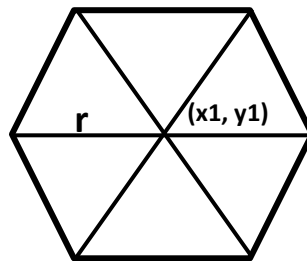


**Regular Polygon Algorithm**

Polygon is define by center  $(x1, y1)$  , radius  $(r)$  and vertices' number  $(n)$ . there are two types of polygon: regular and irregular polygon. To draw a regular polygon we use two points first and second point. First step, the first point get the second point value, then find the second point using the polar coordinates. Finally, draw a line between the first point and the second point. An equal value of angles between the regular polygon points', therefore, we need to find the increment value between two points by divided 360 on the number of polygon points.



The algorithm of draw a regular polygon will be as follow:

**Algorithm Regular\_polygon** center point  $(x1,y1)$ , radius:  $r$ , number of vertices:  $n$ )

**Begin**

$\theta = 0$

$x_s = x1 + r$

$y_s = y1$

$incr = 360 / n$

for  $i = 1$  to  $n$  do

$x_f = x_s$

$y_f = y_s$

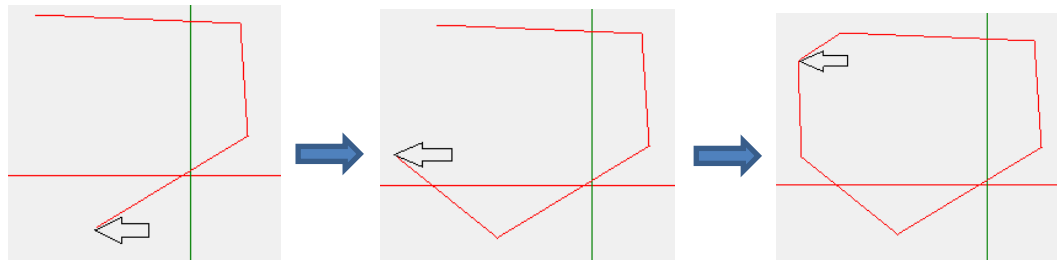
$\theta = \theta + incr$

```
xs = x1 + r * cos ( theta )  
ys = y1 + r * sin ( theta )  
call Line_DDA( xf, yf, xs, ys, color)
```

**end.**

Where (xf, yf) and (xs, ys) are the first and second points, respectively.  
And Line\_DDA is the DDA algorithm that draw a straight line.

To draw **irregular polygon** there is an interactive between the user and the screen during the drawing. There is no loop in this algorithm. it's depend on the event that received from user.



**Home Work:** write a program in V.B. net to draw an irregular polygon.

### **Star Algorithm**

star is define by center (x1, y1) , radius (r) and vertices' number (n).  
in this algorithm, we find the vertices of the star and store it in array data structure, then bind these vertices (points)with each other. To draw star we must notes to:

1. Each point bind with all points except the before and after this point.
2. There is no need to bind the two end points, because these points are bind anyways.
3. We need to two loops i , j. For example, when n =7, the values of i,j points will be as:

i	j
1	3,4,5,6
2	4,5,6,7
3	5,6,7
4	6,7
5	7

**Star Algorithm:** center point (x1,y1), radius: r, number of vertices: n)

**Begin**

theta=0

incr = 360 / n

for i = 1 to n

$x(i) = x1 + r * \cos(\text{theta})$

$y(i) = y1 + r * \sin(\text{theta})$

    theta = theta + incr

next

for i = 1 to n - 2 do

    for j = i+2 to n do

        if( not( i =1 and j = n)) then

            Line\_DDA(x(i) , y(i) , x(j) , y(j) , color)

**End**