Bài tập giữa kì

# Vẽ ngôi sao 5 cánh đều:

a

b

## **Thuật toán vẽ Bresenham vẽ đường thẳng với độ dóc m tùy ý:**

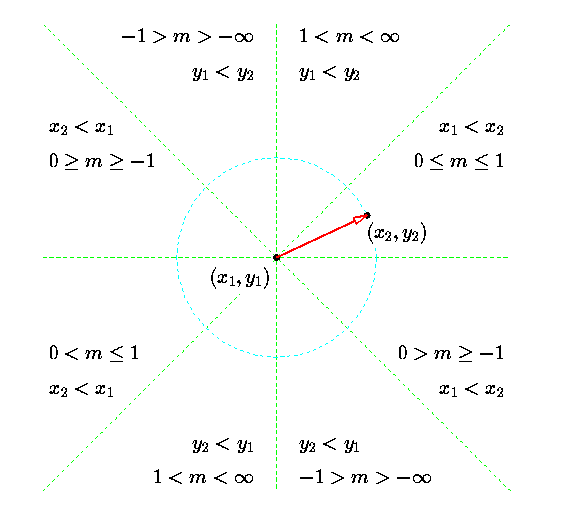
Bài toán: dựa vào tính chất đối xứng trong octant.

Với đường thẳng có m > 1, ta sẽ thay đổi vai trò của các hướng x và y. Đó là ta sẽ tìm giá trị x tiếp theo mà gần với đường thẳng nhất.

Nếu điểm bắt đầu của đường thẳng có độ dóc dương là điểm cuối cùng bên phải, cả x và y đều giảm khi chúng đi từ phải sang trái

Với m < 0, các thủ tục tương tự, khác là 1 tọa độ giảm thì tọa độ khác tăng.

Bảng các phân vùng của octant và các vùng đối xứng: Ví dụ octant 1 đối xứng octant 5



Bảng dưới đây dùng để xác định bộ octant của m:

|  |  |  |  |
| --- | --- | --- | --- |
| ΔY | ΔX?ΔY | slope | Octant |
| ≥ 0 | ≥ | Pos ≤ 1 | 1 (5) |
| ≥ 0 | < | Pos >1 | 2 (6) |
| < 0 | ≤ | Neg ≥ -1 | 7 (3) |
| < 0 | > | Neg < -1 | 8 (4) |

Mã giả cho thuật toán Bresenham với m tùy ý:

**Function:** Bresenham\_drawing\_line

**Inputs:** Start point (X1, Y1) , End point (X2, Y2)

***Begin***

*// Điểm đầu*

*X = X1; Y = Y1;*

*// Tinh ΔX, ΔY*

*ΔX = Abs(X2-X1); ΔY = Abs(Y2-Y1);*

*// Tìm dấu của x2 - x1, y2 – y1*

*S1 = Sign(X2-X1); S2 = Sign(Y2-Y1);*

*// Trao đổi vai trò của x và y*

*If ΔY > ΔX Then*

*// Swap ΔX và ΔY*

*T = ΔX; ΔX = ΔY; ΔY = T;*

*Interchange = 1;*

*Else*

*Interchange = 0;*

*End If*

*// Tính toán các thông số đầu vào*

*P = 2\*ΔY - ΔX;*

*A = 2\*ΔY;*

*B = 2\*ΔY - 2\*ΔX*

*// Vẽ điểm đầu*

*Plot(X,Y);*

*For i = 1 to ΔX Then*

*If (P < 0) Then*

*// Nếu Interchange = 1 thì dời Y*

*// Ngược lại, dời X*

*If Interchange == 1 Then*

*Y = Y + S2;*

*Else*

*X = X + S1;*

*End if*

*// Cập nhật P*

*P = P + A;*

*Else // P >= 0*

*// Tìm tọa độ X, Y*

*Y = Y + S2; X = X + S1;*

*// Cập nhật P*

*P = P + B;*

*End if*

*setPixel(X, Y);*

*End for*

***End***

## **1.1 Phát biểu bài toán:**

Xét đường tròn bán kính a là (C0), đường tròn bán kính b là (C1)

Giới hạn bài toán:

Xét tâm tại (0, 0) tịnh tiến cho ra tâm C tùy ý

Chỉ tính tọa độ 2 điểm đầu: P0 thuộc (C0), P1 thuộc đường (C1). Rồi dùng phép quay điểm P0, P1 1 góc 72 độ cho ra lần lượt các điểm P2 thuộc (C0), P3 thuộc (C1) và cứ thế cho đến khi vẽ hết ngôi sao 5 cánh đều

## **1.2 Phương pháp:**

Giả sử P0(0, a)

Để tính P1(x1, y1) ta dùng công thức sin, cos trong tam giác vuông với góc (a, b) = 36 độ:

x1 = b\*sin(36)

y1 = b\*cos(36)

Sau đó, ta sẽ dùng thuật toán đường thẳng Bresenham với m tùy ý để kẻ đường thẳng từ

P0 -> P1 để tạo thành cánh của ngôi sao.

