử (x,y) là toạ độ của một điểm, vậy điểm đó được hiển thị khi thoả mãn:

* Xwmin <= x <= Xwmax
* Ywmin <= y <= Ywmax

**Ý tưởng:** Các đoạn thẳng có thể rơi vào các trường hợp sau:

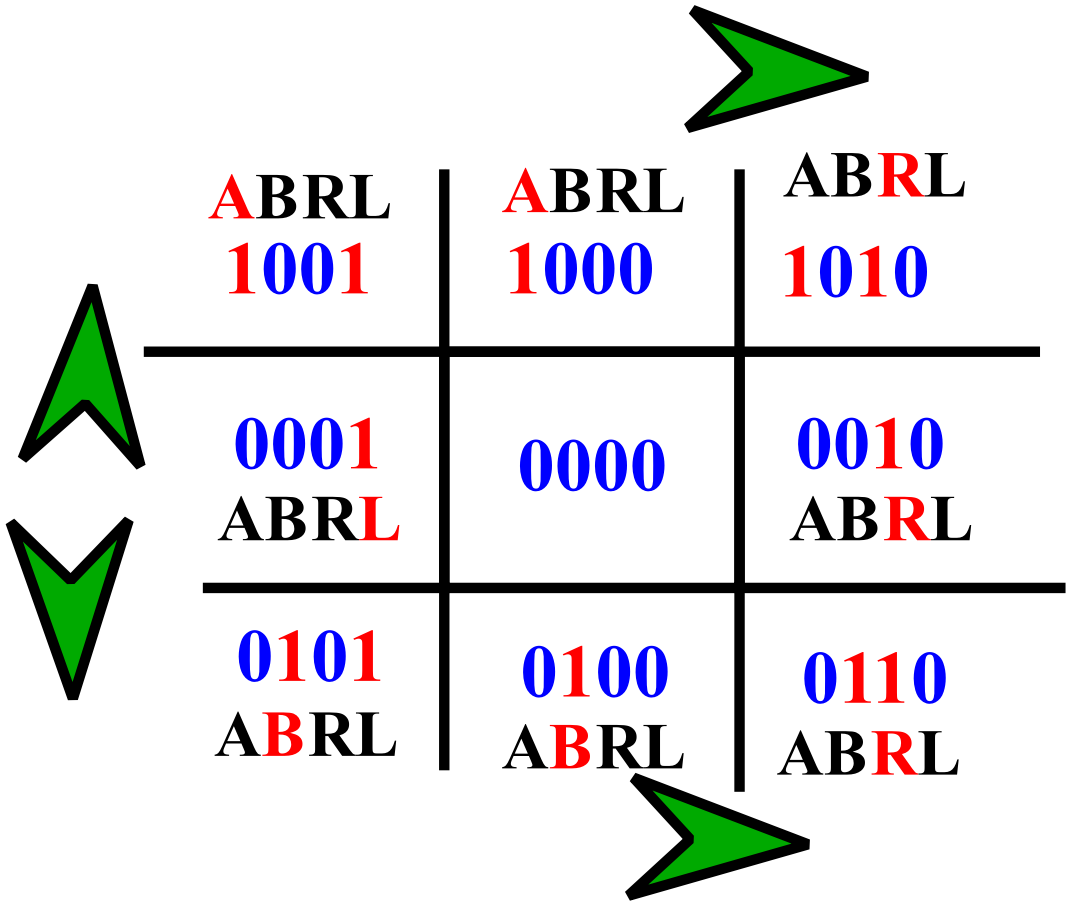
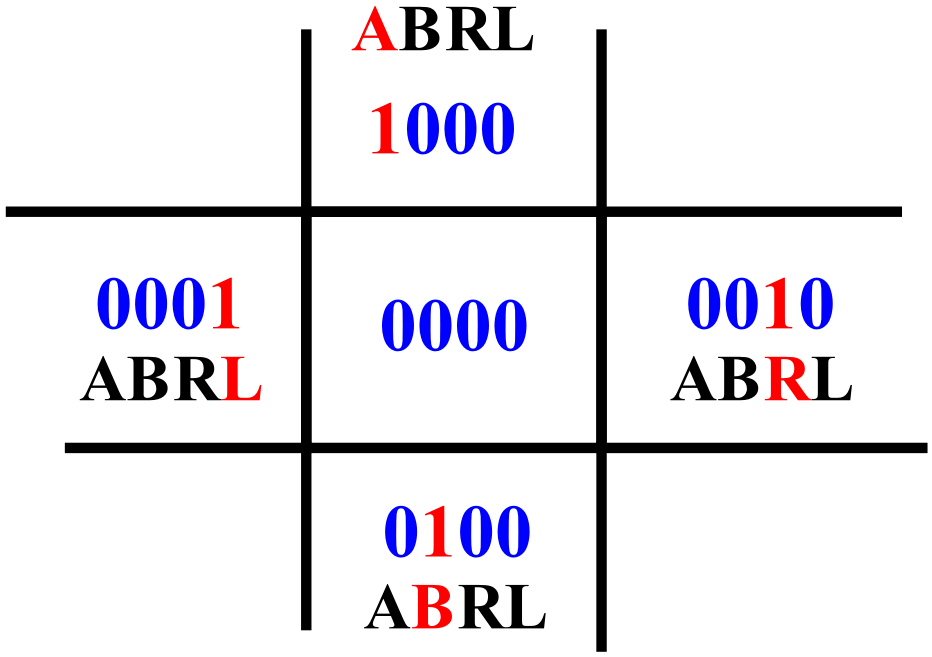
* Hiển thị (visible): cả hai đầu cuối của đoạn thẳng đều nằm bên trong cửa sổ
* Không hiển thị (invisible): đoạn thẳng xác định nằm ngoài cửa sổ. Điều này xảy ra khi  
  đoạn thẳng từ (x1,y1) đến (x2,y2) thoả màn bất kỳ 1 trong 4 bất đẳng thức sau:

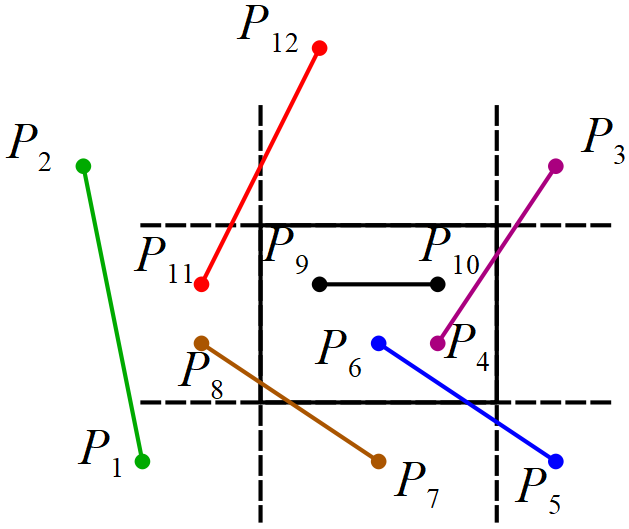
**Giải thuật Cohen – Sutherland**

***Bước 1. Mã hóa các điểm***

Gán mã vùng 4-bit cho mỗi điểm cuối của đoạn thẳng ABRL – Above Below – Right - Left

Mã vùng được xác định theo 9 vùng của mặt phẳng mà các điểm cuối nằm vào đó. Một bít được cài đặt true (1) hoặc false (0).



Ví dụ:

***Bước 2. Tìm số điểm giới hạn bởi đường thẳng và cửa sổ cắt***

* Nếu chứng tỏ đoạn thẳng nằm ngoài cửa sổ cắt
* Nếu chứng tỏ đoạn thẳng giao với x cạnh cửa sổ cắt. Cụ thể:
  + Nếu thì cả 2 điểm (đoạn thẳng) đều thuộc cửa sổ cắt (không cần tìm tọa độ giao điểm)
  + Nếu thì đoạn thẳng giao với 1 cạnh của cửa sổ cắt (có 1 giao điểm)
  + Nếu thì đoạn thẳng giao với 2 cạnh của cửa sổ cắt (có 2 giao điểm)

**Chú ý:** *Trong trường hợp đoạn P11 P12, mặc dù cả 2 mã hóa đều khác 0 và AND bit bằng 0. Nhưng thực tế đoạn thẳng này nằm bên ngoài cửa sổ cắt. Do đó, giải thuật được điều chỉnh: tìm tọa độ giao điểm – sau đó kiểm tra hoành độ và tung độ, nếu tọa đọ giao điểm nằm ngoài cửa sổ cắt thì tiến hành loại bỏ các điểm này ra khỏi kết quả.*

Không có giao điểm

Có 1 giao điểm

Có 1 giao điểm

Có 2 giao điểm

Có 2 giao điểm

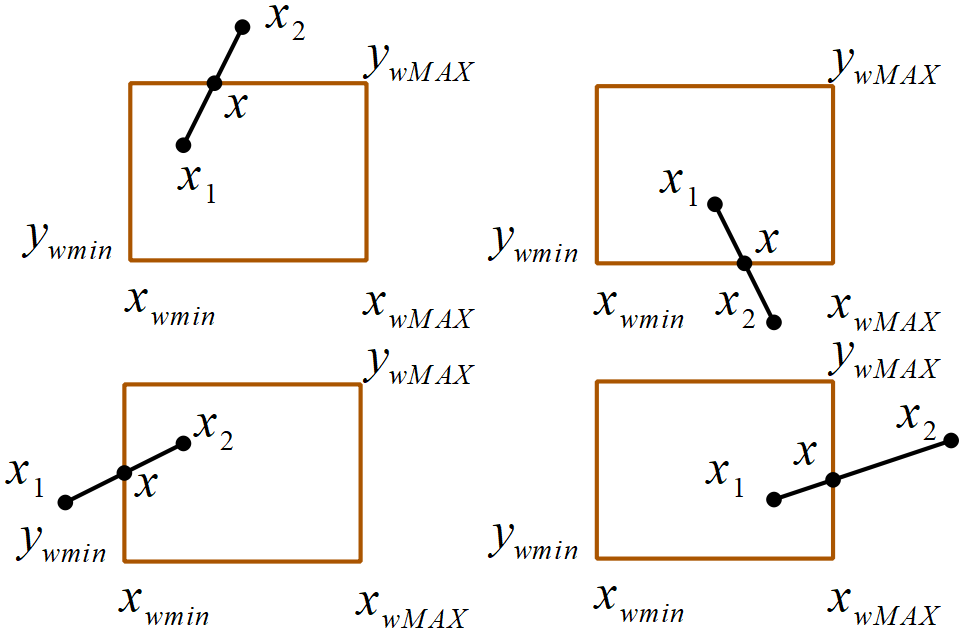
Thuộc cửa sổ cắt

|  |  |
| --- | --- |
| A AND B  0 AND 0: 0  0 AND 1: 0  1 AND 0: 0  1 AND 1: 1 | Sơ dồ mạch điện:  **1: ngắn mạch / dẫn điện**  **0: mạch hở / không có điện**  **B**  **A** |

**Bước 3. Xác định hoành độ/Tung độ giao điểm**

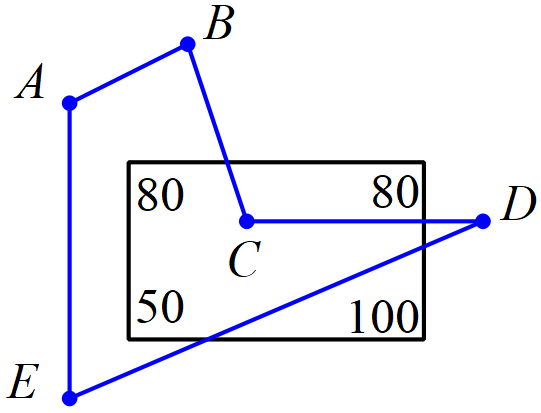
hay

hay

Từ bộ 4 mã hóa: ABRL ta có (ưu tiên các bit từ A – B – R – L):

Hệ số góc (m) tại giao điểm:

* Kiểm tra điều kiện thỏa mãn nếu rơi vào trường hợp đặc biệt như P11 - P12

**Bài tập áp dụng :**

Cho cửa sổ cắt giới hạn bởi:

Cho đa giác giới hạn bởi các đỉnh:

Xác định tọa độ giao điểm giới hạn bởi cửa sổ cắt và đa giác.

***Bước 1. Xây dựng bộ 4 mã hóa và tìm số giao điểm***

Không có giao điểm

Có 1 giao điểm

Có 1 giao điểm

Có 2 giao điểm

Không có giao điểm

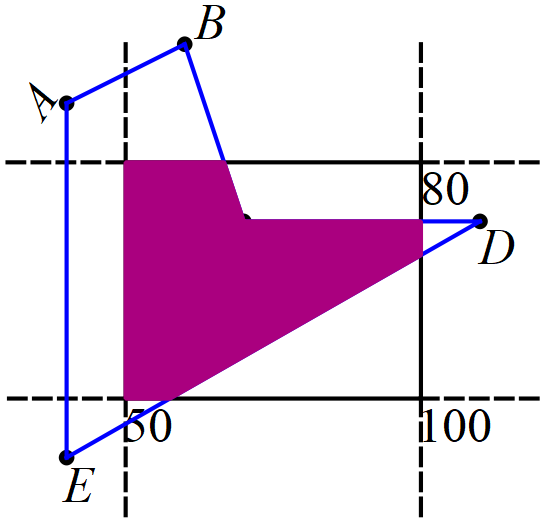
***Bước 2. Tìm tọa độ giao điểm theo thứ tự ưu tiên ABRL***

* Xét cạnh BC có
  + B (1000) nên:

hay

* Xét cạnh CD có
  + D (0001) nên:

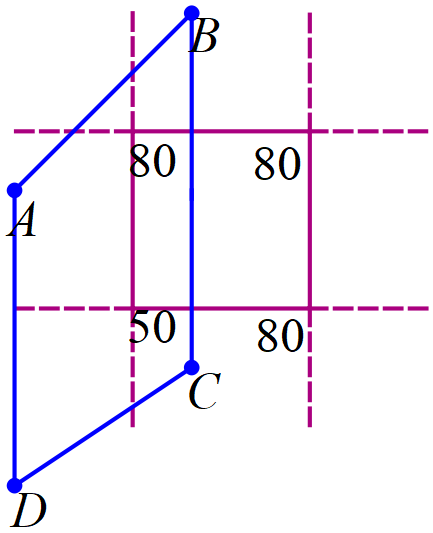
hay

* Xét cạnh DE có
  + D (0001) nên:

hay

* + E (0**1**10) nên:

hay



**Bài tập 2.**

Cho cửa sổ cắt giới hạn bởi:

Cho đa giác giới hạn bởi các đỉnh:

Xác định tọa độ giao điểm giới hạn bởi cửa sổ cắt và đa giác.

***Bước 1. Xây dựng bộ 4 mã hóa và tìm số giao điểm***

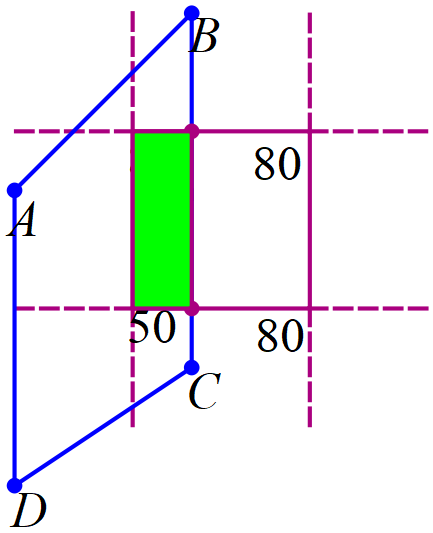
Có 2 giao điểm

Có 2 giao điểm

Không giao điểm

Không có giao điểm

***Bước 2. Tìm tọa độ giao điểm theo thứ tự ưu tiên ABRL***

* Xét cạnh AB có
  + B (1000) nên:

hay

* + A(0001) nên:

hay

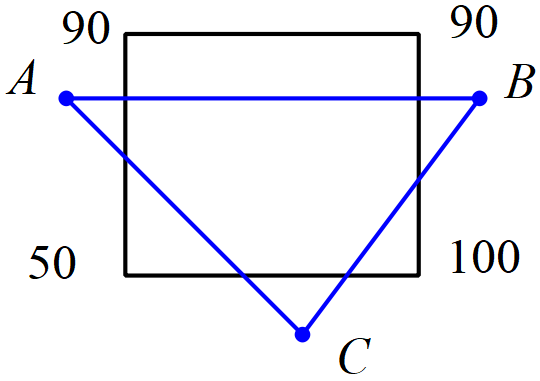
* Xét cạnh BC có
  + B (1000) nên:

Hay

* + C (0100) nên:

Hay

**Bài tập 3** **.**

Cho cửa sổ cắt giới hạn bởi:

Cho đa giác giới hạn bởi các đỉnh:

Xác định tọa độ giao điểm giới hạn bởi cửa sổ cắt và đa giác.

***Bước 1. Xây dựng bộ 4 mã hóa và tìm số giao điểm***

Có 2 giao điểm

Có 2 giao điểm

Có 2 giao điểm

***Bước 2. Tìm tọa độ giao điểm theo thứ tự ưu tiên ABRL***

* Xét cạnh AB có
  + B (0010) nên:

hay

* + A(0001) nên:

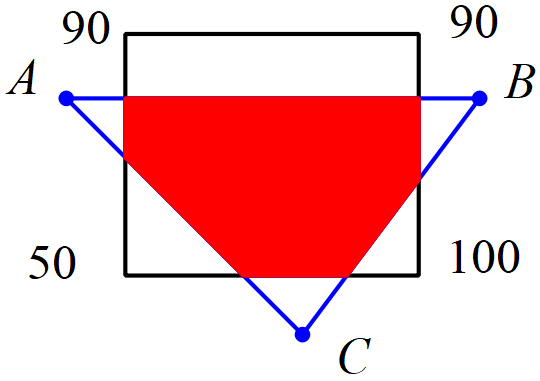
hay

* Xét cạnh BC có
  + B (0010) nên:

Hay

* + C (0100) nên:

Hay

* Xét cạnh AC có
  + C (0100) nên:

Hay

* + A(0001) nên:

hay

**Có thể rút ra nhận xét:**

Trong trường hợp ABRL của (P1 # 0) và (P2 # 0) và (P1 AND P2 = 0) và (dy = 0) thì:

và

**Lập trình mô phỏng:**

#include<stdio.h>

#include<graphics.h>

#define maxgd 100

#include<math.h>

int xwmin = 50;

int xwmax = 100;

int ywmin = 50;

int ywmax = 80;

int xa;

int ya;

int xb;

int yb;

float m;

int X;

int Y;

int X1;

int X2;

int Y1;

int Y2;

int mahoadinh[2];

void nhaptoado()

{

printf("Nhap to do 2 diem cua duong thang: ");

printf("xa = "); scanf("%d",&xa);

printf("ya = "); scanf("%d",&ya);

printf("xb = "); scanf("%d",&xb);

printf("yb = "); scanf("%d",&yb);

//ve cua so cat va duong thang AB

initwindow(400,400);

setcolor(RED);

line(xwmin,ywmin,xwmax,ywmin);

line(xwmax,ywmin,xwmax,ywmax);

line(xwmax,ywmax,xwmin,ywmax);

line(xwmin,ywmax,xwmin,ywmin);

setcolor(BLUE);

line(xa,ya,xb,yb);

}

int ABRL(int tmpx, int tmpy)

{

int code;

//ben trong

if(xwmin<=tmpx && tmpx<=xwmax && ywmin<=tmpy && tmpy<=ywmax)

{

code=0;

}

//ben trai

if(tmpx<xwmin && ywmin<tmpy && tmpy<ywmax)

{

code=1;

}

//ben phai

if(tmpx>xwmax && ywmin<=tmpy && tmpy<=ywmax)

{

code=2;

}

//ben tren

if(tmpy>ywmax && tmpx>xwmin && tmpx<xwmax)

{

code=8;

}

//ben duoi

if(tmpy<ywmin && tmpx>xwmin && tmpx<xwmax)

{

code=4;

}

//goc duoi ben trai

if(tmpy<ywmin && tmpx<xwmin)

{

code=5;

}

//goc duoi ben phai

if(tmpy<ywmin && tmpx>xwmax)

{

code=6;

}

//goc tren ben trai

if(tmpy>ywmax && tmpx<xwmin)

{

code=9;

}

//goc tren ben phai

if(tmpy>ywmax && tmpx>xwmax)

{

code=10;

}

return code;

}

void mahoavasogiaodiem()

{

mahoadinh[0]=ABRL(xa,ya);

mahoadinh[1]=ABRL(xb,yb);

printf("\nMa hoa diem A: %d",ABRL(xa,ya));

printf("\nMa hoa diem B: %d",ABRL(xb,yb));

if((mahoadinh[0]& mahoadinh[1])!=0)

{

printf("\nDoan thang nam ben ngoai cua so");

}

else

{

if(mahoadinh[0]==0 && mahoadinh[1]==0)

{

printf("\nDoan thang nam ben trong cua so");

}

else

{

if((mahoadinh[0]!=0 && mahoadinh[1])!=0)

{

printf("\nCo 2 giao diem");

}

else

{

printf("\nCo 1 giao diem");

}

}

}

}

void cohen\_sutherland(int CODE)

{

X=0;

Y=0;

if(CODE==1) //diem nam ben trai L=1

{

X=xwmin;

Y=round(ya+m\*(X-xa));

}

if(CODE==2) //diem nam ben phai R=1

{

X=xwmax;

Y=round(ya+m\*(X-xa));

}

if(CODE==4) //diem nam ben duoi B=1

{

Y=ywmin;

X=round(xa+(Y-ya)/m);

}

if(CODE==8) //diem nam ben tren A=1

{

Y=ywmax;

X=round(xa+(Y-ya)/m);

}

if(CODE==9) //diem nam tren ben trai A=1 va L=1 (CODE 1)

{

X=xwmin;

Y=round(ya+m\*(X-xa));

}

if(CODE==10) //diem nam tren ben phai A=1 va R=1 (CODE 8)

{

Y=ywmax;

X=round(xa+(Y-ya)/m);

}

if(CODE==5) //diem nam duoi ben trai B=1 va L=1 (CODE 1)

{

X=xwmin;

Y=round(ya+m\*(X-xa));

}

if(CODE==6) //diem nam duoi ben phai B=1 va R=1 (CODE 4)

{

Y=ywmin;

X=round(xa+(Y-ya)/m);

}

//kiem tra dieu kien 2 dinh deu nam ngoai cua so cat

if(X<xwmin|| X>xwmax)

{

X=0;

}

if(Y<ywmin|| Y>ywmax)

{

Y=0;

}

}

void timgiaodiem()

{

//ca 2 diem deu nam ben trong cua so cat

if(mahoadinh[0]==0 && mahoadinh[1]==0)

{

X1=xa;Y1=ya;X2=xb;Y2=yb;

}

//Khi dy=0, duong thang song song voi truc x

if(ya==yb)

{

//diem a nam ben trong va diem b nam ben ngoai

if(mahoadinh[0]==0 && mahoadinh[1]!=0)

{

//diem b nam ben trai cua so cat

if(xa>xb)

{

X1=xa;Y1=ya;X2=xwmin;Y2=ya;//Y2=yb

}

//diem b nam ben phai cua so cat

if(xa<xb)

{

X1=xa;Y1=ya;X2=xwmax;Y2=ya;//Y2=yb

}

}

//diem b nam ben trong va diem a nam ben ngoai

if(mahoadinh[0]!=0 && mahoadinh[1]==0)

{

//diem a nam ben trai cua so cat

if(xa<xb)

{

X1=xb;Y1=yb;X2=xwmin;Y2=ya;//Y2=yb

}

//diem a nam ben phai cua so cat

else

{

X1=xb;Y1=yb;X2=xwmax;Y2=ya;//Y2=yb

}

}

//ca 2 diem deu ben ngoai

else

{

X1=xwmin;Y1=ya;X2=xwmax;Y2=ya;//Y2=yb

}

}

//Khi dx==0, doan thang song song voi truc y

if(xa==xb)

{

//diem a nam ben trong va diem b nam ben ngoai

if(mahoadinh[0]==0 && mahoadinh[1]!=0)

{

//diem b nam ben tren cua so cat

if(ya>yb)

{

X1=xa;Y1=ya;X2=xa;Y2=ywmin;//X2=xb

}

//diem b nam ben duoi cua so cat

else

{

X1=xa;Y1=ya;X2=xa;Y2=ywmax;//X2=yb

}

}

//diem b nam ben trong va diem a nam ben ngoai

if(mahoadinh[0]!=0 && mahoadinh[1]==0)

{

//diem a nam ben tren cua so cat

if(ya<yb)

{

X1=xb;Y1=yb;Y2=ywmin;X2=xa;//X2=xb

}

//diem a nam ben duoi cua so cat

else

{

X1=xb;Y1=yb;Y2=ywmax;X2=xa;//X2=xb

}

}

//ca 2 diem deu ben ngoai

else

{

X1=xa;Y1=ywmin;X2=xb;Y2=ywmax;//X2=xa

}

}

//khi ca dx va dy deu <> 0, co 2 diem cat

if(mahoadinh[0]!=0 && mahoadinh[1]!=0 && (mahoadinh[0] & mahoadinh[1])==0)

{

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

cohen\_sutherland(mahoadinh[0]);

X1=X;

Y1=Y;

cohen\_sutherland(mahoadinh[1]);

X2=X;

Y2=Y;

printf("\nX1= %d, Y1= %d, X2= %d, Y2= %d ",X1,Y1,X2,Y2);

}

//ve duong thang noi 2 diem

setcolor(GREEN);

setlinestyle(1,0,1);

line(X1,Y1,X2,Y2);

}

int main()

{

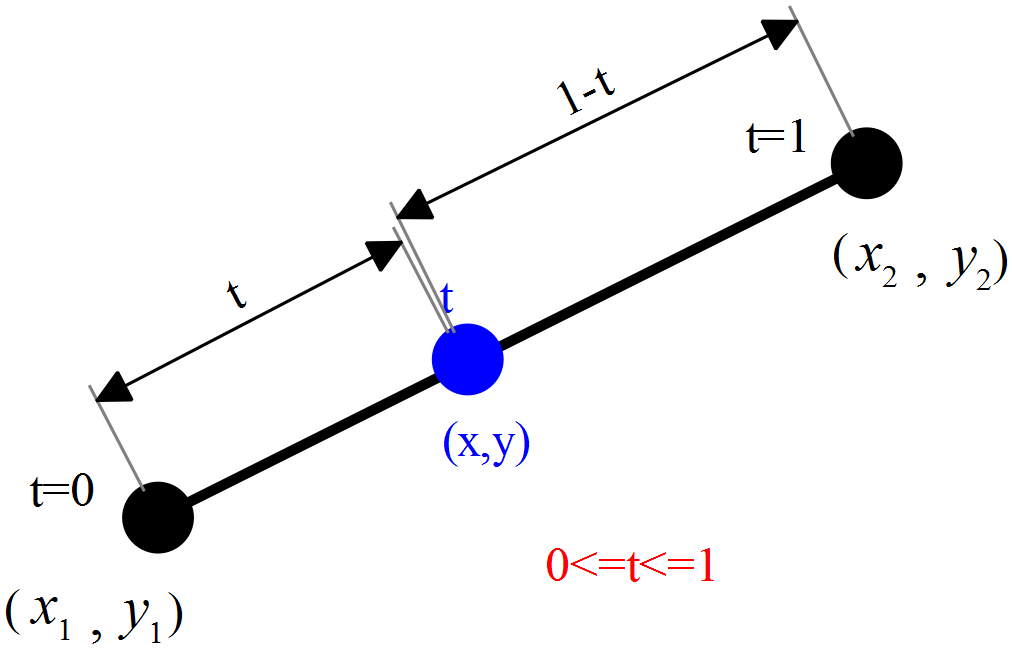
nhaptoado();

mahoavasogiaodiem();

timgiaodiem();

getch();

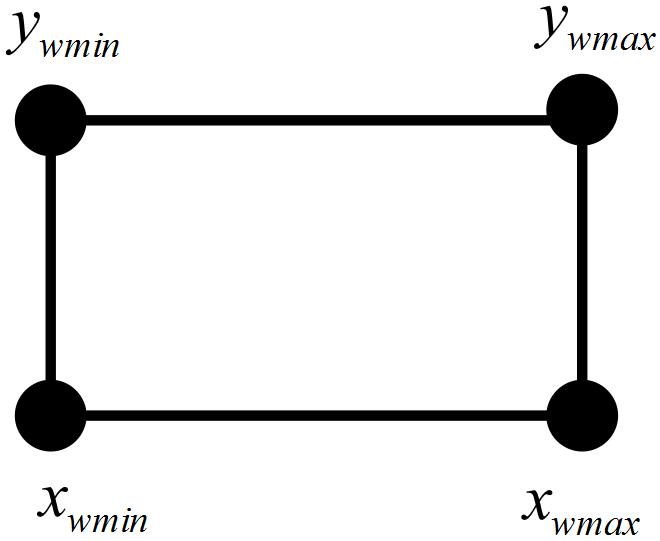
}

***Thuật toán Liang – Barsky***

Xét đường thẳng đi qua 2 điểm (x1,y1) và (x2,y2) như sau:

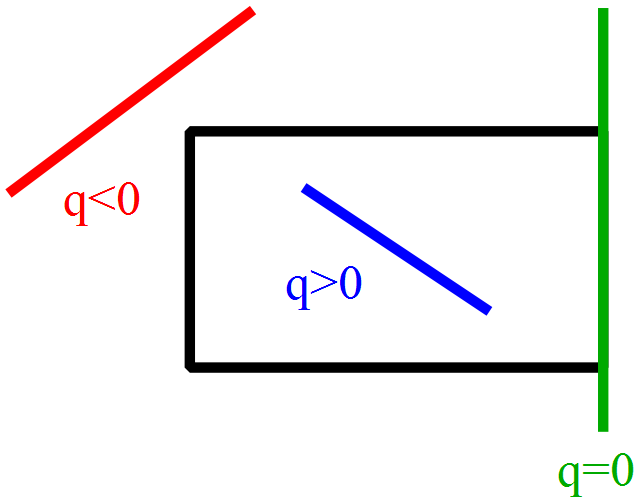
Gọi t=0 tại điểm (x1,y1) và t=1 tại điểm (x2,y2)

Xét 1 điểm thuộc đường thẳng có tọa độ (x,y). Khi đó:

* Xét cửa sổ clip được giới hạn bởi (xwmin,ywmin), (xwmax,ywmax)
* Do vậy:
* Hay:
* Ta có 4 bất đẳng thức sau:

Hay

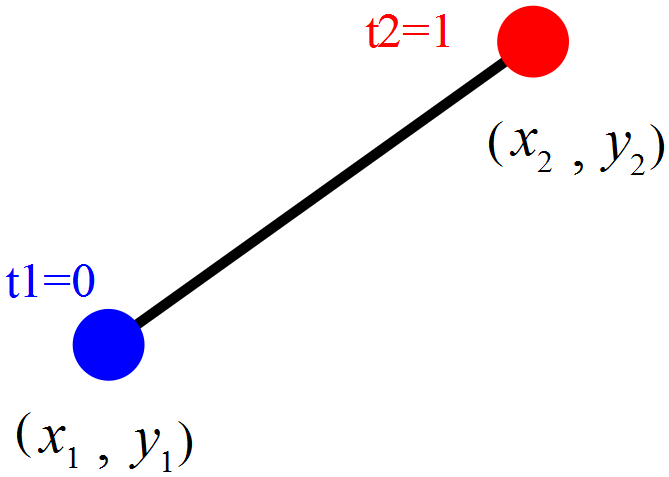
Nhân 2 vế với giá trị -1:

* Tổng quát ta có:
* Từ đó, ta có:
* Nếu pk = 0: điều đó tương đương với việc đoạn thẳng đang xét song song với cạnh thứ k  
  của hình chữ nhật clipping (đường thẳng có dy = 0 hoặc dx=0)

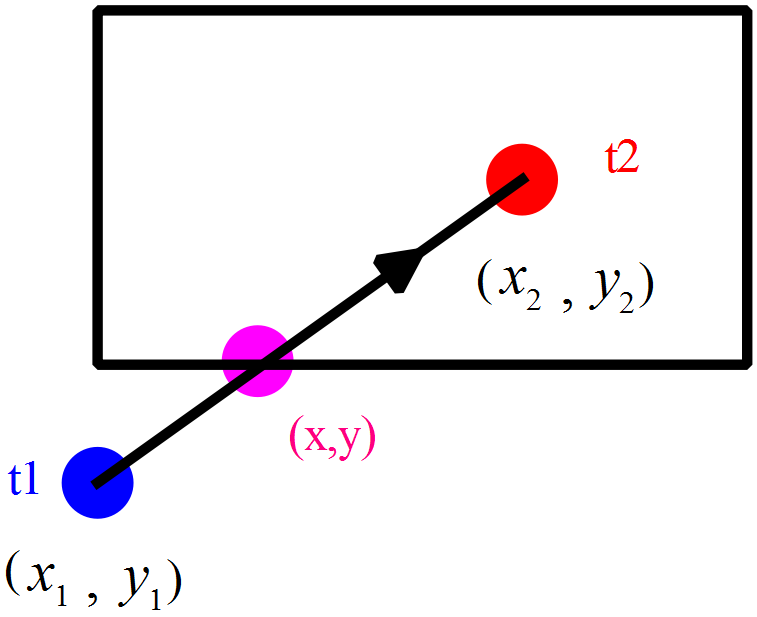
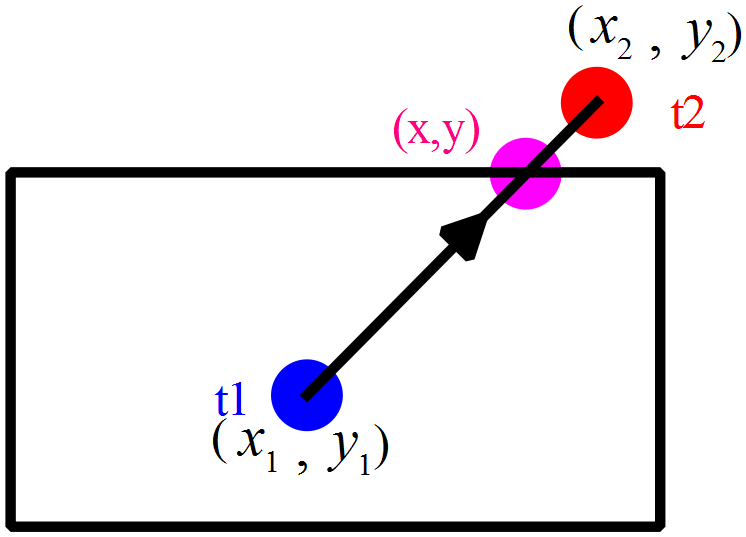
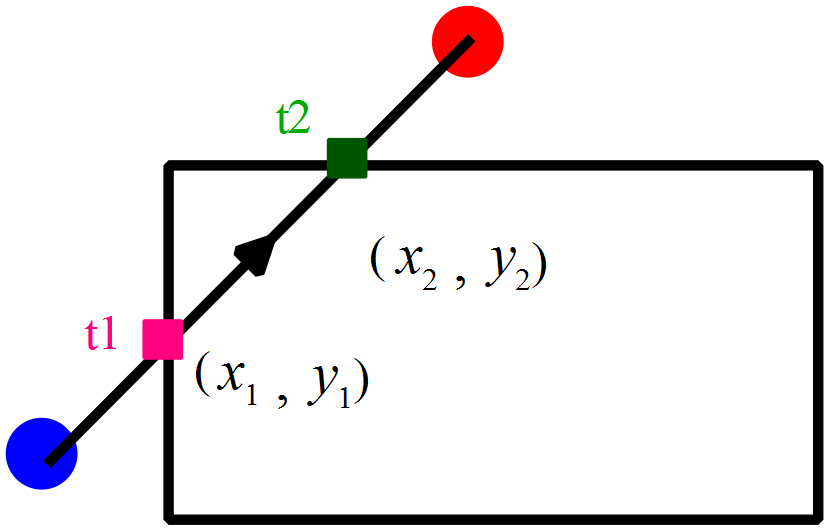
➊ Nếu qk < 0 đoạn thẳng nằm ngoài cửa sổ (hệ bất phương trình trên vô nghiệm)

➋ Nếu qk > 0 thì đoạn thẳng nằm trong 1 phần của cửa sổ clipping

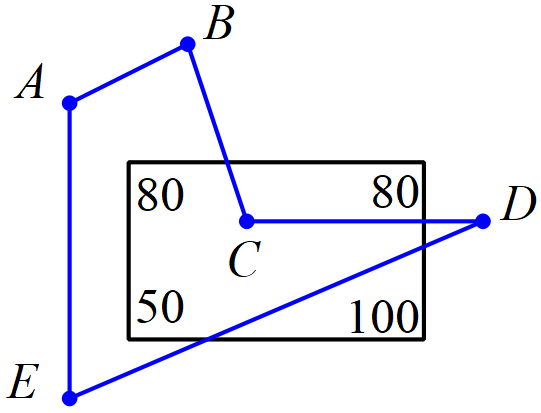
➌ Nếu qk = 0 thì đoạn thẳng nằm trên đường biên của cạnh k (có vô số điểm cắt)

* Nếu pk <0, từ bất đẳng thức
  + Ta có:
* Nếu pk >0, từ bất đẳng thức

➎ Ta có:

* + Nếu
  + Nếu
  + Nếu
* Nhận xét:
  + Nếu t1 thay đổi giá trị tăng dần từ 0 đến , có nghĩa là đường thẳng có chiều từ noài vào trong cửa sổ clipping.
  + Nếu t2 thay đổi giá trị giảm dần từ 1 đến , có nghĩa là đường thẳng có chiều trong cửa sổ clipping ra ngoài.
  + Nếu t1 thay đổi giá trị tăng dần từ 0 đến còn t2 thay đổi giá trị giảm dần từ 1 đến , có nghĩa là đường thẳng có chiều từ ngoài cửa sổ clipping ra bên ngoài cửa sổ clipping.

**Bài tập áp dụng :**

Cho cửa sổ cắt giới hạn bởi:

Cho đa giác giới hạn bởi các đỉnh:

Xác định tọa độ giao điểm giới hạn bởi cửa sổ cắt và đa giác.

Xét cạnh BC:

* + = -10
  + = 10
  + = 30
  + = -30
  + = 10
  + = 40
  + = 50
  + = -20
* Xét trường hợp Pk < 0 (ứng với P1 và P4)
* Xét trường hợp Pk > 0 (ứng với P2 và P3)

**(Không xét)**

Xét cạnh CD:

Xét cạnh DE:

* + = 70
  + = -70
  + = 30
  + = -30
  + = 60
  + = -10
  + = 20
  + = 10
* Xét trường hợp Pk < 0 (ứng với P2, P4)
* Xét trường hợp Pk > 0 (ứng với P1, P3)

**Lập trình mô phỏng**

#include<stdio.h>

#include<graphics.h>

#include<math.h>

int xmin;

int xmax;

int ymin;

int ymax;

int xa;

int ya;

int xb;

int yb;

int dx;

int dy;

int p[4];

int q[4];

int X1;

int Y1;

int X2;

int Y2;

void nhapthongso()

{

xmin=100;

xmax=400;

ymin=100;

ymax=400;

printf("xA = ");scanf("%d",&xa);

printf("yA = ");scanf("%d",&ya);

printf("xB = ");scanf("%d",&xb);

printf("yB = ");scanf("%d",&yb);

//duong cheo cat 2 diem

//xa=60;

//ya=40;

//xb=110;

//yb=60;

//xa=70;

//ya=90;

//xb=110;

//yb=60;

//xa=70;

//ya=90;

//xb=30;

//yb=60;

//xa=60;

//ya=40;

//xb=30;

//yb=60;

//song song truc hoanh

//xa=30;

//ya=60;

//xb=70;

//yb=60;

//xa=70;

//ya=120;

//xb=70;

//yb=60;

//tinh gia tri pk va qk

dx=xb-xa;

dy=yb-ya;

p[0]=-dx;

p[1]=dx;

p[2]=-dy;

p[3]=dy;

q[0]=xa-xmin;

q[1]=xmax-xa;

q[2]=ya-ymin;

q[3]=ymax-ya;

/\*

printf("P[i] va Q[i] la:");

int i;

for(i=0;i<4;i++)

{

printf("\n%d \t",p[i]);

printf("\n%d \t",q[i]);

}

\*/

initwindow(500,500);

line(xmin,ymin,xmax,ymin);

line(xmax,ymin,xmax,ymax);

line(xmax,ymax,xmin,ymax);

line(xmin,ymax,xmin,ymin);

setcolor(BLUE);

line(xa,ya,xb,yb);

}

int check()

{

if(xa<xmin && xb<xmin)

{

return 0;

}

else

{

if(xa>xmax && ya>ymax)

{

return 0;

}

else

{

if(ya<ymin && yb<ymin)

{

return 0;

}

else

{

if(ya>ymax && yb>ymax)

{

return 0;

}

else

{

return 1;

}

}

}

}

}

float MIN(float a, float b)

{

float min;

float max;

if(a<b)

{

min=a;

}

else

{

min=b;

}

return min;

}

float MAX(float a, float b)

{

float max;

if(a<b)

{

max=b;

}

else

{

max=a;

}

return max;

}

void Liang\_Barsky()

{

int i;

float t1;

float t2;

t1=0.0;

t2=1.0;

//printf("\nCheck = %d",check());

if(check()!=0)

{

for(i=0;i<4;i++)

{

if(p[i]<0)//doi voi pk<0

{

t1=MAX(t1,(float)q[i]/p[i]);

//printf("\np[%d]=%d q[%d]= %d %.2f",i,p[i],i,q[i],(float)q[i]/p[i]);

}

else //doi voi pk>0

{

t2=MIN(t2,(float)q[i]/p[i]);

//printf("\np[%d]=%d q[%d]= %d %.2f",i,p[i],i,q[i],(float)q[i]/p[i]);

}

}

printf("\nt1 = %f, t2 = %f",t1,t2);

//tim X va Y

if(dx!=0 && dy!=0)//co 2 giao diem

{

X1=round(xa+t1\*dx);

Y1=round(ya+t1\*dy);

X2=round(xa+t2\*dx);

Y2=round(ya+t2\*dy);

printf("\n(X1 = %d, Y1 = %d, X2 = %d, Y2 = %d)",X1,Y1,X2,Y2);

//kiem tra giao diem co thoa man nam trong cua so?

if(X1<xmin || X1 >xmax || Y1 < ymin || Y1 > ymax || X2<xmin || X2 >xmax || Y2 < ymin || Y2 > ymax)

{

X1=0;

Y1=0;

X2=0;

Y2=0;

}

}

//Co 1 giao diem

else

{

//duong thang song song truc x

if(dy==0)

{

//diem a nam trong cua so

if(xa>=xmin && xa<=xmax)

{

//diem b nam trong

if(xb>=xmin && xb<=xmax)

{

X1=xa;Y1=ya;X2=xb;Y2=yb;

}

//diem b nam ben trai diem a

if(xb<xmin)

{

X1=xa;Y1=ya;X2=xmin;Y2=ya;//Y2=yb

}

//diem b nam ben phai

if(xb>xmax)

{

X1=xa;Y1=ya;X2=xmax;Y2=ya;//Y2=yb

}

}

//diem b nam ben trong cua so

if(xb>=xmin && xb<=xmax)

{

//diem a nam trong

if(xa>=xmin && xa<=xmax)

{

X1=xb;Y1=yb;X2=xa;Y2=ya;

}

//diem a nam ben trai diem a

if(xa<xmin)

{

X1=xb;Y1=yb;X2=xmin;Y2=yb;//Y2=ya

}

//diem a nam ben phai

if(xa>xmax)

{

X1=xb;Y1=yb;X2=xmax;Y2=yb;//Y2=ya

}

}

//ca 2 diem deu nam ben ngoai

if((xa<xmin && xb>xmax) || (xa>xmax && xb<xmin))

{

X1=xmin;Y1=ya;X2=xmax;Y2=ya;

}

printf("\n(X1 = %d, Y1 = %d, X2 = %d, Y2 = %d)",X1,Y1,X2,Y2);

}

else

//dx==0 //nam song song voi truc tung

{

//diem a nam trong cua so

if(ya>=ymin && ya<=ymax)

{

//diem b nam trong

if(yb>=ymin && yb<=ymax)

{

X1=xa;Y1=ya;X2=xb;Y2=yb;

}

//diem b nam ben duoi diem a

if(yb<ymin)

{

X1=xa;Y1=ya;Y2=ymin;X2=xa;//X2=xb

}

//diem b nam ben tren diem a

if(yb>ymax)

{

X1=xa;Y1=ya;Y2=ymax;X2=xa;//X2=xb

}

}

//diem b nam ben trong cua so

if(yb>=ymin && yb<=ymax)

{

//diem a nam trong

if(ya>=ymin && ya<=ymax)

{

X1=xa;Y1=ya;X2=xb;Y2=yb;

}

//diem a nam ben duoi diem b

if(ya<ymin)

{

X1=xb;Y1=yb;Y2=ymin;X2=xa;//X2=xb

}

//diem a nam ben tren diem b

if(ya>ymax)

{

X1=xb;Y1=yb;Y2=ymax;X2=xa;//X2=xb

}

}

//ca 2 diem deu nam ben ngoai

if((ya<ymin && yb>ymax) || (ya>ymax && yb<ymin))

{

X1=xa;Y1=ymin;X2=xb;Y2=ymax;

}

}

}

//ve 2 giao diem

setcolor(GREEN);

setlinestyle(1,0,1);

line(X1,Y1,X2,Y2);

}

else

{

printf("\nKhong co giao diem");

}

}

int main()

{

nhapthongso();

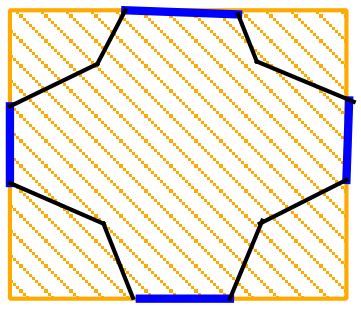
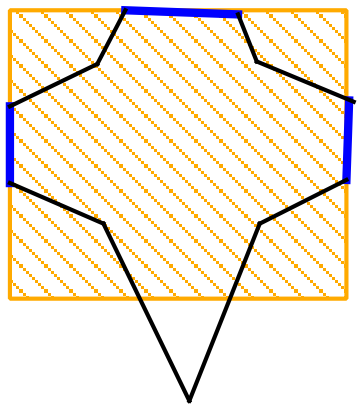
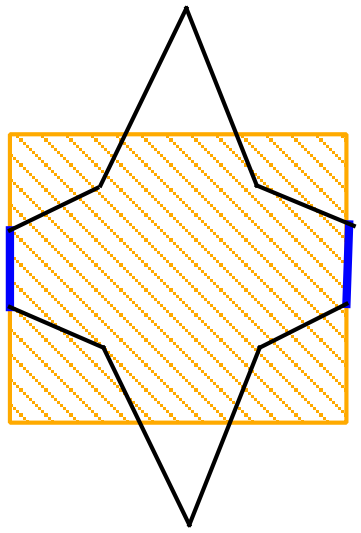
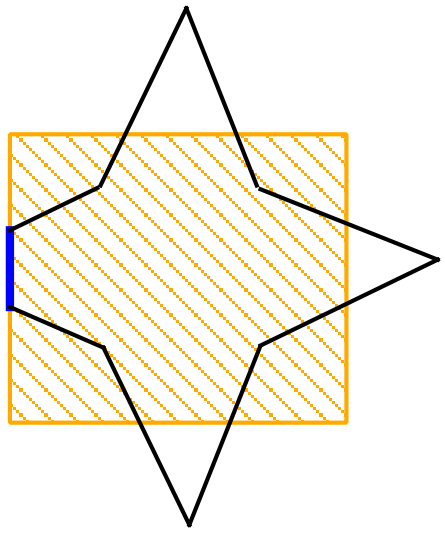
Liang\_Barsky();

getch();

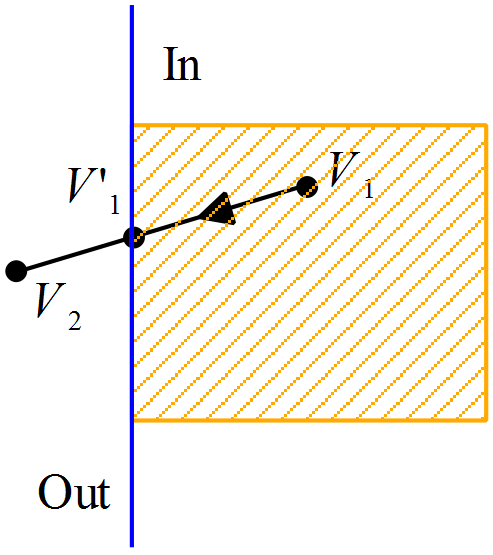
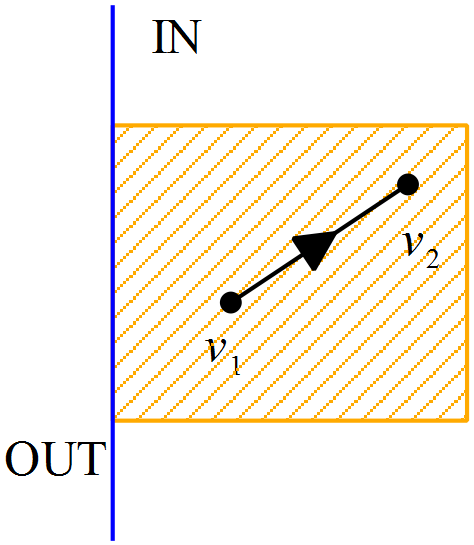
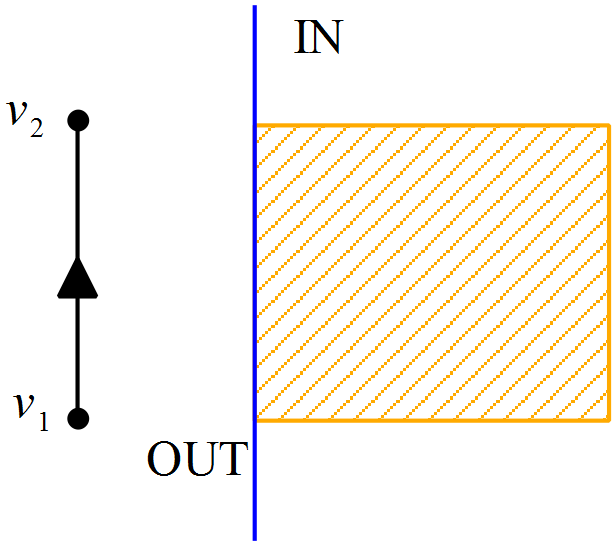
}

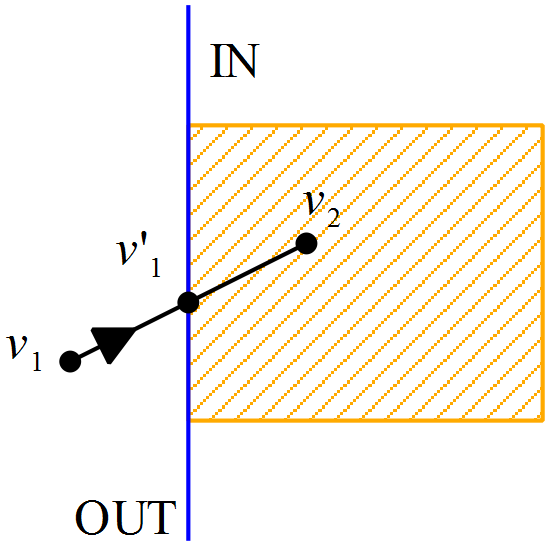
**Giải thuật Hogman**

Cho P1,......,PN là danh sách các đỉnh của đa giác. Cho cửa sổ cắt tỉa ABCD. Thứ tự các trường hợp sẽ được xén như sau:

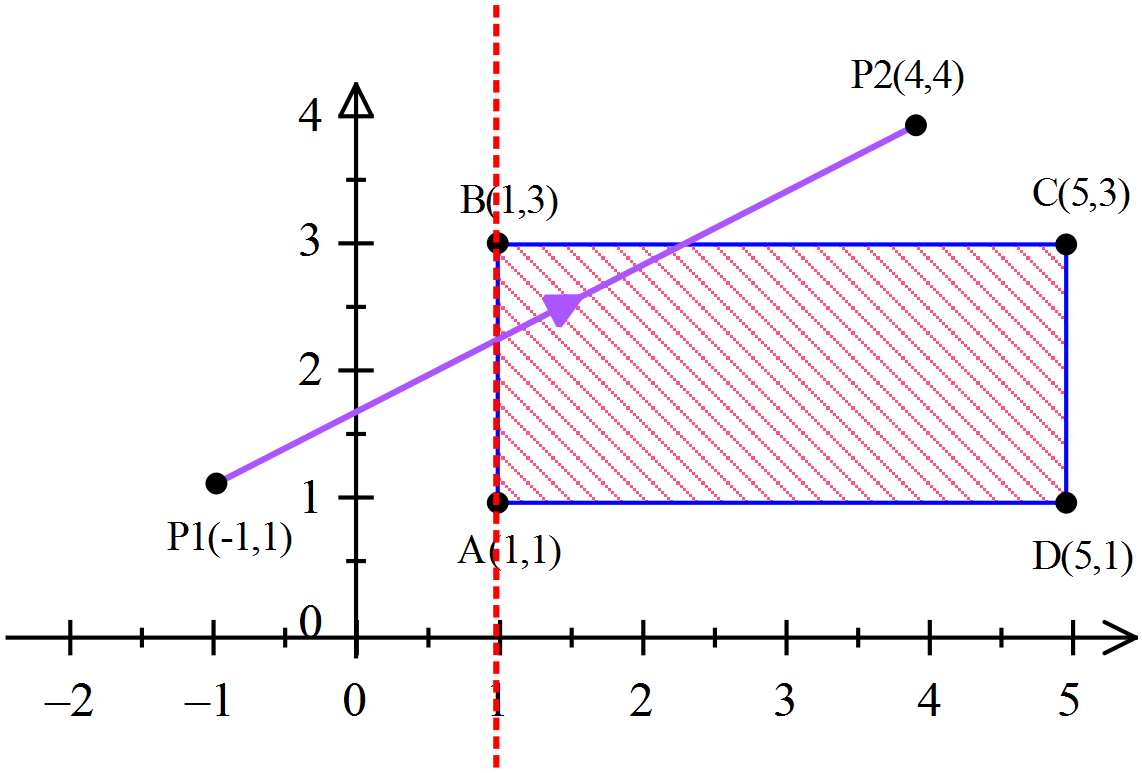


Xét 4 trường hợp:



Output (Out-In): V’1V2Output (In-Out): V’1 Output (In-In): V2 Output: NULL

**Bài tập:**

Sử dụng thuật toán Hodgman để cắt xén đoạn thẳng nối P1(-1,2) đến P2(6,4) trên cửa sổ A(1,1), B(1,3), D(3,5) và E(5,3)

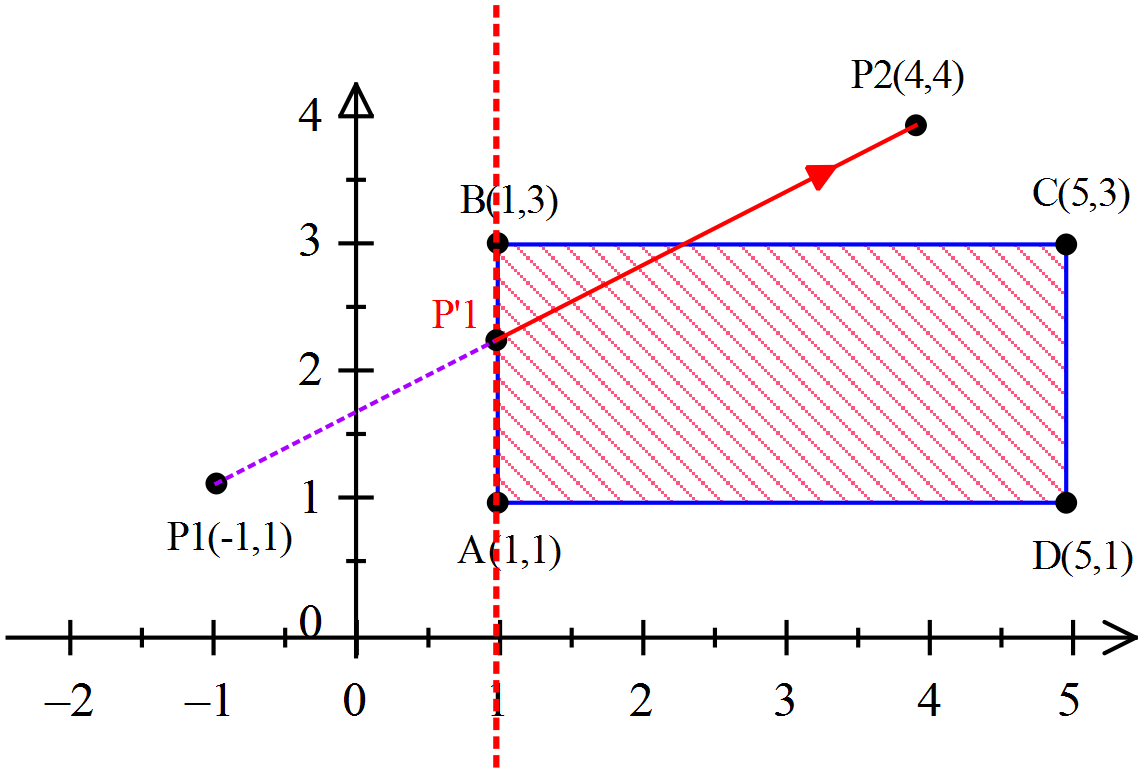
***Theo thuật toán Hodgman ta xén P1P2 dựa trên từng*** cạnh.  
➊ AB:

Điểm P1 nằm bên trái AB

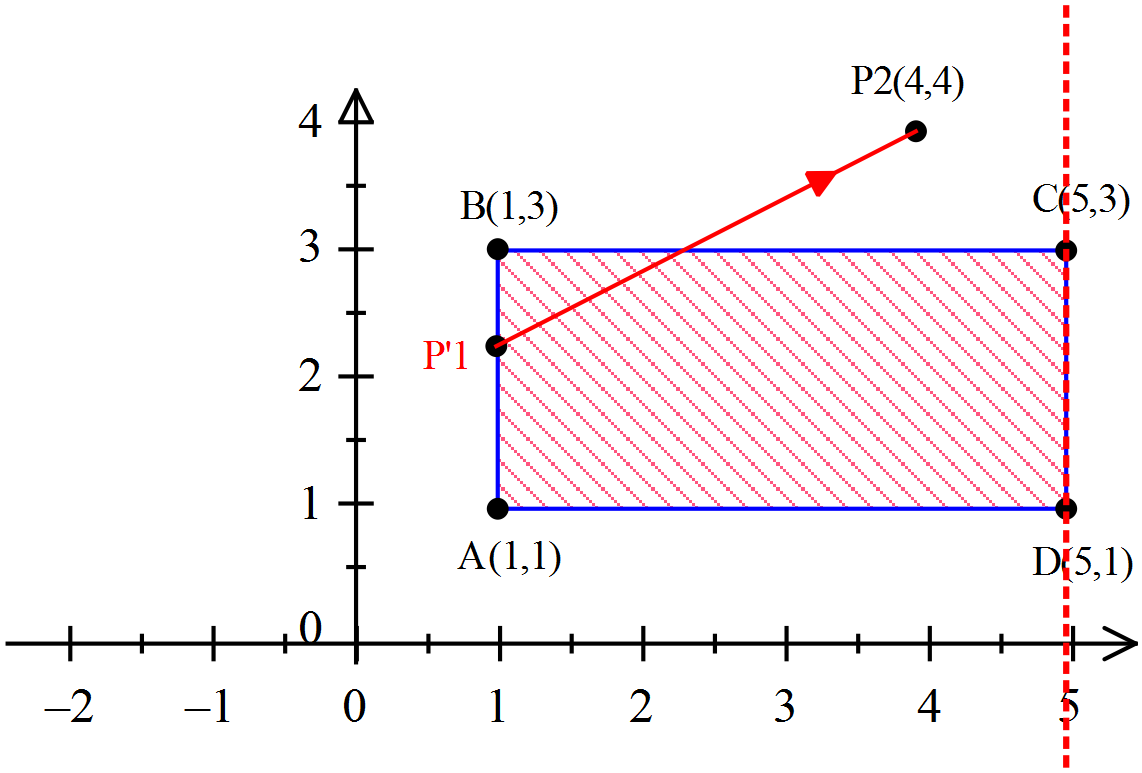
Điểm P2 nằm bên phải AB

Chiều Out-IN

Vậy P’1 P2 được lưu

Tìm giao điểm P’1 thuộc đường thằng P1P2

P’1(1;11/5)

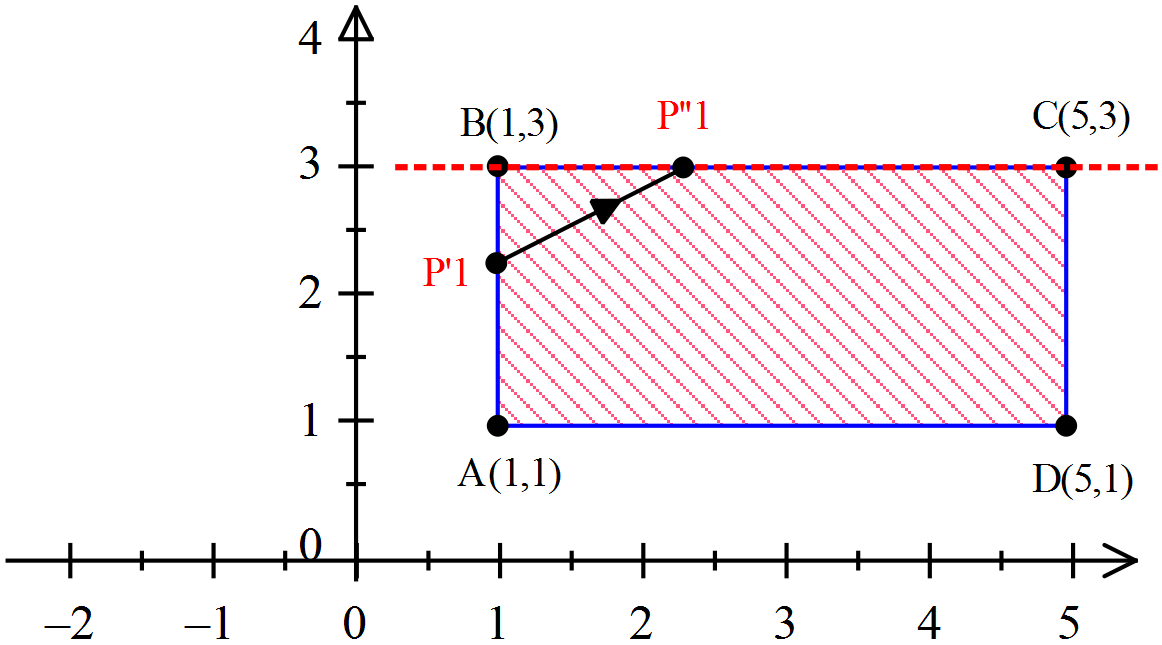
Ta xén P’1P2 dựa trên cạnh CD  
➋ CD:

Điểm P’1: nằm bên trái CD

Điểm P2: nằm bên trái CD

Chiều In-In

Vậy P2 được lưu, với P2(4,4)

Ta xén P’1P2 dựa trên cạnh  
➌ BC:

Điểm P’1: nằm bên trong BC

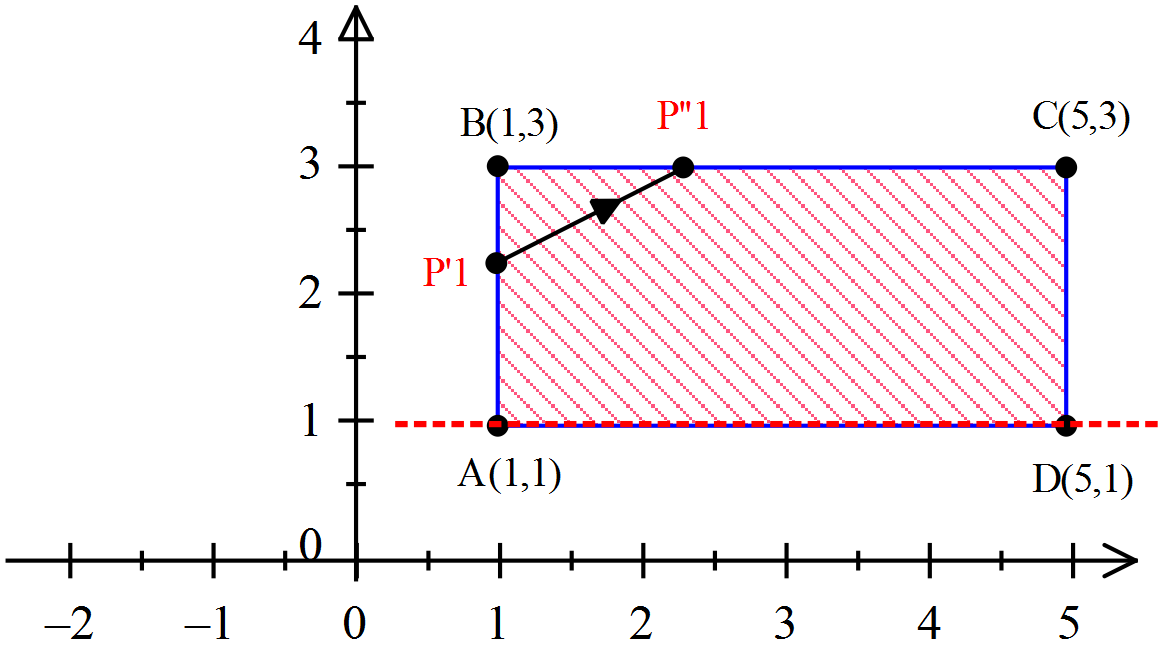
Điểm P2: nằm bên ngoài BC

Chiều In - Out

Vậy P”1 được lưu

Tìm giao điểm P’’1 thuộc đường thằng P’1P2

P’’1(7/3,3)



Ta xén P’1P’’1 dựa trên từng cạnh.  
➍ AD:

Điểm P’1: nằm bên trong AD

Điểm P’’1: nằm bên trong AD

Chiều In-In

Vậy P’1P”1 được lưu

Lập trình mô phỏng

#include <graphics.h>

#include <math.h>

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

#define OUT2IN 1

#define INSIDE 2

#define IN2OUT 3

#define OUTSIDE 4

#define INVALID 5

//khai bao bien

int xc[100];

int yc[100];

int wx[100];

int wy[100];

int pointsize;

int windowsize;

int xc\_tam[100];

int yc\_tam[100];

int pointsize\_tam;

//chuong trinh con

void draw\_object()

{

int i;

//ve cac canh cua da giac

setcolor(RED);

for(i=0;i<pointsize;i++)

{

line(xc[i],yc[i],xc[(i+1)%pointsize],yc[(i+1)%pointsize]);

delay(1000);

}

//ve cac canh cua so clipping

setcolor(BLUE);

for(i=0;i<windowsize;i++)

{

line(wx[i],wy[i],wx[(i+1)%windowsize],wy[(i+1)%windowsize]);

delay(1000);

}

}

//nhap thong so

void nhapdulieu()

{

int i;

printf("Nhap so canh cua da giac = ");

scanf("%d",&pointsize);

for (i=0;i<pointsize;i++)

{

printf("Toa do x cho canh %d = ",i);

scanf("%d",&xc[i]);

printf("Toa do y cho canh %d = ",i);

scanf("%d",&yc[i]);

}

printf("Nhap so canh cua cua so clipping = ");

scanf("%d",&windowsize);

for (i=0;i<windowsize;i++)

{

printf("Toa do x cho canh window %d = ",i);

scanf("%d",&wx[i]);

printf("Toa do y cho canh window %d = ",i);

scanf("%d",&wy[i]);

}

}

int clipLeft(int x,int x1,int x2)

{

if(x1<x && x2>=x)

{

return OUT2IN;

}

else

{

if(x1>=x && x2>=x)

{

return INSIDE;

}

else

{

if(x1>=x && x2<x)

{

return IN2OUT;

}

else

{

if(x1<x && x2<x)

{

return OUTSIDE;

}

else

{

return INVALID;

}

}

}

}

}

int clipBottom(int y,int y1,int y2)

{

if(y1<y && y2>=y)

{

return OUT2IN;

}

else

{

if(y1>=y && y2>=y)

{

return INSIDE;

}

else

{

if(y1>=y && y2<y)

{

return IN2OUT;

}

else

{

if(y1<y && y2<y)

{

return OUTSIDE;

}

else

{

return INVALID;

}

}

}

}

}

int clipRight(int x,int x1,int x2)

{

if(x1>x && x2<=x)

{

return OUT2IN;

}

else

{

if(x1<=x && x2<=x)

{

return INSIDE;

}

else

{

if(x1<=x && x2>x)

{

return IN2OUT;

}

else

{

if(x1>x && x2>x)

{

return OUTSIDE;

}

else

{

return INVALID;

}

}

}

}

}

int clipTop(int y,int y1,int y2)

{

if(y1>y && y2<=y)

{

return OUT2IN;

}

else

{

if(y1<=y && y2<=y)

{

return INSIDE;

}

else

{

if(y1<=y && y2>y)

{

return IN2OUT;

}

else

{

if(y1>y && y2>y)

{

return OUTSIDE;

}

else

{

return INVALID;

}

}

}

}

}

void leftclipper()

{

int chieu;

int i;

int xwmin;

int xa;

int ya;

int xb;

int yb;

int j;

int xnew;

int ynew;

xwmin=wx[0];//dinh 1

j=0;

for(i=0;i<pointsize;i++)

{

xa=xc[i];

ya=yc[i];

xb=xc[i+1];

yb=yc[i+1];

chieu=clipLeft(xwmin,xa,xb);

switch (chieu)

{

case OUT2IN:

{

xnew = xwmin;

ynew = ya + (float)(yb-ya)\*(float)(xwmin-xa)/(float)(xb-xa);

xc\_tam[j]=xnew;

yc\_tam[j]=ynew;

j++;

xc\_tam[j]=xb;

yc\_tam[j]=yb;

j++;

break;

}

case INSIDE:

{

xnew = xb;

ynew = yb;

xc\_tam[j]=xnew;

yc\_tam[j]=ynew;

j++;

break;

}

case IN2OUT:

{

xnew = xwmin;

ynew = ya + (float)(yb-ya)\*(float)(xwmin-xa)/(float)(xb-xa);

xc\_tam[j]=xnew;

yc\_tam[j]=ynew;

j++;

break;

}

case OUTSIDE:

{

break;

}

}

printf("\n chieu = %d\n",chieu);

}

//gan lai danh sach canh moi

for(i=0;i<j;i++)

{

xc[i]=xc\_tam[i];

yc[i]=yc\_tam[i];

}

//gan so dinh moi sau khi leftclipper

pointsize=j;

for(i=0;i<j;i++)

{

printf("(%d,%d); ",xc[i],yc[i]);

}

printf("\n");

}

void rightclipper()

{

int chieu;

int i;

int xwmax;

int xa;

int ya;

int xb;

int yb;

int j;

int xnew;

int ynew;

xwmax=wx[2];;//dinh 3

j=0;

for(i=0;i<pointsize;i++)

{

xa=xc[i];

ya=yc[i];

xb=xc[i+1];

yb=yc[i+1];

chieu=clipRight(xwmax,xa,xb);

switch (chieu)

{

case OUT2IN:

{

xnew = xwmax;

ynew = ya + (float)(yb-ya)\*(float)(xwmax-xa)/(float)(xb-xa);

xc\_tam[j]=xnew;

yc\_tam[j]=ynew;

j++;

xc\_tam[j]=xb;

yc\_tam[j]=yb;

j++;

break;

}

case INSIDE:

{

xnew = xb;

ynew = yb;

xc\_tam[j]=xnew;

yc\_tam[j]=ynew;

j++;

break;

}

case IN2OUT:

{

xnew = xwmax;

ynew = ya + (float)(yb-ya)\*(float)(xwmax-xa)/(float)(xb-xa);

xc\_tam[j]=xnew;

yc\_tam[j]=ynew;

j++;

break;

}

case OUTSIDE:

{

break;

}

}

printf("\n chieu = %d\n",chieu);

}

//gan lai danh sach canh moi

for(i=0;i<j;i++)

{

xc[i]=xc\_tam[i];

yc[i]=yc\_tam[i];

}

//gan so dinh moi sau khi rightclipper

pointsize=j;

for(i=0;i<j;i++)

{

printf("(%d,%d); ",xc[i],yc[i]);

}

printf("\n");

}

void bottomclipper()

{

int chieu;

int i;

int ywmax;

int xa;

int ya;

int xb;

int yb;

int j;

int xnew;

int ynew;

ywmax=wy[2];//dinh 3

j=0;

for(i=0;i<pointsize;i++)

{

xa=xc[i];

ya=yc[i];

xb=xc[i+1];

yb=yc[i+1];

chieu=clipTop(ywmax,ya,yb);

switch (chieu)

{

case OUT2IN:

{

xnew = xa + (float)(ywmax-ya)\*(xb-xa)/(yb-ya);

ynew = ywmax;

xc\_tam[j]=xnew;

yc\_tam[j]=ynew;

j++;

xc\_tam[j]=xb;

yc\_tam[j]=yb;

j++;

break;

}

case INSIDE:

{

xnew = xb;

ynew = yb;

xc\_tam[j]=xnew;

yc\_tam[j]=ynew;

j++;

break;

}

case IN2OUT:

{

xnew = xa + (float)(ywmax-ya)\*(xb-xa)/(yb-ya);

ynew = ywmax;

xc\_tam[j]=xnew;

yc\_tam[j]=ynew;

j++;

break;

}

case OUTSIDE:

{

break;

}

}

printf("\n chieu = %d\n",chieu);

}

//gan lai danh sach canh moi

for(i=0;i<j;i++)

{

xc[i]=xc\_tam[i];

yc[i]=yc\_tam[i];

}

//gan so dinh moi sau khi rightclipper

pointsize=j;

for(i=0;i<j;i++)

{

printf("(%d,%d); ",xc[i],yc[i]);

}

printf("\n");

}

void topclipper()

{

int chieu;

int i;

int ywmin;

int xa;

int ya;

int xb;

int yb;

int j;

int xnew;

int ynew;

ywmin=wy[0];//dinh 1

j=0;

for(i=0;i<pointsize;i++)

{

xa=xc[i];

ya=yc[i];

xb=xc[i+1];

yb=yc[i+1];

chieu=clipBottom(ywmin,ya,yb);

switch (chieu)

{

case OUT2IN:

{

xnew = xa + (float)(ywmin-ya)\*(xb-xa)/(yb-ya);

ynew = ywmin;

xc\_tam[j]=xnew;

yc\_tam[j]=ynew;

j++;

xc\_tam[j]=xb;

yc\_tam[j]=yb;

j++;

break;

}

case INSIDE:

{

xnew = xb;

ynew = yb;

xc\_tam[j]=xnew;

yc\_tam[j]=ynew;

j++;

break;

}

case IN2OUT:

{

xnew = xa + (float)(ywmin-ya)\*(xb-xa)/(yb-ya);

ynew = ywmin;

xc\_tam[j]=xnew;

yc\_tam[j]=ynew;

j++;

break;

}

case OUTSIDE:

{

break;

}

}

printf("\n chieu = %d\n",chieu);

}

//gan lai danh sach canh moi

for(i=0;i<j;i++)

{

xc[i]=xc\_tam[i];

yc[i]=yc\_tam[i];

}

//gan so dinh moi sau khi rightclipper

pointsize=j;

for(i=0;i<j;i++)

{

printf("(%d,%d); ",xc[i],yc[i]);

}

printf("\n");

}

void draw\_object\_afterclip()

{

int i;

//ve cac canh cua da giac

setcolor(YELLOW);

setlinestyle(1,0,1);

for(i=0;i<pointsize;i++)

{

line(xc[i],yc[i],xc[(i+1)%pointsize],yc[(i+1)%pointsize]);

delay(1000);

}

}

//chuong trinh chinh

int main()

{

nhapdulieu();

initwindow(800,800);

draw\_object();

//gan them 1 dinh o cuoi cung chinh la dinh xuat phat

xc[pointsize]=xc[0];

yc[pointsize]=yc[0];

leftclipper();

//gan them 1 dinh o cuoi cung chinh la dinh xuat phat

xc[pointsize]=xc[0];

yc[pointsize]=yc[0];

rightclipper();

//gan them 1 dinh o cuoi cung chinh la dinh xuat phat

xc[pointsize]=xc[0];

yc[pointsize]=yc[0];

bottomclipper();

//gan them 1 dinh o cuoi cung chinh la dinh xuat phat

xc[pointsize]=xc[0];

yc[pointsize]=yc[0];

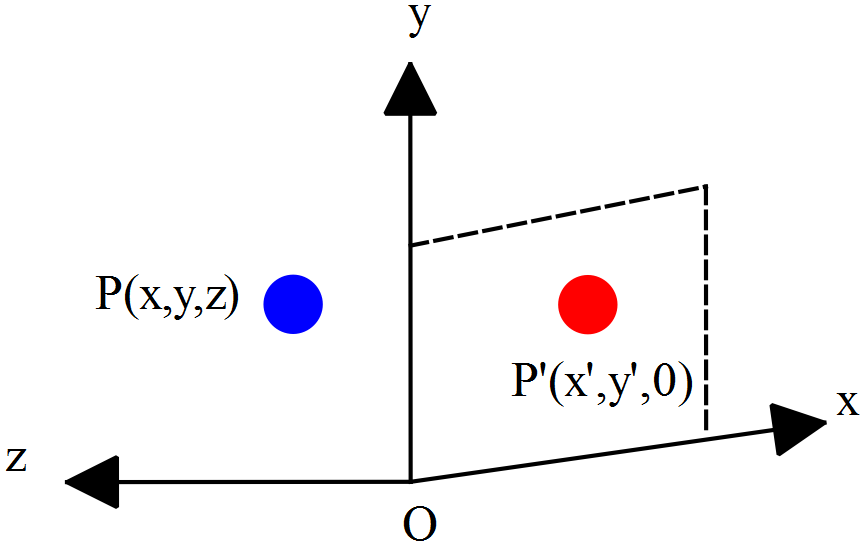
topclipper();

draw\_object\_afterclip();

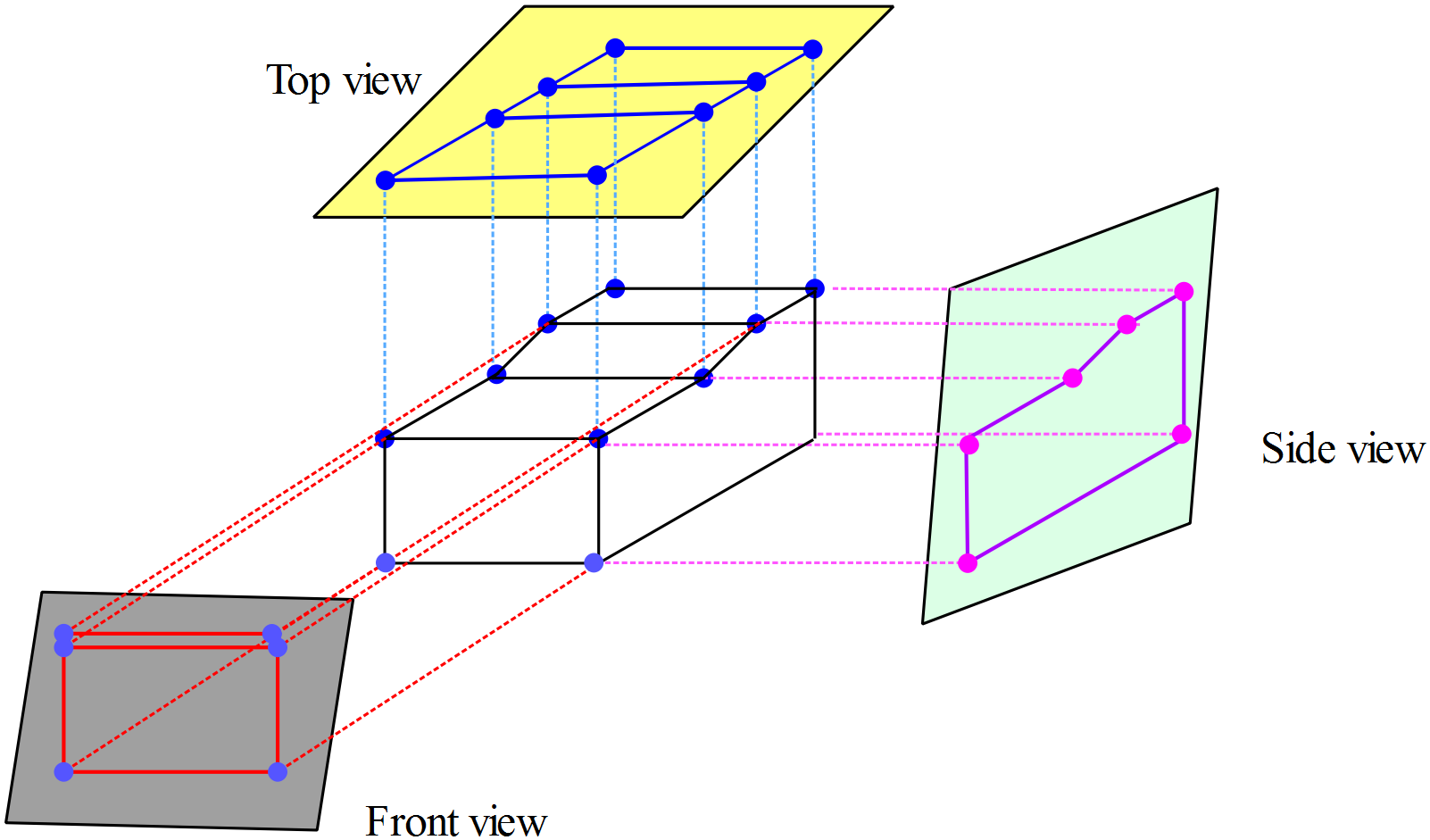
getch();

return 0;

}

**Phép chiếu trực giao** (Orthographic projection) là phép chiếu song song và tia chiếu vuông góc với mặt phẳng chiếu thường dùng mặt phẳng z=0

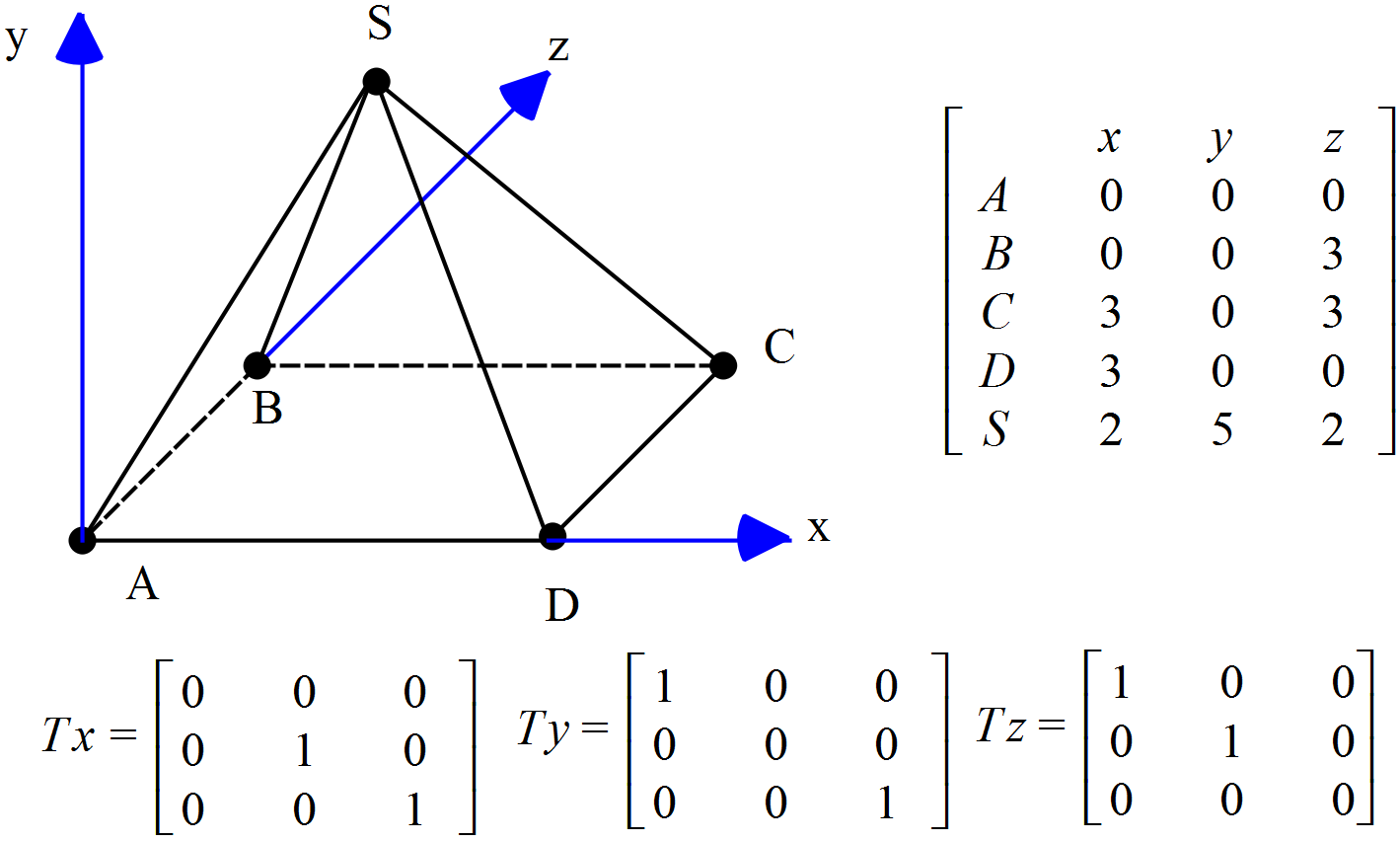
Ứng với mỗi mặt phẳng chiếu ta có 1 ma trận chiếu tương ứng

Các phép chiếu trực giao hầu như được dùng để tạo ra quang cảnh nhìn từ phía trước, bên sườn, và trên đỉnh của đối tượng.

* Quang cảnh phía trước, bên sườn, và phía sau của đối tượng được gọi là “mặt chiếu” (elevation).
* Quang cảnh phía trên được gọi là “mặt phẳng” (plane).

Các bản vẽ trong kỹ thuật thường dùng các phép chiếu trực giao này, vì các chiều dài và góc miêu tả chính xác và có thể đo được từ bản vẽ.

**Bài tập 1.**

Xác định các hình chiếu của đối tượng 3D lên các mặt phẳng chiếu (top, left, front)

Ta có:

* Front=Pchop\*Tz
* Left=Pchop\*Tx
* Top=Pchop\*Ty

A close up of smoke

Description automatically generatedA close up of a logo

Description automatically generatedA picture containing glasses, looking, watching, set

Description automatically generatedTa được:

Top:

[,1] [,2] [,3]

[1,] 0 0 0

[2,] 0 0 3

[3,] 3 0 3

[4,] 3 0 0

[5,] 2 0 2

Left:

[,1] [,2] [,3]

[1,] 0 0 0

[2,] 0 0 3

[3,] 0 0 3

[4,] 0 0 0

[5,] 0 5 2

Front:

[,1] [,2] [,3]

[1,] 0 0 0

[2,] 0 0 0

[3,] 3 0 0

[4,] 3 0 0

[5,] 2 5 0

Front:

[,1] [,2] [,3]

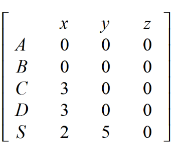
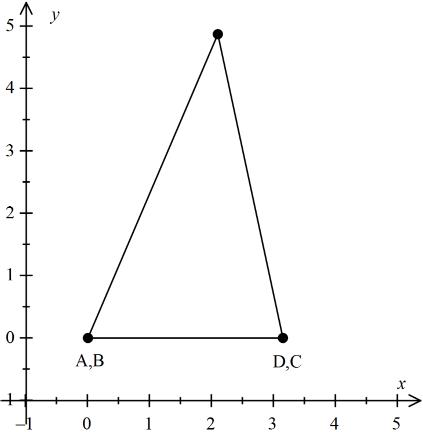
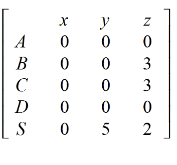
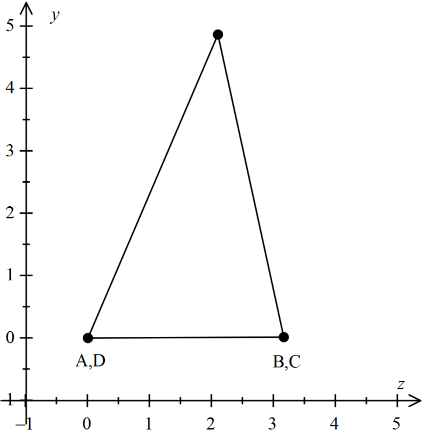
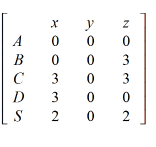
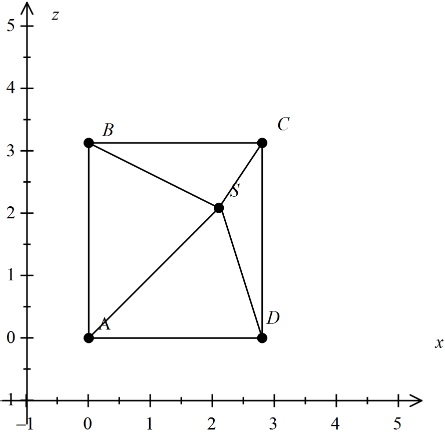
[1,] 0 0 0

[2,] 0 0 0

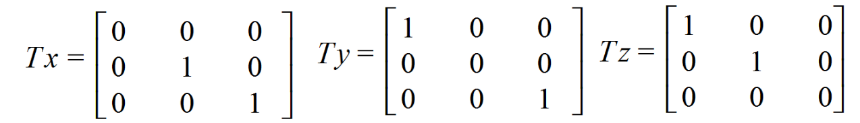
[3,] 3 0 0

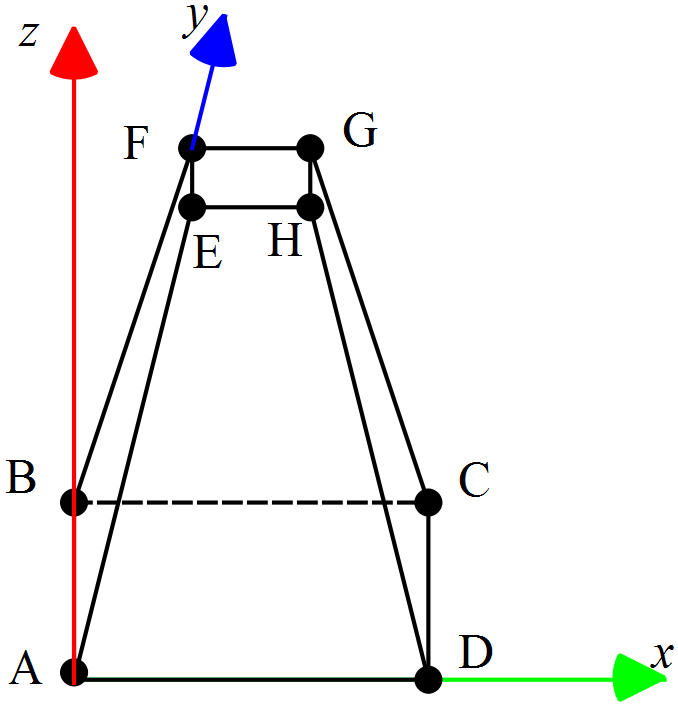
[4,] 3 0 0

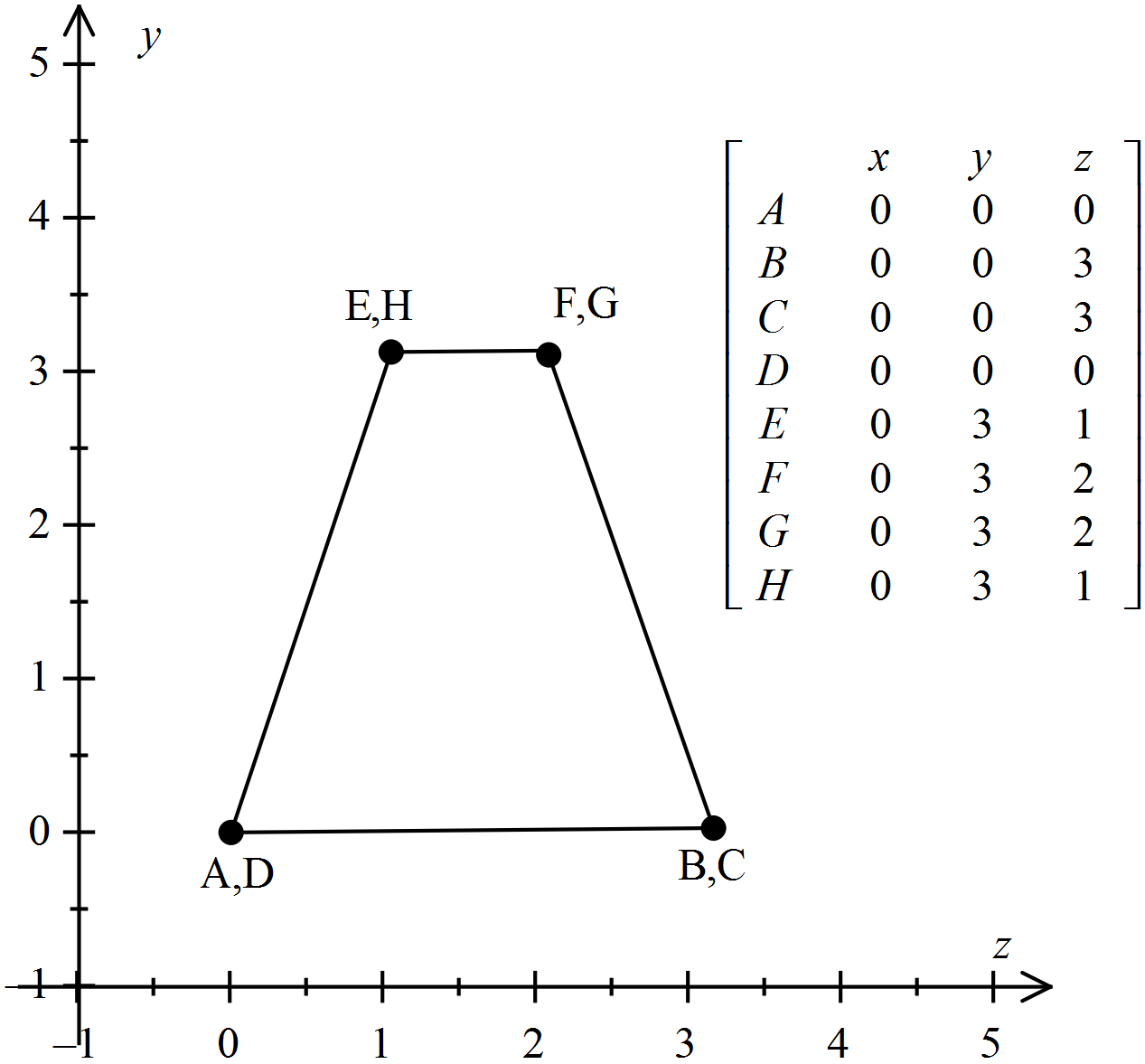
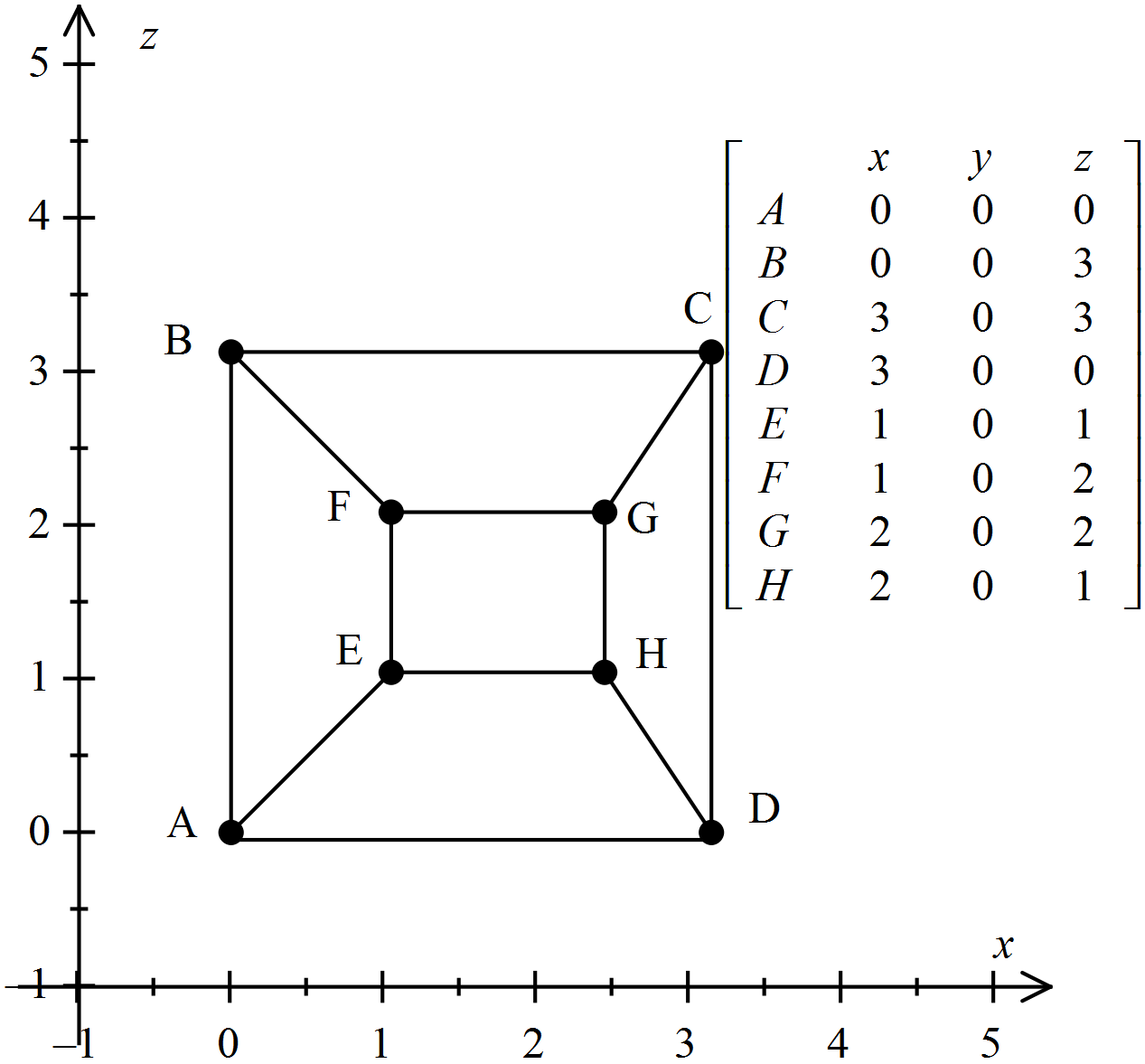
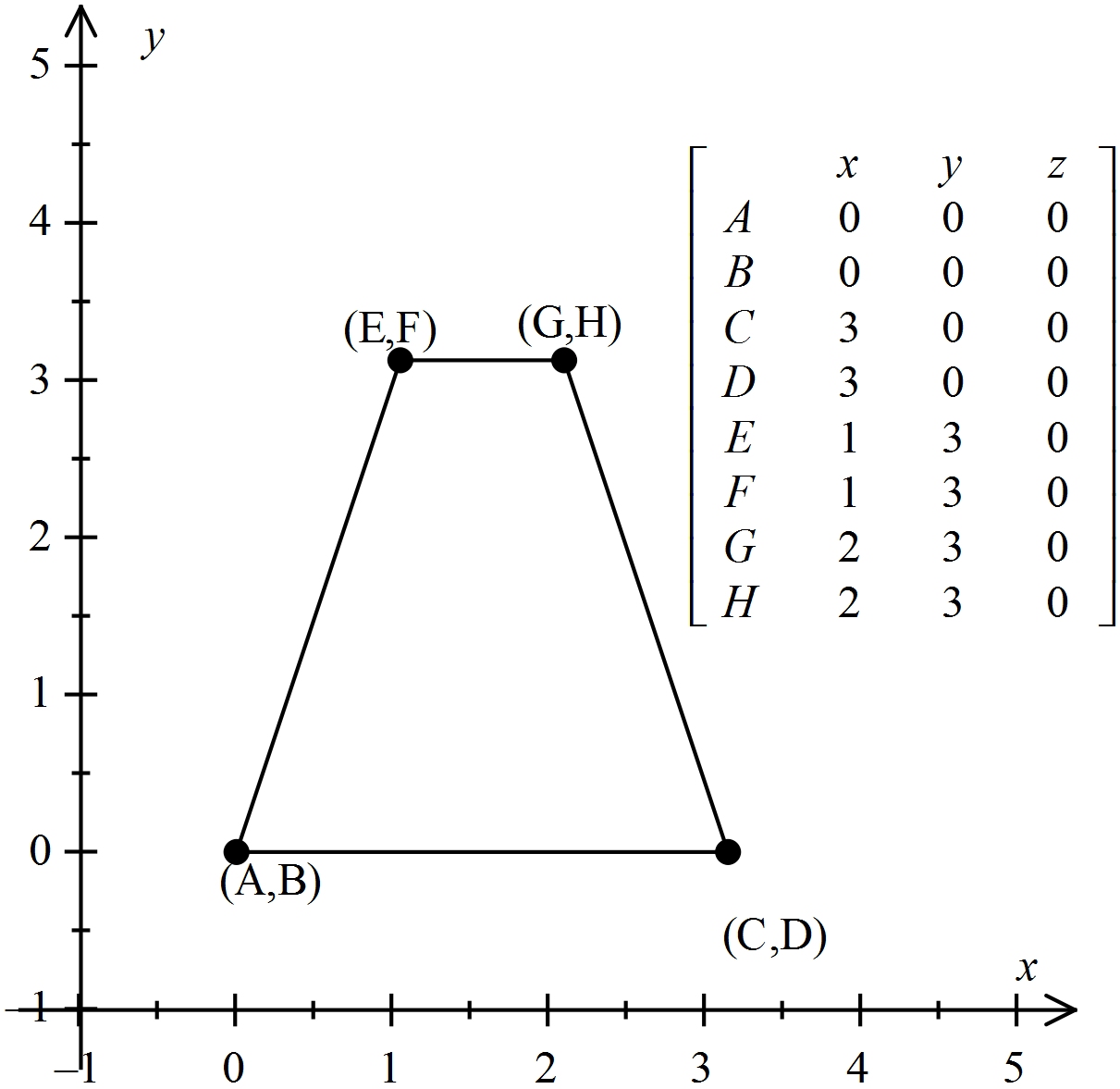
[5,] 2 5 0



**Bài tập 2.**

Xác định các hình chiếu của đối tượng 3D lên các mặt phẳng chiếu (top, left, front)





**Lập trình mô phỏng (chóp nhọn)**

//khai bao thu vien

#include<graphics.h>

//khai bao bien

//int X[10],Y[10],Z[10]; //toi da 10 dinh cua doi tuong 3D

int X[5]={100,100,300,300,200};

int Y[5]={100,100,100,100,500};

int Z[5]={100,300,300,100,200};

//int D[10][10]; //ma tran ke noi cac dinh

int D[5][5]=

{

{1,1,0,1,1},

{1,1,1,0,1},

{0,1,1,1,1},

{1,0,1,1,1},

{1,1,1,1,1},

};

int sodinh =5;

//chuong trinh con

void nhapdiem()

{

int i,j;

printf("Nhap so luong dinh: ");

scanf("%d",&sodinh);

for(i=0;i<sodinh;i++)

{

printf("X[%d]= ",i);

scanf("%d",&X[i]);

printf("Y[%d]= ",i);

scanf("%d",&Y[i]);

printf("Z[%d]= ",i);

scanf("%d",&Z[i]);

}

}

void intoadoadiem()

{

int i,j;

for(i=0;i<sodinh;i++)

{

printf("%d\t",X[i]);

}

printf("\n");

for(i=0;i<sodinh;i++)

{

printf("%d\t",Y[i]);

}

printf("\n");

for(i=0;i<sodinh;i++)

{

printf("%d\t",Z[i]);

}

}

void matranke()

{

int i,j;

for(i=0;i<sodinh;i++)

{

D[i][i]=1;

}

for(i=0;i<sodinh-1;i++)

{

for(j=i+1;j<sodinh;j++)

{

printf("D[%d,%d]= ",i,j);

scanf("%d",&D[i][j]);

}

}

for(i=0;i<sodinh;i++)

{

for(j=0;j<sodinh;j++)

{

if(i>j)

{

D[i][j]=D[j][i];

}

}

}

}

void inmatranke()

{

int i,j;

//in ra ma tran ke

printf("\nDanh sach ma tran ke:\n");

for(i=0;i<sodinh;i++)

{

for(j=0;j<sodinh;j++)

{

printf("%d\t",D[i][j]);

}

printf("\n");

}

}

void oxy()

{

int i,j;

//cai nay do minh dat ban dau, neu sau nay tu nhap tu ban phim thi thay bang so cu the

for(i=0;i<sodinh-1;i++)

{

for(j=i+1;j<sodinh;j++)

{

if(D[i][j]==1)

{

line(X[i],Y[i],X[j],Y[j]);

}

}

}

}

void oxz()

{

int i,j;

//cai nay do minh dat ban dau, neu sau nay tu nhap tu ban phim thi thay bang so cu the

for(i=0;i<sodinh-1;i++)

{

for(j=i+1;j<sodinh;j++)

{

if(D[i][j]==1)

{

line(X[i],Z[i],X[j],Z[j]);

}

}

}

}

void oyz()

{

int i,j;

//cai nay do minh dat ban dau, neu sau nay tu nhap tu ban phim thi thay bang so cu the

for(i=0;i<sodinh-1;i++)

{

for(j=i+1;j<sodinh;j++)

{

if(D[i][j]==1)

{

line(Z[i],Y[i],Z[j],Y[j]);

}

}

}

}

void ve3D()

{

outtextxy(0,0,"Hinh chieu tren mat phang XY");

setcolor(RED);

oxy();

getch();

cleardevice();

outtextxy(0,0,"Hinh chieu tren mat phang XZ");

setcolor(GREEN);

oxz();

getch();

cleardevice();

outtextxy(0,0,"Hinh chieu tren mat phang YZ");

setcolor(YELLOW);

oyz();

}

//chuong trinh chinh

int main()

{

//nhap toa do cac diem

//nhapdiem();

intoadoadiem();

//nhap ma tran ke

//matranke();

//in ra ma tran ke

inmatranke();

//khoi tao man hinh do hoa

initwindow(600,600);

ve3D();

getch();

}

**Lập trình mô phỏng (chóp cụt)**

//khai bao thu vien

#include<graphics.h>

//khai bao bien

//int X[10],Y[10],Z[10]; //toi da 10 dinh cua doi tuong 3D

int X[8]={100,100,400,400,200,200,300,300};

int Y[8]={100,100,100,100,400,400,400,400};

int Z[8]={100,400,400,100,200,300,300,200};

//int D[10][10]; //ma tran ke noi cac dinh

int D[8][8]=

{

//A,B,C,D,E,F,G,H

{1,1,0,1,1,0,0,0},//A

{1,1,1,0,0,1,0,0},//B

{0,1,1,1,0,0,1,0},//C

{1,0,1,1,0,0,0,1},//D

{1,0,0,0,1,1,0,1},//E

{0,1,0,0,1,1,1,0},//F

{0,0,1,0,0,1,1,1},//G

{0,0,0,1,1,0,1,1},//H

};

int sodinh =8;

//chuong trinh con

void nhapdiem()

{

int i,j;

printf("Nhap so luong dinh: ");

scanf("%d",&sodinh);

for(i=0;i<sodinh;i++)

{

printf("X[%d]= ",i);

scanf("%d",&X[i]);

printf("Y[%d]= ",i);

scanf("%d",&Y[i]);

printf("Z[%d]= ",i);

scanf("%d",&Z[i]);

}

}

void intoadoadiem()

{

int i,j;

for(i=0;i<sodinh;i++)

{

printf("%d\t",X[i]);

}

printf("\n");

for(i=0;i<sodinh;i++)

{

printf("%d\t",Y[i]);

}

printf("\n");

for(i=0;i<sodinh;i++)

{

printf("%d\t",Z[i]);

}

}

void matranke()

{

int i,j;

for(i=0;i<sodinh;i++)

{

D[i][i]=1;

}

for(i=0;i<sodinh-1;i++)

{

for(j=i+1;j<sodinh;j++)

{

printf("D[%d,%d]= ",i,j);

scanf("%d",&D[i][j]);

}

}

for(i=0;i<sodinh;i++)

{

for(j=0;j<sodinh;j++)

{

if(i>j)

{

D[i][j]=D[j][i];

}

}

}

}

void inmatranke()

{

int i,j;

//in ra ma tran ke

printf("\nDanh sach ma tran ke:\n");

for(i=0;i<sodinh;i++)

{

for(j=0;j<sodinh;j++)

{

printf("%d\t",D[i][j]);

}

printf("\n");

}

}

void oxy()

{

int i,j;

//cai nay do minh dat ban dau, neu sau nay tu nhap tu ban phim thi thay bang so cu the

for(i=0;i<sodinh-1;i++)

{

for(j=i+1;j<sodinh;j++)

{

if(D[i][j]==1)

{

line(X[i],Y[i],X[j],Y[j]);

}

}

}

}

void oxz()

{

int i,j;

//cai nay do minh dat ban dau, neu sau nay tu nhap tu ban phim thi thay bang so cu the

for(i=0;i<sodinh-1;i++)

{

for(j=i+1;j<sodinh;j++)

{

if(D[i][j]==1)

{

line(X[i],Z[i],X[j],Z[j]);

}

}

}

}

void oyz()

{

int i,j;

//cai nay do minh dat ban dau, neu sau nay tu nhap tu ban phim thi thay bang so cu the

for(i=0;i<sodinh-1;i++)

{

for(j=i+1;j<sodinh;j++)

{

if(D[i][j]==1)

{

line(Z[i],Y[i],Z[j],Y[j]);

}

}

}

}

void ve3D()

{

outtextxy(0,0,"Hinh chieu tren mat phang XY");

setcolor(RED);

oxy();

getch();

cleardevice();

outtextxy(0,0,"Hinh chieu tren mat phang XZ");

setcolor(GREEN);

oxz();

getch();

cleardevice();

outtextxy(0,0,"Hinh chieu tren mat phang YZ");

setcolor(YELLOW);

oyz();

}

//chuong trinh chinh

int main()

{

//nhap toa do cac diem

//nhapdiem();

intoadoadiem();

//nhap ma tran ke

//matranke();

//in ra ma tran ke

inmatranke();

//khoi tao man hinh do hoa

initwindow(600,600);

ve3D();

getch();

}

***Phép chiếu xiên (Oblique Projection)***

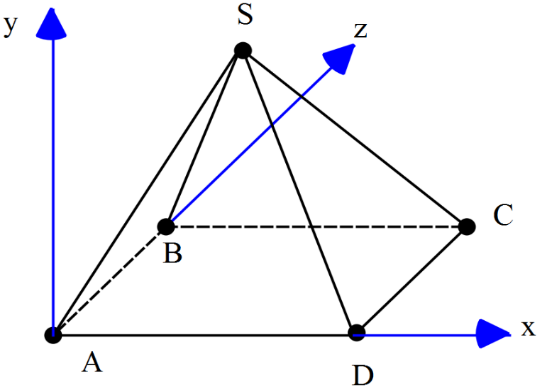
Một phép chiếu xiên đạt được bằng việc chiếu các điểm theo các đường thẳng song song, các đường thẳng này không vuông góc với mặt phẳng chiếu.

Hình chiếu xiên của điểm (x, y, z) theo một đường thẳng chiếu đến vị trí (x’, y’,0).

* Ta có: và
* và
* Suy ra:

Suy ra: L=z.L1

**Bài tập 1.**

Xác định hình chiếu của đối tượng 3D lên các mặt phẳng chiếu như sau:

A picture containing photo, black, red, drawing

Description automatically generated

Using R studio

R command

phi=30

L1=0.5

phi\_rad = 3.14\*phi/180;

Pchop=matrix(c(100,100,100,100,100,300,300,100,300,300,100,100,200,500,200),nrow=3,byrow=TRUE)

T = matrix(c(1,0, L1\*cos(phi\_rad),0,1, L1\*sin(phi\_rad),0,0,0),ncol=3,byrow=TRUE)

Pnew=round(T %\*%Pchop,0)

A star filled sky

Description automatically generatedA picture containing light, star, standing

Description automatically generated***Kết quả ta có (Pnew):***

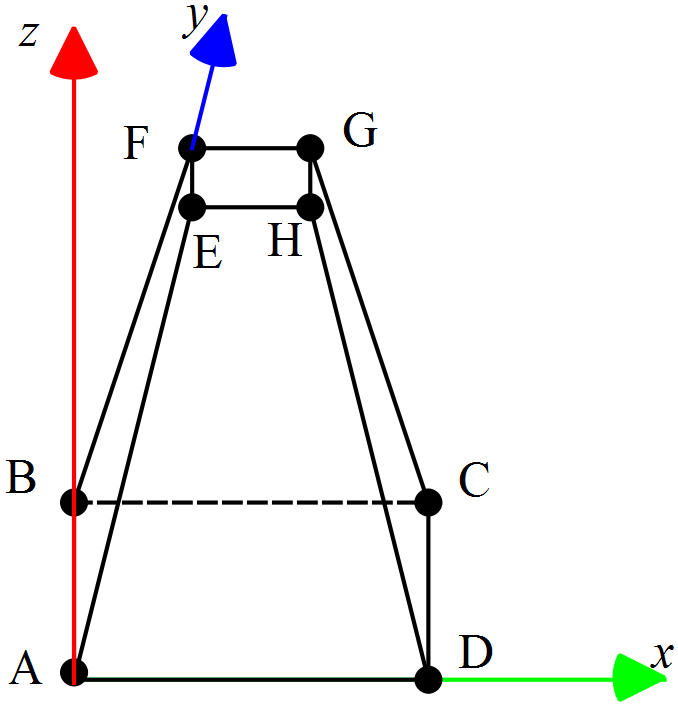
[,1] [,2] [,3] [,4] [,5]

[1,] 143 143 187 317 187

[2,] 325 325 150 425 350

[3,] 0 0 0 0 0

**Bài tập 2:**

Xác định các hình chiếu của đối tượng 3D lên các mặt phẳng chiếu (top, left, front)

A picture containing light

Description automatically generated

R command

phi=30

L1=0.5

phi\_rad = 3.14\*phi/180;

Pchop=matrix(c(0,0,0,0,0,3,3,0,3,3,0,0,2,5,2),nrow=5,byrow=TRUE)

T = matrix(c(1,0, L1\*cos(phi\_rad),0,1, L1\*sin(phi\_rad),0,0,0),ncol=3,byrow=TRUE)

Pnew=round(Pchop%T%)

A picture containing dark, standing, light

Description automatically generated [,1] [,2] [,3]

[1,] 0 0 0

[2,] 0 0 0

[3,] 3 0 1

[4,] 3 0 1

[5,] 2 5 2

**Lập trình mô phỏng (chóp nhọn)**

#include<stdio.h>

#include<math.h>

#include<graphics.h>

#define dinh 5

//Phep chieu cabinet. cho hinh hop chu nhat co dinh = 8

//khai bao bien

int phi;

double L1 = 0.5;

double phi\_rad;

int a[3][dinh]; //ma tran hinh hop chu nhat

int P[3][dinh]; //ma tran ket qua

float T[3][3]; //ma tran bien doi

int X[dinh];

int Y[dinh];

int m=3; //so hang cua ma tran toa do dinh

int p=5; //so cot cua ma tran toa do dinh

int n=3; //so hang = so cot cua ma tran bien doi T

//int D[10][10]; //ma tran ke noi cac dinh

int D[5][5]=

{

{1,1,0,1,1},

{1,1,1,0,1},

{0,1,1,1,1},

{1,0,1,1,1},

{1,1,1,1,1},

};

//chuong trinh con

void matranke()

{

int i,j;

for(i=0;i<dinh;i++)

{

D[i][i]=1;

}

for(i=0;i<dinh-1;i++)

{

for(j=i+1;j<dinh;j++)

{

printf("D[%d,%d]= ",i,j);

scanf("%d",&D[i][j]);

}

}

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

if(i>j)

{

D[i][j]=D[j][i];

}

}

}

}

void inmatranke()

{

int i,j;

//in ra ma tran ke

printf("\nDanh sach ma tran ke:\n");

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

printf("%d\t",D[i][j]);

}

printf("\n");

}

}

void nhaptoado()

{

//X Y Z

a[0][0] = 100; a[1][0] = 100; a[2][0] = 100;//A

a[0][1] = 100; a[1][1] = 100; a[2][1] = 300;//B

a[0][2] = 300; a[1][2] = 100; a[2][2] = 300;//C

a[0][3] = 300; a[1][3] = 100; a[2][3] = 100;//D

a[0][4] = 200; a[1][4] = 500; a[2][4] = 200;//S

printf("\nNhap he so goc phi = \n");

scanf("%d",&phi);

while(phi<0 || phi >180)

{

printf("\nNhap lai he so goc phi tu 0 - 180 do \n");

scanf("%d",&phi);

}

phi\_rad = (double) (3.14\*phi)/180;

}

void matranbiendoi()

{

T[0][0] = 1; T[0][1] = 0; T[0][2] = L1\*cos(phi\_rad); T[0][3] = 0;

T[1][0] = 0; T[1][1] = 1; T[1][2] = L1\*sin(phi\_rad); T[1][3] = 0;

T[2][0] = 0; T[2][1] = 0; T[2][2] = 0; T[2][3] = 0;

}

void cabinet()

{

int i,j,k;

//khoi tao ma tran ket qua

for(i=0; i<m; i++) //so hang

{

for(k=0; k<p; k++) //so cot

{

P[i][k] = 0;

}

}

//nhan 2 ma tran

for(i=0; i<m; i++) //so hang a

{

for(j=0; j<p; j++)//so cot a

{

for(k=0; k<n; k++)//so cot T

{

P[i][j] += (float)T[i][k]\* a[k][j];

}

}

}

}

void inketqua()

{

int i,j;

//in toa do cac dinh

printf("Toa do cac dinh cua hinh hop:\n");

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("\nGia tri cua ma tran cabinet\n");

for(i=0;i<m;i++)

{

for(j=0;j<n;j++)

{

printf("%.2f\t",T[i][j]);

}

printf("\n");

}

printf("\nToa do cac dinh cua hinh hop sau khi chieu:\n");

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d\t",P[i][j]);

}

printf("\n");

}

}

//in hinh chieu ra man hinh

void inhinhchieu()

{

int i,j;

for(i=0; i<dinh;i++)

{

X[i]=P[0][i];

Y[i]=P[1][i];

}

//in ra mang X va Y

for(i=0;i<dinh;i++)

{

printf("(%d,%d)\t",X[i],Y[i]);

}

}

void oxy()

{

int i,j;

//cai nay do minh dat ban dau, neu sau nay tu nhap tu ban phim thi thay bang so cu the

setcolor(RED);

for(i=0;i<dinh-1;i++)

{

for(j=i+1;j<dinh;j++)

{

if(D[i][j]==1)

{

line(X[i],Y[i],X[j],Y[j]);

delay(100);

}

}

}

}

//chuong trinh chinh

int main()

{

//nhap toa do cac dinh cua hinh hop

nhaptoado();

//nhap ma tran bien doi

matranbiendoi();

//nhap ma tran ke

//nhapmatranke()

//in ra ma tran ke

inmatranke();

//tinh toan ma tran ket qua

cabinet();

//in ket qua cac phep toan

inketqua();

//in ket qua cua phep chieu

inhinhchieu();

//khoi tao man hinh do hoa

initwindow(600,600);

oxy();

getch();

}

**Lập trình mô phỏng (chóp cụt)**

#include<stdio.h>

#include<math.h>

#include<graphics.h>

#define dinh 8

//Phep chieu cabinet. cho hinh hop chu nhat co dinh = 8

//khai bao bien

int phi;

double L1 = 0.5;

double phi\_rad;

int a[4][dinh]; //ma tran hinh hop chu nhat

int P[4][dinh]; //ma tran ket qua

float T[4][4]; //ma tran bien doi

int X[dinh];

int Y[dinh];

int m=3; //so hang cua ma tran toa do dinh

int p=8; //so cot cua ma tran toa do dinh

int n=3; //so hang = so cot cua ma tran bien doi T

//int D[10][10]; //ma tran ke noi cac dinh

int D[8][8]=

{

//A,B,C,D,E,F,G,H

{1,1,0,1,1,0,0,0},//A

{1,1,1,0,0,1,0,0},//B

{0,1,1,1,0,0,1,0},//C

{1,0,1,1,0,0,0,1},//D

{1,0,0,0,1,1,0,1},//E

{0,1,0,0,1,1,1,0},//F

{0,0,1,0,0,1,1,1},//G

{0,0,0,1,1,0,1,1},//H

};

//chuong trinh con

void matranke()

{

int i,j;

for(i=0;i<dinh;i++)

{

D[i][i]=1;

}

for(i=0;i<dinh-1;i++)

{

for(j=i+1;j<dinh;j++)

{

printf("D[%d,%d]= ",i,j);

scanf("%d",&D[i][j]);

}

}

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

if(i>j)

{

D[i][j]=D[j][i];

}

}

}

}

void inmatranke()

{

int i,j;

//in ra ma tran ke

printf("\nDanh sach ma tran ke:\n");

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

printf("%d\t",D[i][j]);

}

printf("\n");

}

}

void nhaptoado()

{

//X Y Z

a[0][0] = 100; a[1][0] = 100; a[2][0] = 100;//A

a[0][1] = 100; a[1][1] = 100; a[2][1] = 400;//B

a[0][2] = 400; a[1][2] = 100; a[2][2] = 400;//C

a[0][3] = 400; a[1][3] = 100; a[2][3] = 100;//D

a[0][4] = 200; a[1][4] = 400; a[2][4] = 200;//E

a[0][5] = 200; a[1][5] = 400; a[2][5] = 300;//F

a[0][6] = 300; a[1][6] = 400; a[2][6] = 300;//G

a[0][7] = 300; a[1][7] = 400; a[2][7] = 200;//H

printf("\nNhap he so goc phi = \n");

scanf("%d",&phi);

while(phi<0 || phi >180)

{

printf("\nNhap lai he so goc phi tu 0 - 180 do \n");

scanf("%d",&phi);

}

phi\_rad = (double) (3.14\*phi)/180;

}

void matranbiendoi()

{

T[0][0] = 1; T[0][1] = 0; T[0][2] = L1\*cos(phi\_rad); T[0][3] = 0;

T[1][0] = 0; T[1][1] = 1; T[1][2] = L1\*sin(phi\_rad); T[1][3] = 0;

T[2][0] = 0; T[2][1] = 0; T[2][2] = 0; T[2][3] = 0;

//T[3][0] = 0; T[3][1] = 0; T[3][2] = 0; T[3][3] = 1;

}

void cabinet()

{

int i,j,k;

//khoi tao ma tran ket qua

for(i=0; i<m; i++) //so hang

{

for(k=0; k<p; k++) //so cot

{

P[i][k] = 0;

}

}

//nhan 2 ma tran

for(i=0; i<m; i++) //so hang

{

for(k=0; k<p; k++)//so cot

{

for(j=0; j<n; j++)

{

P[i][k] += (float)T[i][j] \* a[j][k];

}

}

}

}

void inketqua()

{

int i,j;

//in toa do cac dinh

printf("Toa do cac dinh cua hinh hop:\n");

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("\nGia tri cua ma tran cabinet\n");

for(i=0;i<m;i++)

{

for(j=0;j<n;j++)

{

printf("%.2f\t",T[i][j]);

}

printf("\n");

}

printf("\nToa do cac dinh cua hinh hop sau khi chieu:\n");

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d\t",P[i][j]);

}

printf("\n");

}

}

//in hinh chieu ra man hinh

void inhinhchieu()

{

int i,j;

for(i=0; i<dinh;i++)

{

X[i]=P[0][i];

Y[i]=P[1][i];

}

//in ra mang X va Y

for(i=0;i<dinh;i++)

{

printf("(%d,%d)\t",X[i],Y[i]);

}

}

void oxy()

{

int i,j;

//cai nay do minh dat ban dau, neu sau nay tu nhap tu ban phim thi thay bang so cu the

setcolor(RED);

for(i=0;i<dinh-1;i++)

{

for(j=i+1;j<dinh;j++)

{

if(D[i][j]==1)

{

line(X[i],Y[i],X[j],Y[j]);

delay(500);

}

}

}

}

//chuong trinh chinh

int main()

{

//nhap toa do cac dinh cua hinh hop

nhaptoado();

//nhap ma tran bien doi

matranbiendoi();

//nhap ma tran ke

//nhapmatranke()

//in ra ma tran ke

inmatranke();

//tinh toan ma tran ket qua

cabinet();

//in ket qua cac phep toan

inketqua();

//in ket qua cua phep chieu

inhinhchieu();

//khoi tao man hinh do hoa

initwindow(600,600);

oxy();

getch();

}

**Lập trình mô phỏng (Hình hộp)**

#include<stdio.h>

#include<math.h>

#include<graphics.h>

#define dinh 8

//Phep chieu cabinet. cho hinh hop chu nhat co dinh = 8

//khai bao bien

int phi;

double L1 = 0.5;

double phi\_rad;

int a[4][dinh]; //ma tran hinh hop chu nhat

int P[4][dinh]; //ma tran ket qua

float T[4][4]; //ma tran bien doi

int X[dinh];

int Y[dinh];

int m=3; //so hang cua ma tran toa do dinh

int p=8; //so cot cua ma tran toa do dinh

int n=3; //so hang = so cot cua ma tran bien doi T

//int D[10][10]; //ma tran ke noi cac dinh

int D[8][8]=

{

{1,1,0,1,1,0,0,0},//A

{1,1,1,0,0,1,0,0},//B

{0,1,1,1,0,0,1,0},//C

{1,0,1,1,0,0,0,1},//D

{1,0,0,0,1,1,0,1},//E

{0,1,0,0,1,1,1,0},//F

{0,0,1,0,0,1,1,1},//G

{0,0,0,1,1,0,1,1}//H

};

//chuong trinh con

void matranke()

{

int i,j;

for(i=0;i<dinh;i++)

{

D[i][i]=1;

}

for(i=0;i<dinh-1;i++)

{

for(j=i+1;j<dinh;j++)

{

printf("D[%d,%d]= ",i,j);

scanf("%d",&D[i][j]);

}

}

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

if(i>j)

{

D[i][j]=D[j][i];

}

}

}

}

void inmatranke()

{

int i,j;

//in ra ma tran ke

printf("\nDanh sach ma tran ke:\n");

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

printf("%d\t",D[i][j]);

}

printf("\n");

}

}

void nhaptoado()

{

//X Y Z

a[0][0] = 100; a[1][0] = 100; a[2][0] = 100;//A

a[0][1] = 100; a[1][1] = 100; a[2][1] = 200;//B

a[0][2] = 200; a[1][2] = 100; a[2][2] = 200;//C

a[0][3] = 200; a[1][3] = 100; a[2][3] = 100;//D

a[0][4] = 100; a[1][4] = 200; a[2][4] = 100;//E

a[0][5] = 100; a[1][5] = 200; a[2][5] = 200;//F

a[0][6] = 200; a[1][6] = 200; a[2][6] = 200;//G

a[0][7] = 200; a[1][7] = 200; a[2][7] = 100;//H

printf("\nNhap he so goc phi = \n");

scanf("%d",&phi);

while(phi<0 || phi >180)

{

printf("\nNhap lai he so goc phi tu 0 - 180 do \n");

scanf("%d",&phi);

}

phi\_rad = (double) (3.14\*phi)/180;

}

void matranbiendoi()

{

T[0][0] = 1; T[0][1] = 0; T[0][2] = L1\*cos(phi\_rad); T[0][3] = 0;

T[1][0] = 0; T[1][1] = 1; T[1][2] = L1\*sin(phi\_rad); T[1][3] = 0;

T[2][0] = 0; T[2][1] = 0; T[2][2] = 0; T[2][3] = 0;

//T[3][0] = 0; T[3][1] = 0; T[3][2] = 0; T[3][3] = 1;

}

void cabinet()

{

int i,j,k;

//khoi tao ma tran ket qua

for(i=0; i<m; i++) //so hang

{

for(k=0; k<p; k++) //so cot

{

P[i][k] = 0;

}

}

//nhan 2 ma tran

for(i=0; i<m; i++) //so hang

{

for(k=0; k<p; k++)//so cot

{

for(j=0; j<n; j++)

{

P[i][k] += (float)T[i][j] \* a[j][k];

}

}

}

}

void inketqua()

{

int i,j;

//in toa do cac dinh

printf("Toa do cac dinh cua hinh hop:\n");

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("\nGia tri cua ma tran cabinet\n");

for(i=0;i<m;i++)

{

for(j=0;j<n;j++)

{

printf("%.2f\t",T[i][j]);

}

printf("\n");

}

printf("\nToa do cac dinh cua hinh hop sau khi chieu:\n");

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d\t",P[i][j]);

}

printf("\n");

}

}

//in hinh chieu ra man hinh

void inhinhchieu()

{

int i,j;

for(i=0; i<dinh;i++)

{

X[i]=P[0][i];

Y[i]=P[1][i];

}

//in ra mang X va Y

for(i=0;i<dinh;i++)

{

printf("(%d,%d)\t",X[i],Y[i]);

}

}

void oxy()

{

int i,j;

//cai nay do minh dat ban dau, neu sau nay tu nhap tu ban phim thi thay bang so cu the

setcolor(RED);

for(i=0;i<dinh-1;i++)

{

for(j=i+1;j<dinh;j++)

{

if(D[i][j]==1)

{

line(X[i],Y[i],X[j],Y[j]);

delay(100);

}

}

}

}

//chuong trinh chinh

int main()

{

//nhap toa do cac dinh cua hinh hop

nhaptoado();

//nhap ma tran bien doi

matranbiendoi();

//nhap ma tran ke

//nhapmatranke()

//in ra ma tran ke

inmatranke();

//tinh toan ma tran ket qua

cabinet();

//in ket qua cac phep toan

inketqua();

//in ket qua cua phep chieu

inhinhchieu();

//khoi tao man hinh do hoa

initwindow(600,600);

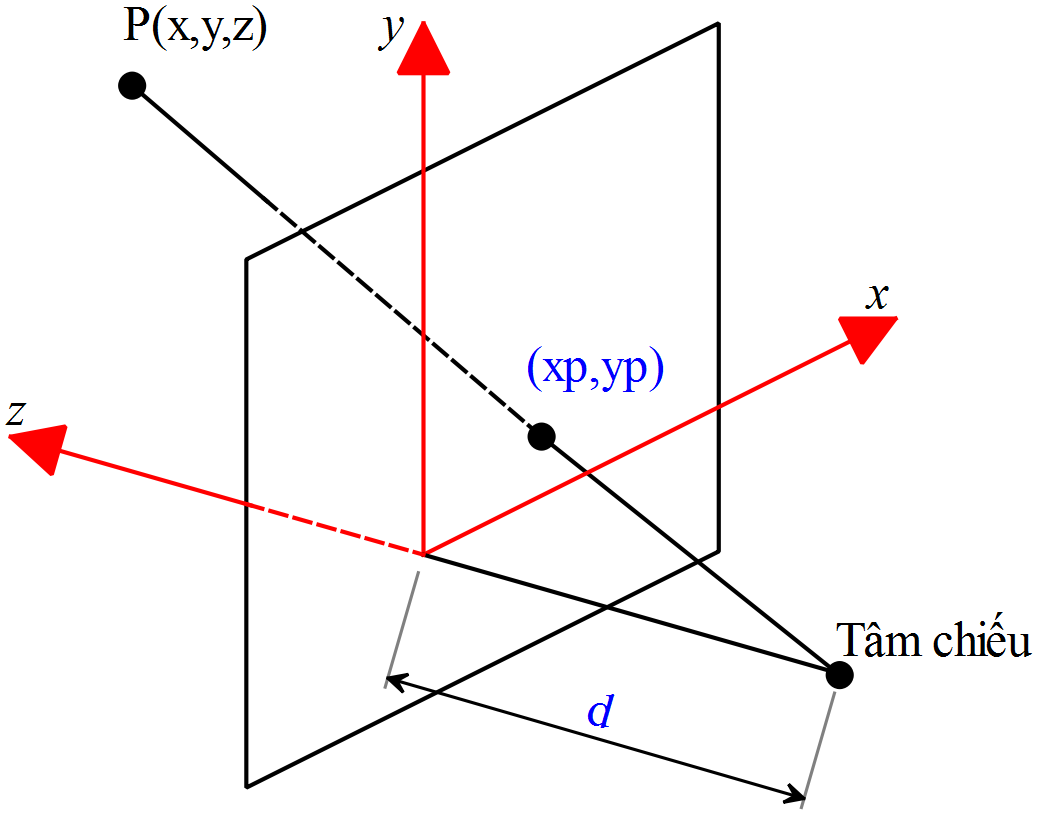
oxy();

getch();

}

**Phép chiếu phối cảnh (Perspective Projection)**

*Phép chiếu phối cảnh* là phép chiếu mà các tia chiếu không song song với nhau mà xuất phát từ một điểm gọi là tâm chiếu.

Phép chiếu phối cảnh tạo ra hiệu ứng về luật xa gần tạo cảm giác về độ sâu của đối tượng trong thế giới thật mà phép chiếu song song không lột tả được.

Các đoạn thẳng song song của mô hình 3D sau phép chiếu hội tụ tại một điểm gọi là điểm triệt tiêu (vanishing point).

*Phân loại phép chiếu phối cảnh* dựa vào tâm chiếu (*Centre Of Projection -*COP) và mặt phẳng chiếu (projection plane)

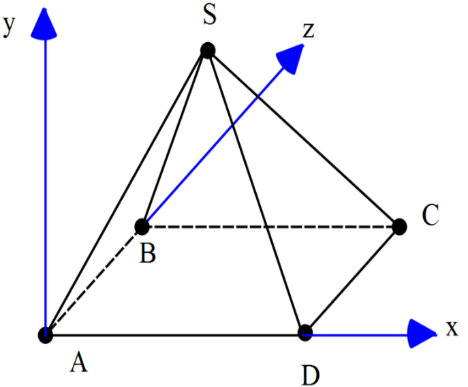
Để đạt được phép chiếu phối cảnh của đối tượng ba chiều, chúng ta chiếu các điểm theo đường thẳng chiếu để các đường này gặp nhau ở tâm chiếu.

Ma trận biến đổi một điểm phối cảnh [ Tr ] có dạng:

**Ví dụ:** Cho đối tượng 3D là hình chop nhọn có tọa độ như hình.

*Tâm chiếu có tọa độ (xp,yp,zp) là (50,50,100)*

A picture containing sitting, light, red, star

Description automatically generated*Hãy xác định các điểm trên mặt phẳng chiếu.*

//X Y Z

a[0][0] = 100; a[1][0] = 100; a[2][0] = 100;//A

a[0][1] = 100; a[1][1] = 100; a[2][1] = 300;//B

a[0][2] = 300; a[1][2] = 100; a[2][2] = 300;//C

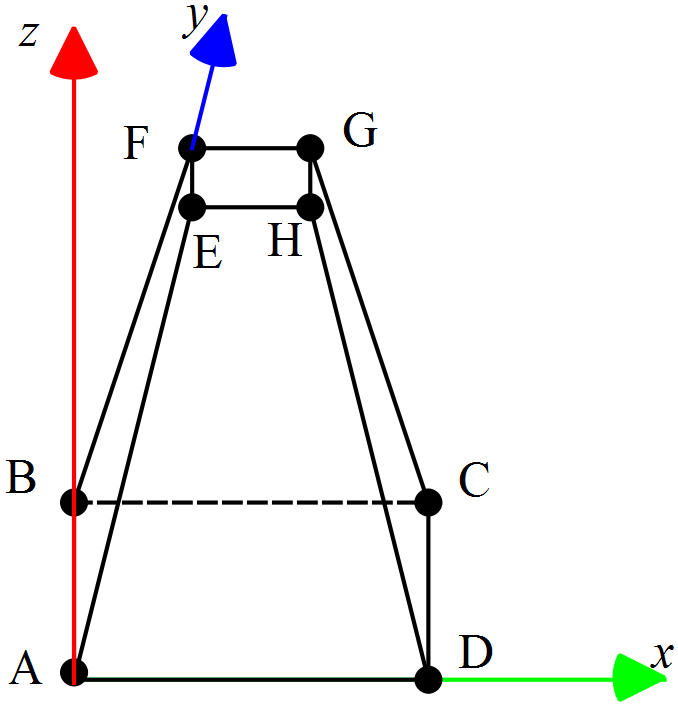
a[0][3] = 300; a[1][3] = 100; a[2][3] = 100;//D

a[0][4] = 200; a[1][4] = 500; a[2][4] = 200;//S

**Ví dụ 2:**

Cho đối tượng 3D là hình chop cụt có tọa độ như hình.

*Tâm chiếu có tọa độ (xp,yp,zp) là (50,50,100)*

*Hãy xác định các điểm trên mặt phẳng chiếu.*

//X Y Z

a[0][0] = 100; a[1][0] = 100; a[2][0] = 100;//A

a[0][1] = 100; a[1][1] = 100; a[2][1] = 400;//B

a[0][2] = 400; a[1][2] = 100; a[2][2] = 400;//C

a[0][3] = 400; a[1][3] = 100; a[2][3] = 100;//D

a[0][4] = 200; a[1][4] = 400; a[2][4] = 200;//E

a[0][5] = 200; a[1][5] = 400; a[2][5] = 300;//F

a[0][6] = 300; a[1][6] = 400; a[2][6] = 300;//G

a[0][7] = 300; a[1][7] = 400; a[2][7] = 200;//H

A picture containing light

Description automatically generated

**Lập trình mô phỏng (Hình chóp nhọn)**

#include<stdio.h>

#include<math.h>

#include<graphics.h>

#define dinh 5

//Phep chieu cabinet. cho hinh hop chu nhat co dinh = 8

//khai bao bien

int xp, yp, zp; //tam chieu

int a[4][dinh]; //ma tran hinh hop chu nhat

int P[4][dinh]; //ma tran ket qua

int X[dinh];

int Y[dinh];

int m=5;//so dinh

int p=5;

//int D[10][10]; //ma tran ke noi cac dinh

int D[5][5]=

{

{1,1,0,1,1},

{1,1,1,0,1},

{0,1,1,1,1},

{1,0,1,1,1},

{1,1,1,1,1},

};

//chuong trinh con

void matranke()

{

int i,j;

for(i=0;i<dinh;i++)

{

D[i][i]=1;

}

for(i=0;i<dinh-1;i++)

{

for(j=i+1;j<dinh;j++)

{

printf("D[%d,%d]= ",i,j);

scanf("%d",&D[i][j]);

}

}

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

if(i>j)

{

D[i][j]=D[j][i];

}

}

}

}

void inmatranke()

{

int i,j;

//in ra ma tran ke

printf("\nDanh sach ma tran ke:\n");

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

printf("%d\t",D[i][j]);

}

printf("\n");

}

}

void nhaptoado()

{

//X Y Z

a[0][0] = 100; a[1][0] = 100; a[2][0] = 100;//A

a[0][1] = 100; a[1][1] = 100; a[2][1] = 300;//B

a[0][2] = 300; a[1][2] = 100; a[2][2] = 300;//C

a[0][3] = 300; a[1][3] = 100; a[2][3] = 100;//D

a[0][4] = 200; a[1][4] = 500; a[2][4] = 200;//S

printf("\nNhap toa do tam chieu (xp,yp,zp)=\n");

printf("xp = ");

scanf("%d",&xp);

printf("yp = ");

scanf("%d",&yp);

printf("zp = ");

scanf("%d",&zp);

//xp = 300; yp = 320; zp = 100;

}

void perspective\_1tam()

{

int i,j;

int x;

int y;

int z;

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

P[i][j]=0;

}

}

//tinh toan gia tri cac dinh tren man chieu

for(i=0;i<dinh;i++)

{

x=a[0][i];

y=a[1][i];

z=a[2][i];

P[0][i]= xp+(float)(x+xp)/(z+zp)\*zp; //Toa do x'

P[1][i]= yp+(float)(y+yp)/(z+zp)\*zp;//Toa do y'

//P[0][i]= (float)(x\*zp)/(z+zp); //Toa do x'

//P[1][i]= (float)(y\*zp)/(z+zp);//Toa do y'

}

}

void inketqua()

{

int i,j;

//in toa do cac dinh

printf("Toa do cac dinh cua hinh hop:\n");

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("\nToa do cac dinh cua hinh hop sau khi chieu:\n");

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d\t",P[i][j]);

}

printf("\n");

}

}

//in hinh chieu ra man hinh

void inhinhchieu()

{

int i,j;

for(i=0; i<dinh;i++)

{

X[i]=P[0][i];

Y[i]=P[1][i];

}

//in ra mang X va Y

for(i=0;i<dinh;i++)

{

printf("(%d,%d)\t",X[i],Y[i]);

}

}

void oxy()

{

int i,j;

//cai nay do minh dat ban dau, neu sau nay tu nhap tu ban phim thi thay bang so cu the

setcolor(RED);

for(i=0;i<dinh-1;i++)

{

for(j=i+1;j<dinh;j++)

{

if(D[i][j]==1)

{

line(X[i],Y[i],X[j],Y[j]);

delay(100);

}

}

}

}

//chuong trinh chinh

int main()

{

//nhap toa do cac dinh cua hinh hop

nhaptoado();

//nhap ma tran ke

//nhapmatranke()

//in ra ma tran ke

inmatranke();

//tinh toan ma tran ket qua

perspective\_1tam();

//in ket qua cac phep toan

inketqua();

//in ket qua cua phep chieu

inhinhchieu();

//khoi tao man hinh do hoa

initwindow(600,600);

oxy();

getch();

}

**Lập trình mô phỏng (Chóp cụt)**

#include<stdio.h>

#include<math.h>

#include<graphics.h>

#define dinh 8

//Phep chieu cabinet. cho hinh hop chu nhat co dinh = 8

//khai bao bien

int xp, yp, zp; //tam chieu

int a[4][dinh]; //ma tran hinh hop chu nhat

int P[4][dinh]; //ma tran ket qua

int X[dinh];

int Y[dinh];

int m=8;//so dinh

int p=8;

//int D[10][10]; //ma tran ke noi cac dinh

int D[8][8]=

{

//A,B,C,D,E,F,G,H

{1,1,0,1,1,0,0,0},//A

{1,1,1,0,0,1,0,0},//B

{0,1,1,1,0,0,1,0},//C

{1,0,1,1,0,0,0,1},//D

{1,0,0,0,1,1,0,1},//E

{0,1,0,0,1,1,1,0},//F

{0,0,1,0,0,1,1,1},//G

{0,0,0,1,1,0,1,1},//H

};

//chuong trinh con

void matranke()

{

int i,j;

for(i=0;i<dinh;i++)

{

D[i][i]=1;

}

for(i=0;i<dinh-1;i++)

{

for(j=i+1;j<dinh;j++)

{

printf("D[%d,%d]= ",i,j);

scanf("%d",&D[i][j]);

}

}

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

if(i>j)

{

D[i][j]=D[j][i];

}

}

}

}

void inmatranke()

{

int i,j;

//in ra ma tran ke

printf("\nDanh sach ma tran ke:\n");

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

printf("%d\t",D[i][j]);

}

printf("\n");

}

}

void nhaptoado()

{

//X Y Z

a[0][0] = 100; a[1][0] = 100; a[2][0] = 100;//A

a[0][1] = 100; a[1][1] = 100; a[2][1] = 400;//B

a[0][2] = 400; a[1][2] = 100; a[2][2] = 400;//C

a[0][3] = 400; a[1][3] = 100; a[2][3] = 100;//D

a[0][4] = 200; a[1][4] = 400; a[2][4] = 200;//E

a[0][5] = 200; a[1][5] = 400; a[2][5] = 300;//F

a[0][6] = 300; a[1][6] = 400; a[2][6] = 300;//G

a[0][7] = 300; a[1][7] = 400; a[2][7] = 200;//H

printf("\nNhap toa do tam chieu (xp,yp,zp)=\n");

printf("xp = ");

scanf("%d",&xp);

printf("yp = ");

scanf("%d",&yp);

printf("zp = ");

scanf("%d",&zp);

//xp = 300; yp = 320; zp = 100;

}

void perspective\_1tam()

{

int i,j;

int x;

int y;

int z;

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

P[i][j]=0;

}

}

//tinh toan gia tri cac dinh tren man chieu

for(i=0;i<dinh;i++)

{

x=a[0][i];

y=a[1][i];

z=a[2][i];

P[0][i]= xp+(float)(x+xp)/(z+zp)\*zp; //Toa do x'

P[1][i]= yp+(float)(y+yp)/(z+zp)\*zp;//Toa do y'

//P[0][i]= (float)(x\*zp)/(z+zp); //Toa do x'

//P[1][i]= (float)(y\*zp)/(z+zp);//Toa do y'

}

}

void inketqua()

{

int i,j;

//in toa do cac dinh

printf("Toa do cac dinh cua hinh hop:\n");

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("\nToa do cac dinh cua hinh hop sau khi chieu:\n");

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d\t",P[i][j]);

}

printf("\n");

}

}

//in hinh chieu ra man hinh

void inhinhchieu()

{

int i,j;

for(i=0; i<dinh;i++)

{

X[i]=P[0][i];

Y[i]=P[1][i];

}

//in ra mang X va Y

for(i=0;i<dinh;i++)

{

printf("(%d,%d)\t",X[i],Y[i]);

}

}

void oxy()

{

int i,j;

//cai nay do minh dat ban dau, neu sau nay tu nhap tu ban phim thi thay bang so cu the

setcolor(GREEN);

for(i=0;i<dinh-1;i++)

{

for(j=i+1;j<dinh;j++)

{

if(D[i][j]==1)

{

line(X[i],Y[i],X[j],Y[j]);

delay(100);

}

}

}

}

//chuong trinh chinh

int main()

{

//nhap toa do cac dinh cua hinh hop

nhaptoado();

//nhap ma tran ke

//nhapmatranke()

//in ra ma tran ke

inmatranke();

//tinh toan ma tran ket qua

perspective\_1tam();

//in ket qua cac phep toan

inketqua();

//in ket qua cua phep chieu

inhinhchieu();

//khoi tao man hinh do hoa

initwindow(600,600);

oxy();

getch();

}

**Lập trình mô phỏng (Hình hộp)**

#include<stdio.h>

#include<math.h>

#include<graphics.h>

#define dinh 8

//Phep chieu cabinet. cho hinh hop chu nhat co dinh = 8

//khai bao bien

int xp, yp, zp; //tam chieu

int a[4][dinh]; //ma tran hinh hop chu nhat

int P[4][dinh]; //ma tran ket qua

int X[dinh];

int Y[dinh];

int m=8;//so dinh

int p=8;

//int D[10][10]; //ma tran ke noi cac dinh

int D[8][8]=

{

{1,1,0,1,1,0,0,0},//A

{1,1,1,0,0,1,0,0},//B

{0,1,1,1,0,0,1,0},//C

{1,0,1,1,0,0,0,1},//D

{1,0,0,0,1,1,0,1},//E

{0,1,0,0,1,1,1,0},//F

{0,0,1,0,0,1,1,1},//G

{0,0,0,1,1,0,1,1}//H

};

//chuong trinh con

void matranke()

{

int i,j;

for(i=0;i<dinh;i++)

{

D[i][i]=1;

}

for(i=0;i<dinh-1;i++)

{

for(j=i+1;j<dinh;j++)

{

printf("D[%d,%d]= ",i,j);

scanf("%d",&D[i][j]);

}

}

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

if(i>j)

{

D[i][j]=D[j][i];

}

}

}

}

void inmatranke()

{

int i,j;

//in ra ma tran ke

printf("\nDanh sach ma tran ke:\n");

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

printf("%d\t",D[i][j]);

}

printf("\n");

}

}

void nhaptoado()

{

//X Y Z

a[0][0] = 100; a[1][0] = 100; a[2][0] = 100;//A

a[0][1] = 100; a[1][1] = 100; a[2][1] = 200;//B

a[0][2] = 200; a[1][2] = 100; a[2][2] = 200;//C

a[0][3] = 200; a[1][3] = 100; a[2][3] = 100;//D

a[0][4] = 100; a[1][4] = 200; a[2][4] = 100;//E

a[0][5] = 100; a[1][5] = 200; a[2][5] = 200;//F

a[0][6] = 200; a[1][6] = 200; a[2][6] = 200;//G

a[0][7] = 200; a[1][7] = 200; a[2][7] = 100;//H

printf("\nNhap toa do tam chieu (xp,yp,zp)=\n");

printf("xp = ");

scanf("%d",&xp);

printf("yp = ");

scanf("%d",&yp);

printf("zp = ");

scanf("%d",&zp);

//xp = 300; yp = 320; zp = 100;

}

void perspective\_1tam()

{

int i,j;

int x;

int y;

int z;

for(i=0;i<dinh;i++)

{

for(j=0;j<dinh;j++)

{

P[i][j]=0;

}

}

//tinh toan gia tri cac dinh tren man chieu

for(i=0;i<dinh;i++)

{

x=a[0][i];

y=a[1][i];

z=a[2][i];

P[0][i]= xp+(float)(x+xp)/(z+zp)\*zp; //Toa do x'

P[1][i]= yp+(float)(y+yp)/(z+zp)\*zp;//Toa do y'

//P[0][i]= (float)(x\*zp)/(z+zp); //Toa do x'

//P[1][i]= (float)(y\*zp)/(z+zp);//Toa do y'

}

}

void inketqua()

{

int i,j;

//in toa do cac dinh

printf("Toa do cac dinh cua hinh hop:\n");

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("\nToa do cac dinh cua hinh hop sau khi chieu:\n");

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d\t",P[i][j]);

}

printf("\n");

}

}

//in hinh chieu ra man hinh

void inhinhchieu()

{

int i,j;

for(i=0; i<dinh;i++)

{

X[i]=P[0][i];

Y[i]=P[1][i];

}

//in ra mang X va Y

for(i=0;i<dinh;i++)

{

printf("(%d,%d)\t",X[i],Y[i]);

}

}

void oxy()

{

int i,j;

//cai nay do minh dat ban dau, neu sau nay tu nhap tu ban phim thi thay bang so cu the

setcolor(RED);

for(i=0;i<dinh-1;i++)

{

for(j=i+1;j<dinh;j++)

{

if(D[i][j]==1)

{

line(X[i],Y[i],X[j],Y[j]);

delay(100);

}

}

}

}

//chuong trinh chinh

int main()

{

//nhap toa do cac dinh cua hinh hop

nhaptoado();

//nhap ma tran ke

//nhapmatranke()

//in ra ma tran ke

inmatranke();

//tinh toan ma tran ket qua

perspective\_1tam();

//in ket qua cac phep toan

inketqua();

//in ket qua cua phep chieu

inhinhchieu();

//khoi tao man hinh do hoa

initwindow(600,600);

oxy();

getch();

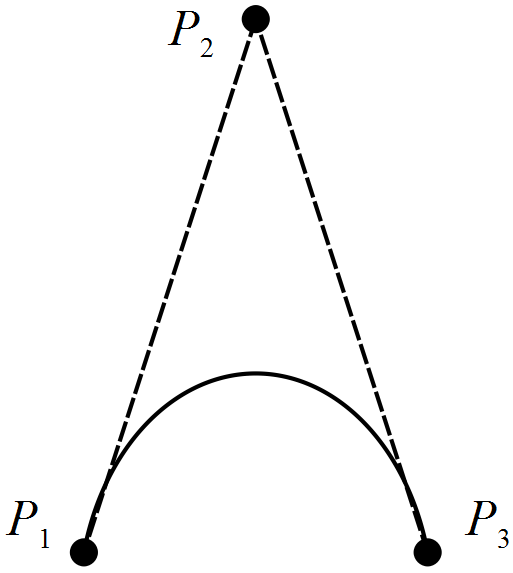
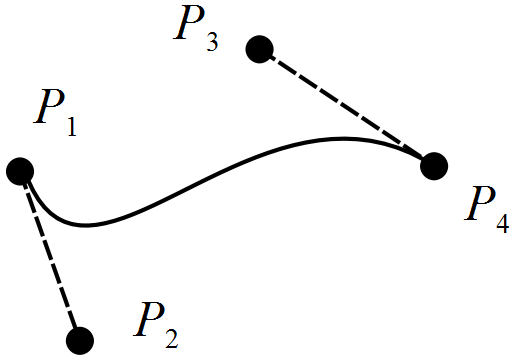
}

Bezier đã sử dụng các điểm kiểm soát cho đường cong tại những đỉnh của đa giác và tiếp  
tuyến tại đó (p0,p1,p2,p3,…).

Với 2 điểm kiểm soát (p1,p2):

Với 3 điểm kiểm soát (p1,p2,p3):

Với 4 điểm kiểm soát (p1,p2,p3,p4)



Biểu thức biểu diễn đường cong Bezier:

Trong đó,

Với

i, các điểm kiểm soát

Cụ thể:

Sử dụng tam giác Pascal để đối chiếu

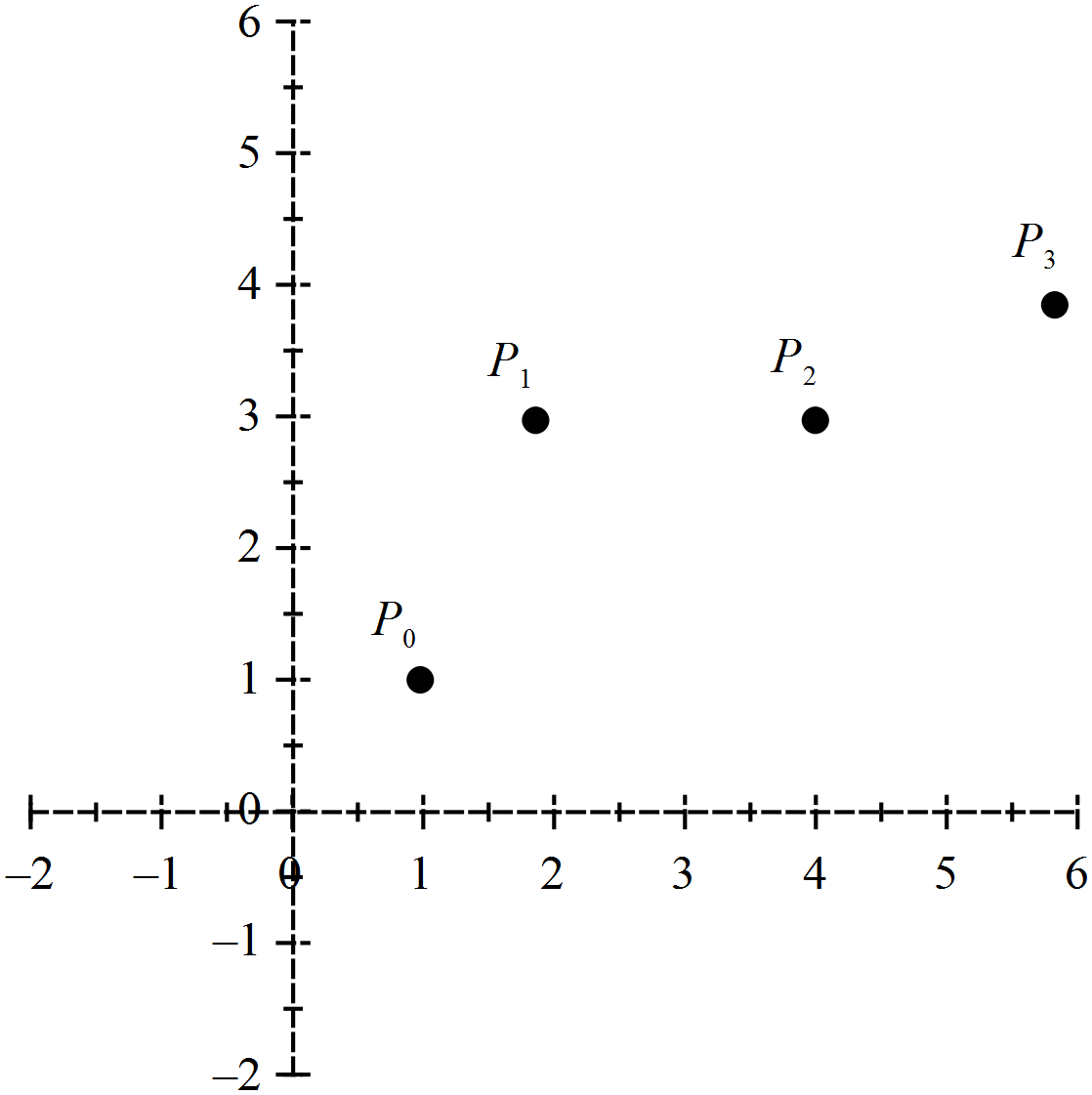
[C-programming/Chuong1\_Bai19\_TamgiacPascal.c at master · thinhdoanvu/C-programming (github.com)](https://github.com/thinhdoanvu/C-programming/blob/master/Chuong1_Cautruc/Code/Chuong1_Bai19_TamgiacPascal.c)

A screenshot of a social media post

Description automatically generated

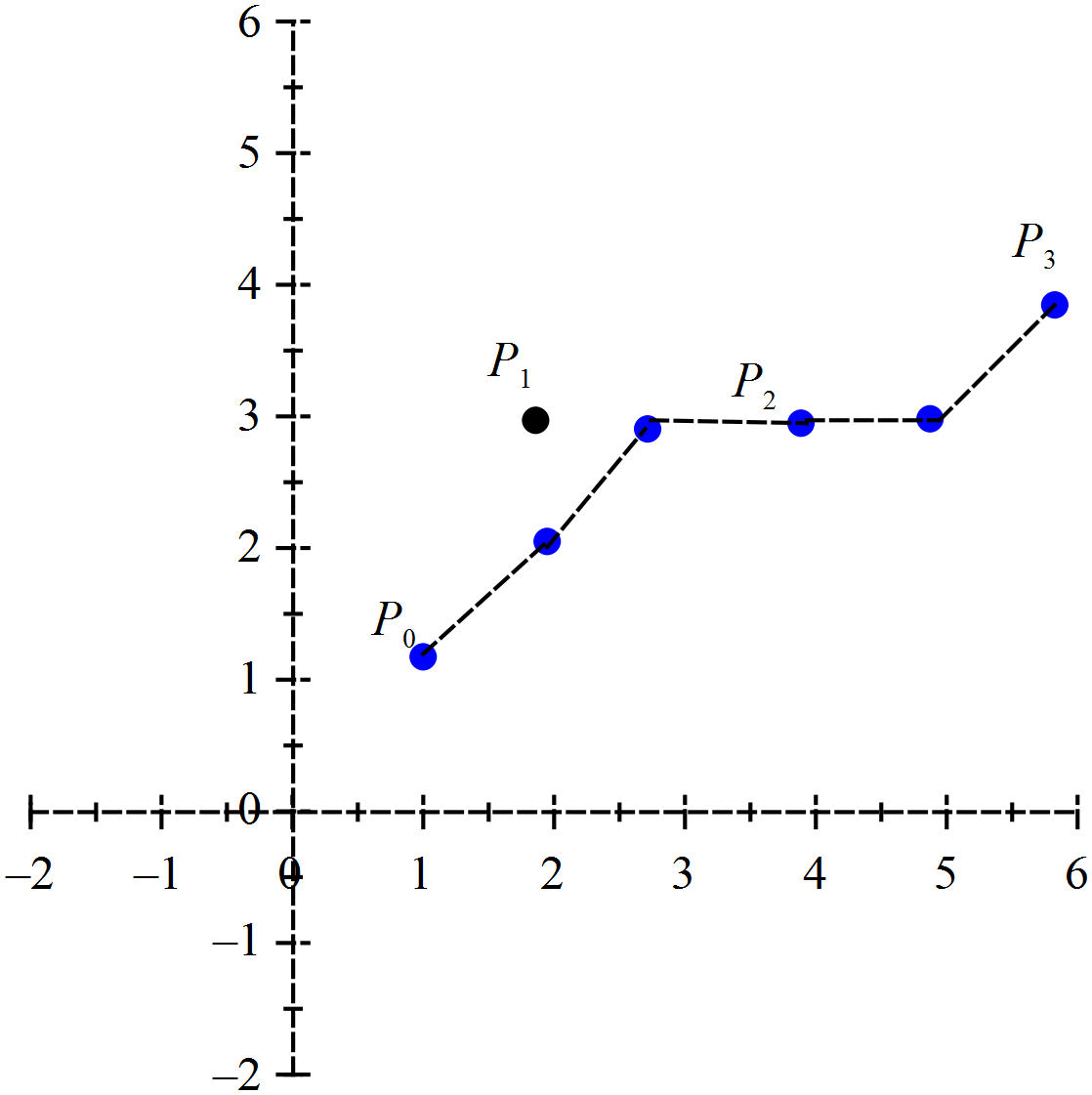
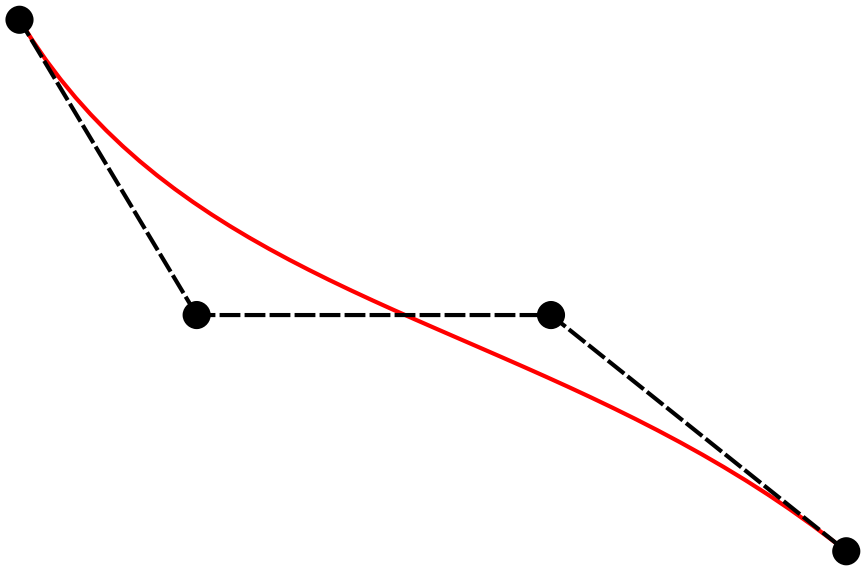
**Bài tập:**

Xác định đường cong Bezier qua 4 điểm kiểm soát (p0..3) có giá trị như sau (n = 4-1 = 3).

Pi cho trước.

Lập bảng tính 10 điểm

|  |  |  |  |
| --- | --- | --- | --- |
| u | x(u) | y(u) | ~p(u) |
| 0 | 1 | 1 | (1,1) |
| 0.1 | 1.329 | 1.543 | (1,2) |
| 0.2 | 1.712 | 1.984 | (2,2) |
| 0.3 | 2.143 | 2.341 | (2,2) |
| 0.4 | 2.616 | 2.632 | (3,3) |
| 0.5 | 3.125 | 2.875 | (3,3) |
| 0.6 | 3.664 | 3.088 | (4,3) |
| 0.7 | 4.227 | 3.289 | (4,3) |
| 0.8 | 4.808 | 3.496 | (5,3) |
| 0.9 | 5.401 | 3.727 | (5,4) |
| 1 | 6 | 4 | (6,4) |

Khi tăng số lượng điểm ta sẽ có kết quả là đường sẽ mịn hơn

~ 10000 points (p0 – p9999)

**Lập trình mô phỏng**

#include<stdio.h>

#include<graphics.h>

#include<math.h>

#define max\_dinh 4

int x[4], y[4];

int i;

//nhap cac dinh

void nhapdinh()

{

for(i=0;i<4;i++)

{

printf("x[%d]= ",i);

scanf("%d",&x[i]);

printf("y[%d]= ",i);

scanf("%d",&y[i]);

}

}

//ve cac dinh

void vedinh()

{

for(i=0;i<3;i++)

{

line(x[i],y[i],x[i+1],y[i+1]);

}

}

void bezier()

{

double t;

for(t=0.0; t<1.0; t=t+0.00005)

{

double xt=pow(1-t,3)\*x[0] + 3\*t\*pow(1-t,2)\*x[1] + 3\*pow(t,2)\*(1-t)\*x[2] + pow(t,3)\*x[3];

double yt=pow(1-t,3)\*y[0] + 3\*t\*pow(1-t,2)\*y[1] + 3\*pow(t,2)\*(1-t)\*y[2] + pow(t,3)\*y[3];

putpixel(xt,yt,WHITE);

}

//ve 4 dinh da set

for(i=0;i<4;i++)

{

putpixel(x[i],y[i],RED);

}

getch();

}

int main()

{

//nhap toa do

nhapdinh();

initwindow(600,600);

//ve cac dinh

setcolor(GREEN);

vedinh();

//ve duong cong

bezier();

getch();

}