Sau đó, ta dùng phép quay để quay điểm P0, P1 theo 1 góc alpha = 72 độ sẽ tạo ra điểm P2, P3 và cứ như thế sẽ tính được hết các điểm của ngôi sao. Công thức phép quay điểm:

x’ = cos(alpha) \* x – sina \* y

y’ = sin(alpha) \* x + cosa \* y

Lưu ý: phép quay với góc alpha dương thường thực hiện theo ngược chiều kim đồng hồ.

Tương tự, khi tính được 2 điểm mới từ phép quay ta sẽ thực hiện nối 2 điểm đó lại với nhau để tạo thành cánh của ngôi sao.

## **1.3 Giải thuật:**

**Function:** Draw\_5point\_star

**Inputs:** Tâm C(xc, yc), Bán kính đường tròn ngoài (a), Bán kính đường tròn trong (b)

***Begin***

*// Tinh do doi theo tuc hoanh va truc tung*

*trX = xc - 0;*

*trY = yc - 0;*

*// Tính điểm P0, P1*

*P0.X = 0; P0.Y = a;*

*P1.X = round(b \* sin(36 \* (PI / 180))); P1.Y = round(b \* cos(36 \* (PI / 180)));*

*// Tinh tien P0*

*P0 = Translate(P0, trX, trY);*

*P1 = Translate(P1, trX, trY);*

*// Kẻ đường thẳng nối P0, P1*

*Bresenham\_drawing\_line(P0, P1);*

*// Khoi báo 2 biến PreviousP, P2, P3*

*PreviousP = P0, P2 = P0, P3 = P1;*

*While P3 != P1 then // Lay diem P1 lam moc*

*// quay điểm P3 trước va tinh tien*

*P3 = Rotate(P3, 72);*

*P3 = Translate(P3, trX, trY);*

*// Nối PreviousP với P3*

*Bresenham\_drawing\_line(PreviousP, P3);*

*// quay điểm điểm P2*

*P2 = Rotate(P2, 72);*

*P2 = Translate(P2, trX, trY);*

*// Nối P3 đến P2*

*Bresenham\_drawing\_line(P3, P2);*

*// set lại PreviousP*

*PreviousP = P2;*

*end while*

***End***

**Function:** Rotate

**Inputs:** Điểm quay (P), góc quay (alpha)

**Outputs:** Trả về điểm đã quay theo góc quay alpha

***Begin***

Q.x = cos(alpha \* *(PI / 180*)) \* P.x - sin(alpha \* *(PI / 180)*) \* P.y;

Q.y = sin(alpha \* *(PI / 180)*) \* P.x + cos(alpha \* *(PI / 180)*) \* P.y;

return Q

***End***

**Function:** Translate

**Inputs:** Diem tinh tien (P), Độ dời trục hoành (trX), Độ dời trục tung (trY)

**Outputs:** Return điểm đã tịnh tiến

***Begin***

Q.x = P.x + trX;

Q.y = P.y + trY;

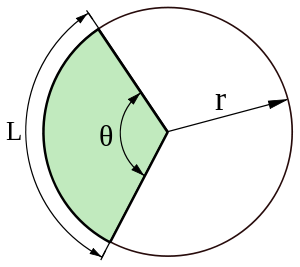
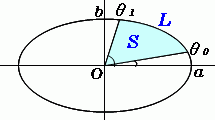
return Q

***End***

# Vẽ cung tròn/cung ellipse:

Đường tròn: Tâm C, R, P1, P2

Ellipse: Tâm C, Rx, Ry, P1, P2



P1

C

P2

2.1 Cung tròn:

2.1.1 Phát biểu bài toán:

- Phương trình đường tròn tâm C(xc, yc), bán kính R

(x – xc)^2 + (y – yc)^2 = R^2

- Vẽ cung tròn trên lưới tọa độ nguyên

- Các điểm vẽ phải thỏa yêu cầu liên tục trong lân cận 8 của điểm ảnh.

- Kiểm tra xem P1, P2 nằm trong cung chắn thứ mấy?

- Giới hạn:

Vẽ tại tâm 0

Khảo sát cung chắn 1/8

2.2 Cung ellipse:

# Vẽ đường cong Bezier bằng phương pháp de Casteljau với điểm chia theo tỷ lệ 1/2:

# Xác định số viên gạch kích thước cxd (mm) cần để nền hồ Ellipse (Bán trục lớn: a (cm), Bán trục nhỏ: b (cm)) hoặc Circle: