Pranjal Save (SY-COMPS)

Computer Graphics – 22318

Unit II : Raster Scan Graphics

(Weightage-18marks)

Q1.Define polygon . Enlist its types (2marks)

Or Define concave and convex polygon(2marks)

Answer-

A closed plane figure made up of several line segments that are joined together .The side do not cross each other(Intersect).Exactly two sides meet at every vertex.

There are two types of polygon- Convex polygon & Concave polygon

Define convex and concave polygons.

Convex Polygon: It is a polygon in which if you take any two positions of polygon then all the points on the line segment joining these two points fall within the polygon itself.

Concave Polygon: It is a polygon in which if you take any two positions of polygon then all the points on the line segment joining these two points does not fall entirely within the polygon.

Convex -A polygon in which the line segment joining any two points within polygon lies completely inside the polygon is called convex polygon.

Concave-A polygon in which the line segment joining any two points within polygon may ot may not lies completely inside but partially outside is Called concave polygon.

Scanned with CamScanner

Q2.Explain Inside-Outside test(4marks)

Or Explain Odd-Even test.(4marks) / Enlist two Inside-outside rule(2marks)

Answer –

Inside Outside test- This method is also known as counting number method. While filling an object, we often need to identify whether particular point is inside the object or outside it.

In standard polygon shapes like triangle, rectangle, its components ages are joined only at vertices and edges . They do not have common point in plane and edges are not intersecting. But some graphics application produces shapes application produces shapes which which has intersecting edges . For particular shapes to find out if particular point is inside or outside the polygon . It need rule or test to follow .

There are two methods by which we can identify whether particular point is inside an object or outside namely, Odd-Even Rule, and Non-zero winding number rule.

1. Odd-Even Rule: In this technique, we count the edge crossing along the line from any point (x, y) to infinity. If the number of interactions is odd then the point (x, y) is an interior point. If the number of interactions is even then point (x, y) is an exterior point.

The last vertex is assumed to be the same as the first vertex that polygon is closed.

If the number of times this ray intersects segments making up polygon is even then the point is outside the polygon.

Whereas, if the number of intersections is odd then the points lies inside the polygons.

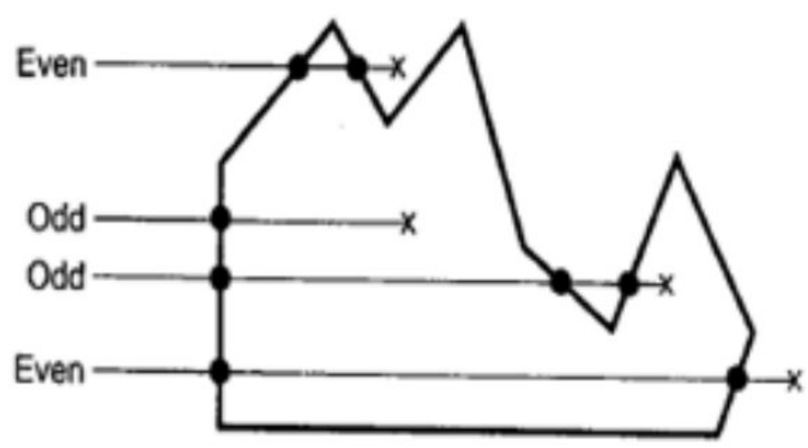


Fig. (a)

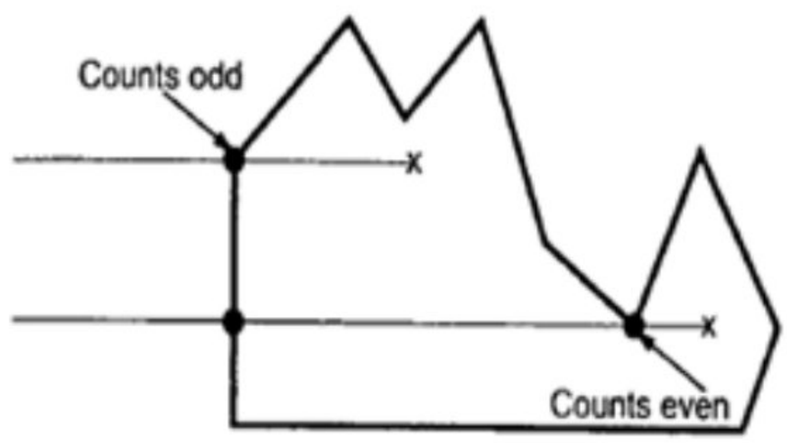


Fig. (b)

2. Non-zero Winding Number Rule: This method is also used with the simple polygons to test the given point is interior or not. It can be simply understood with the help of a pin and a rubber band. Fix up the pin on one of the edge of the polygon and tie-up the rubber band in it and then stretch the rubber band along the edges of the polygon. When all the edges of the polygon are covered by the rubber band, check out the pin which has been fixed up at the point to be test. If we find at least one wind at the point consider it within the polygon, else we can say that the point is not inside the polygon.

Scanned with CamScanner

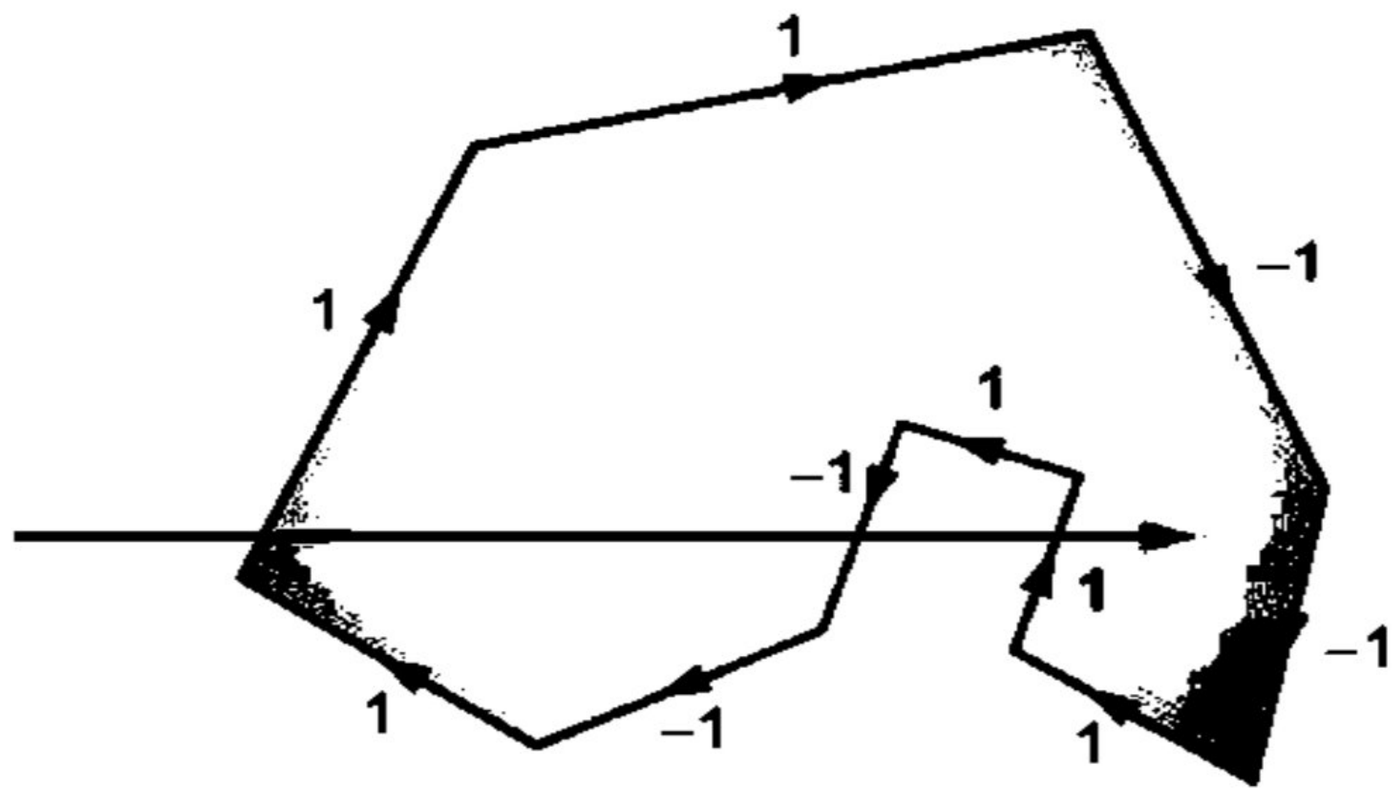


Fig b: Non-zero Winding Number Rule:

In another alternative method, give directions to all the edges of the polygon. Draw a scan line from the point to be test towards the left most of X direction. Given the value 1 to all the edges which are going to upward direction and all other – 1 as direction values.
Check the edge direction values from which the scan line is passing and sum up them.
If the total sum of this direction value is non-zero, then this point to be tested is an interior point, otherwise it is an exterior point.
In the above figure, we sum up the direction values from which the scan line is passing then the total is 1 – 1 + 1 = 1; which is non-zero. So the point is said to be an interior point.

Q3.State two polygon filling algorithm.(2marks) …scan fill algorithm..

Or Write down procedure to fill polygon using flood fill.(3marks/4marks)

Or Write down procedure to fill polygon using seed fill.(6marks)

Or Boundary fill algorithm with pseudocode. Also mention its limitations if any. (4marks)

Answer

Filling any polygon means colouring all the pixels which lie inside the polygon with colour other than the boundary colour .

We try to fill polygon by solid fill colour or pattern.

Filling of algorithm is based on co-ordinate system.

Boundary Fill Algorithm / Seed Fill Algorithm

- Boundary fill algorithm starts at a pixel , inside the polygon to be filled and paints the interior proceding outwards the boundary.
- The algorithm works only if the colour with which the region has to be filled and colour of the boundary of region are different.
- If the boundary is of one single colour, this approach proceeds outwards pixel by pixel until it hits the boundary of region.

Scanned with CamScanner

Pranjal Save (SY-COMPS)

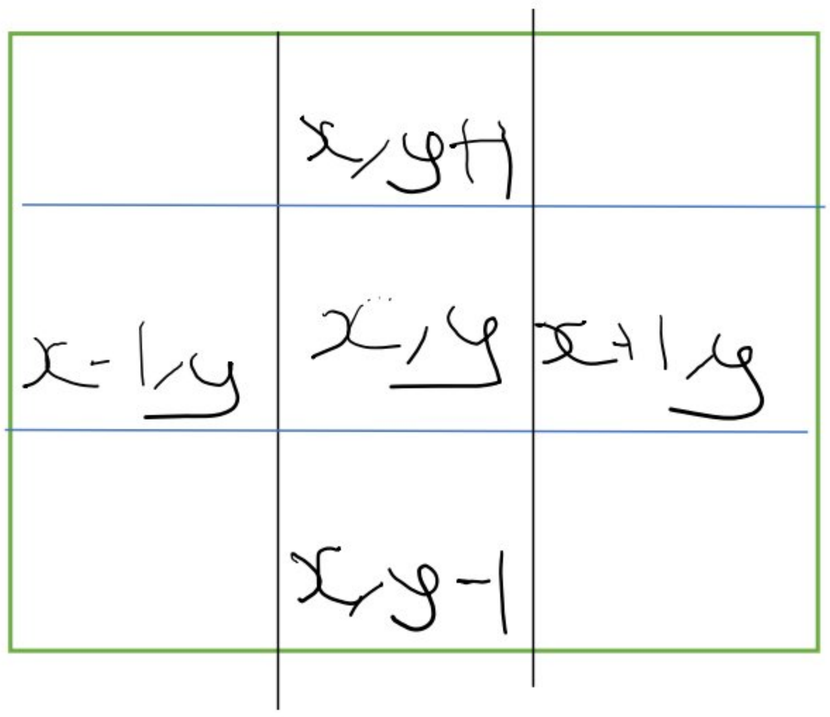
- It takes an initial point(x,y) a fill colour , and boundary color as the input.
- The algorithm start checking the colour of (x,y)
- If the colour is not equal to the fill colour and the boundary colour the its painted with fill colour and function is called all neighbours of (x,y)
- If a point is found to be of fill colour or boundary colour , the function does not call its neighbours and returns.
- This process continues until all points up to boundary colour for region have been tested.
- The boundary fill algorithm can be implemented by 4-connected pixels or 8-connected pixels.

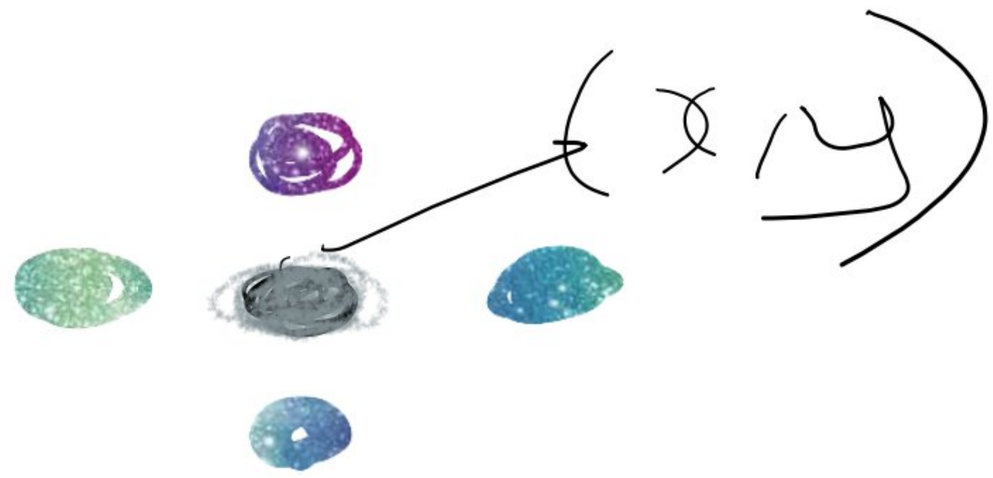
4-connected

After painting a pixel the function is called four neighbouring points.

These are the pixel positions that are right ,left , above and below the current pixel.

Area filled by the method are called 4-connected



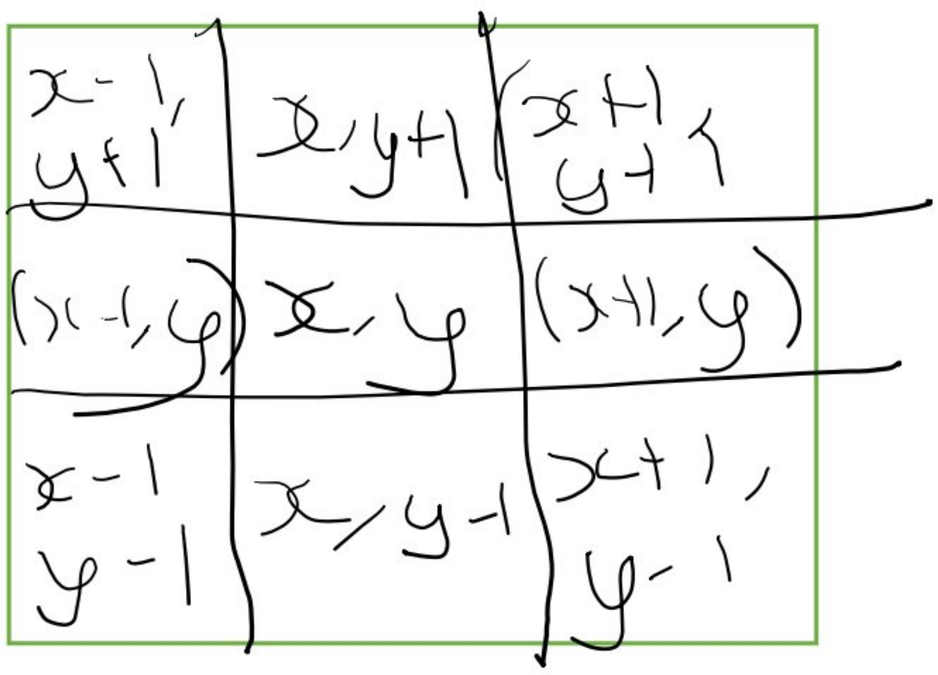


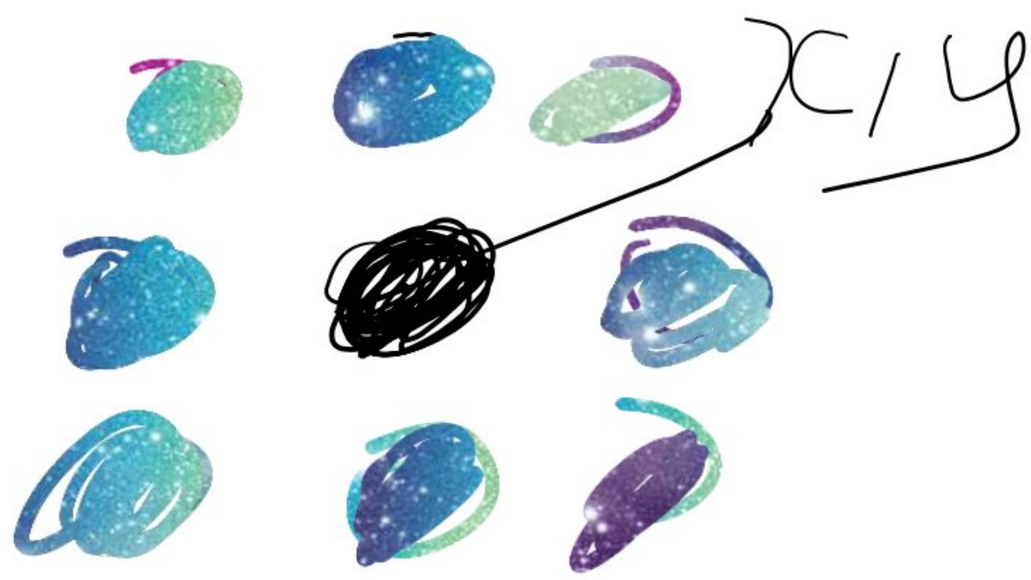
8-Connected

More complex figures are filled using this approach.

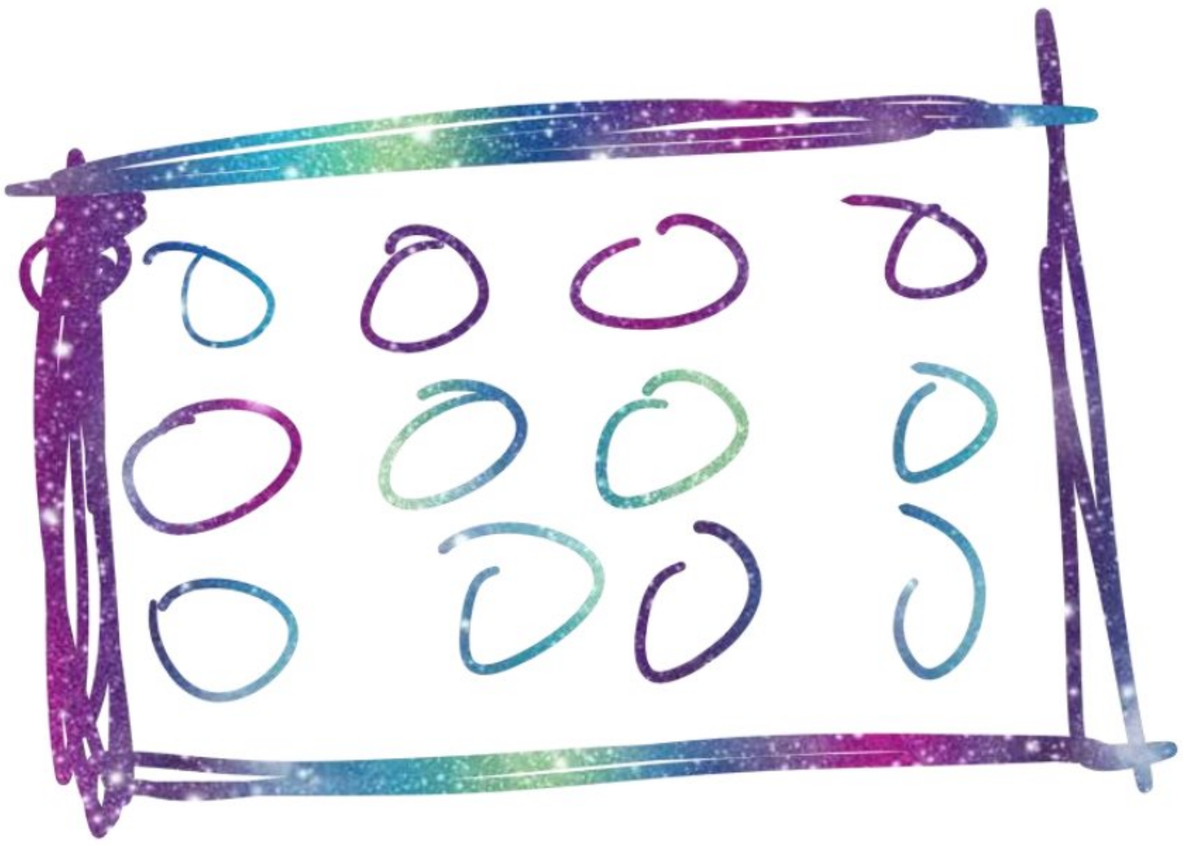
The pixels to be tested are the 8 neighbouring pixels the pixel on the right ,left,top,bottom and 4 diagonal pixels .

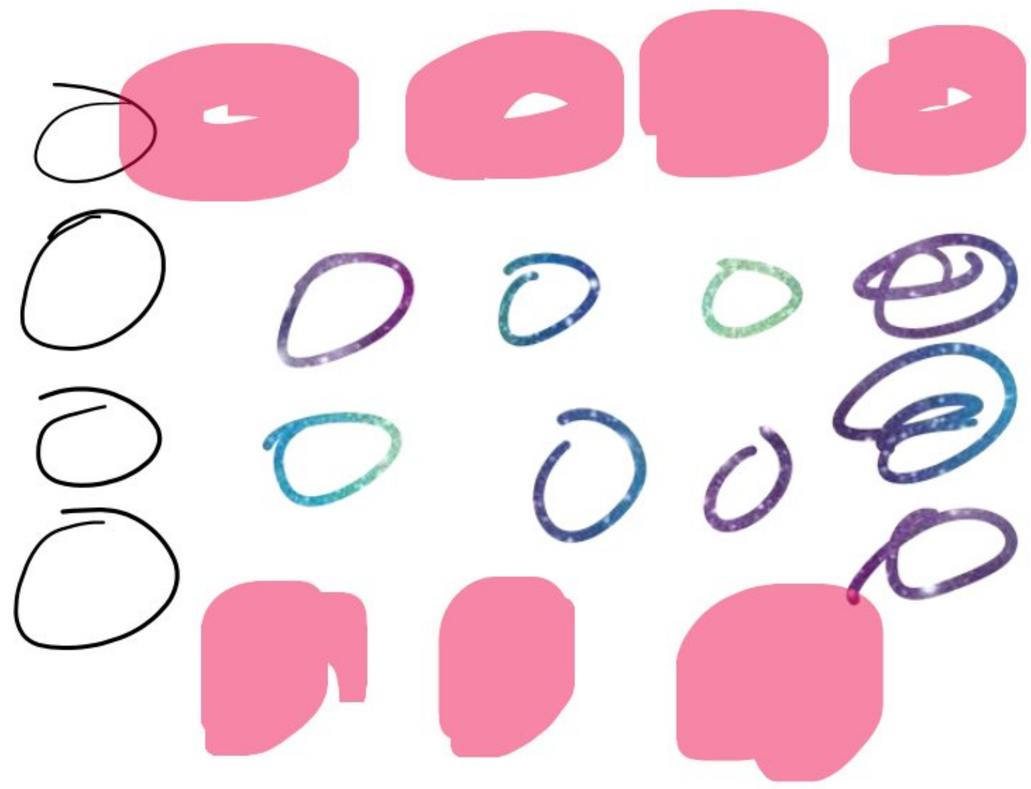
Area filled by this method are called 8 connected





Scanned with CamScanner





Q4.State the character generation methods.(2marks)

Or State & explain the character generation methods.(4marks)

Or Explain stroke and bitmap method.(4marks)

Or Explain stroke method(2marks or Explain Bitmap method(2marks)

Or State & explain the character generation methods. i) stroke ii) Bitmap iii)Starbust (6marks)

<u>Answer.</u>

- With the help of character generation methods we can display letter , numbers and other charachter on display devices.
- We can display letters , numbers and chrachters in variety of size and style
- There are three character generation methods .

Stroke	starburst	Bitmap

<u>Stroke Method</u>

1.This method uses small line segments to generate a character .

2.The small series of line segment drawn like a stroke of a pen form a character shown in figure below .

Scanned with CamScanner

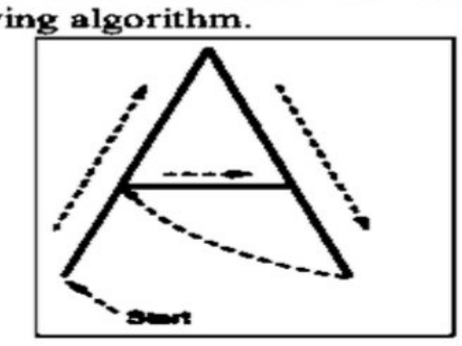
3.We can built our own stroke method by calling line drawing algorithm.

4.It is necessary to decide which line segments are needed for each character and then drawing this segments using line segments using line drawing algorithm.

5.This method supports scaling of charchter ,It does this by changing the length of line segments used for character drawing.

Ans 1) STROKE METHOD

- Stroke method is based on natural method of text written by human being. In this method graph is drawing in the form of line by line.
- Line drawing algorithm DDA follows this method for line drawing.
- This method uses small line segments to generate a character. The small series of line segments are drawn like a stroke of pen to form a character.
- We can build our own stroke method character generator by calls to the line drawing algorithm. Here it is necessary to decide which line segments are needed for each character and then drawing these segments using line drawing algorithm.



3) Starbust method:

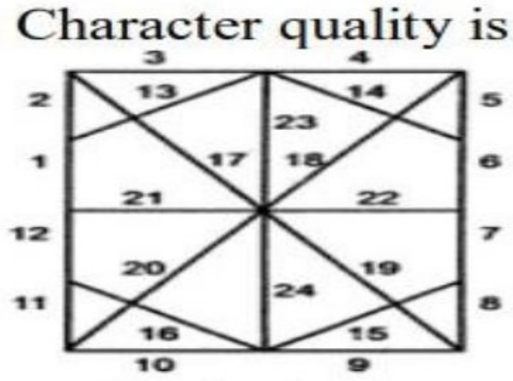
In this method a fix pattern of line segments are used to generate characters. Out of these 24 line segments, segments required to display for particular character are highlighted. This method of character generation is called starbust method because of its characteristic appearance

The starbust patterns for characters A and M. the patterns for particular characters are stored in the form of 24 bit code, each bit representing one line segment. The bit is set to one to highlight the line segment; otherwise it is set to zero. For example, 24-bit code for Character A is 0011 0000 0011 1100 1110 0001 and for

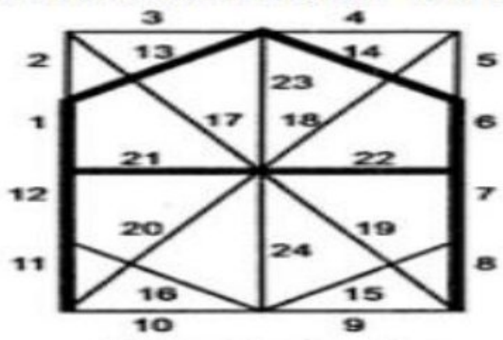
character M is 0000 0011 0000 1100 1111 0011.

This method of character generation has some disadvantages. They are

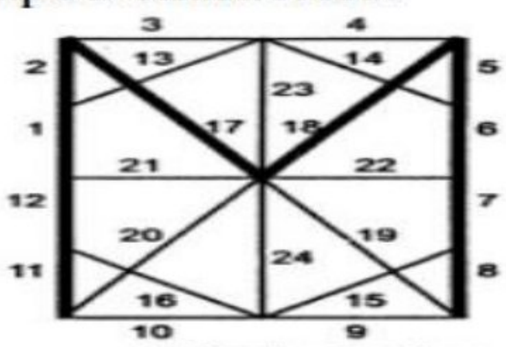
1. The 24-bits are required to represent a character. Hence more memory is required.
2. Requires code conversion software to display character from its 24-bitcode.
3. Character quality is poor. It is worst for curve shaped characters.



a) Star bust pattern of 24 line segments



b) Star bust pattern for character A



c) Star bust pattern for character M

Character A : 0011 0000 0011 1100 11100001

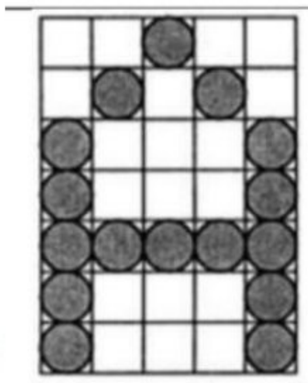
Character M:0000 0011 0000 1100 11110011

Scanned with CamScanner

2)BITMAP METHOD

- Bitmap method is a called dot-matrix method as the name suggests this method use array of bits for generating a character. These dots are the points for array whose size is fixed.
- In bit matrix method when the dots is stored in the form of array the value 1 in array represent the characters i.e. where the dots appear we represent that position with numerical value 1 and the value where dots are not present is represented by 0 in array.
- It is also called dot matrix because in this method characters are represented by an array of dots in the matrix form. It is a two dimensional array having columns and rows.

A 5x7 array is commonly used to represent characters. However 7x9 and 9x13 arrays are also used. Higher resolution devices such as inkjet printer or laser printer may use character arrays that are over 100x100.



Character A in 5 × 7 dot matri format

Q5.State different line drawing algorithmn.(2marks)

or Name two line drawing algorithmns.(2marks)

Answer-

1)DDA (Digital Differential Algorithm).

2)Bresenhm's line drawing Algorithm.

Q6. Explain DDA algorithm with an example.

Answer .

This algorithm generates a line from differential equations of line and hence the name DDA. • DDA algorithm is an incremental scan conversion method.

• A DDA is hardware or software used for linear interpolation of variables over an interval between start and end point.

• DDAs are used for rasterization of lines, triangles and polygons.

• DDA method is referred by this name because this method is very similar to the numerical differential equations.

The DDA is a mechanical device that solves differential equations by numerical method.

DDA works on the principle that we simultanesly increment X and Y by small steps proportional to first derivative of X and Y.

Scanned with CamScanner

Pranjal Save (SY-COMPS)

DDA Algorithm.

Step 1 :Read the end points of the lines.(x1,y1)(x2,y2)

Step 2 : Δx=(X2-X1) & Δy = (Y2 – Y1)

Step 3: if $\Delta x \geq \Delta y$

then(length = Δx)

else (length = Δy)

end if

Step 4: Δx = (x2 – x1)/length

Step 5: Δy = (y2 – y1)/length

Step 6: x= x1 + 0.5 * sign (Δx) & y= y1 + 0.5 * sign (Δy)

Step 7: i = 1 while (i $\leq$ length)

{ plot (integer (x), integer (y))

x = x + Δx

y = y + Δy

i = i + 1 }

Step 8: End

Q7.Explain Bresenhmn's line drawing algorithm.

The drawback's of DDA algorithm have been overcome in Bresenhmn's algorithm.

Breshnhams algorithm's plot the line using only increment integer calculation .

As computer can perform integer operation rapidly this algorithm is very efficient method for straight lines.

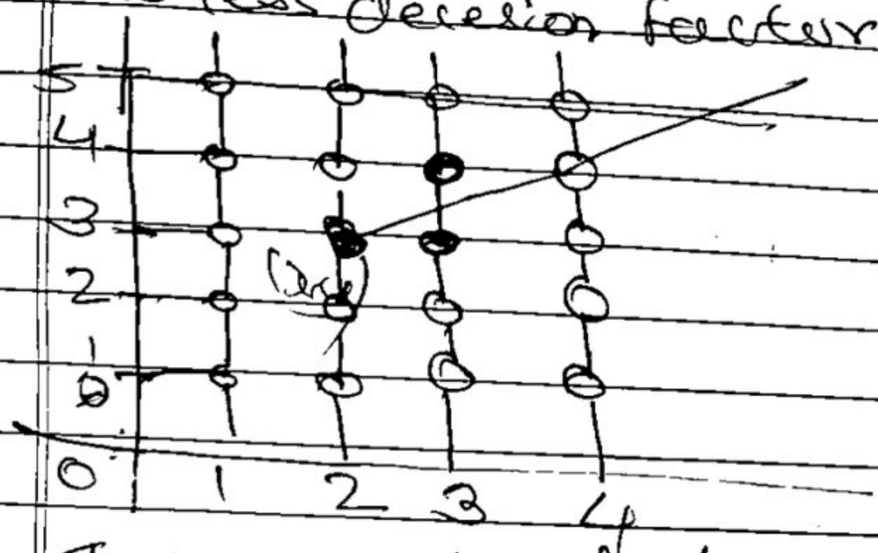
The algorithm increments always either x and y by 1unit value depending.

The Bresenham Line Algorithm is a powerful tool for efficient line drawing and line-related operations. Its ability to determine the most accurate pixels to plot using integer arithmetic makes it a popular choice in computer graphics, image processing, and computational geometry.

By understanding the inner workings of the Bresenham Line Algorithm and its applications, you can leverage its efficiency to optimize line drawing in your projects. So go ahead, implement the Bresenham Line Algorithm, and explore the possibilities it offers for smooth and accurate line representations.

Scanned with CamScanner

For example, from position (2,3) we have to plot next point for that we have to chose between (2,3) & (3,4) best on whichever is nearest pixel to the orginll line that is which has less decision factor



The increment value in other variable is determined by examing the distance between actual line location & the nearest pixel. This distance is called as decision factor

Algorithm

Step 1 : Read line end points as .(x1,y1)(x2,y2)

Step 2 : Δx=(X2-X1) & Δy = (Y2 – Y1)

Step 3: Initalize starting point of line ,

i,e x=x1 y=y1

Step 4:Plot (x,y) i.e plot first

Step 5 : Obtain initial value value of decision parameter Pk as,

Pk=2Δy-Δx

Step 6:If Pk<0(Here Pk=P0 as initial value)

{

x=x+1;

y=y;

Pk=Pk+ 2Δy

}

Plot(x,y)

Pk>=0

x=x+1

y=y+1

Pk=Pk+2Δy-2Δx

Step 7: Repeat step 6(Δx) times.

Step 8 : Stop

Scanned with CamScanner

Pranjal Save (SY-COMPS)

Q8 .List any two line drawing algorithms. Also, list two merits of any line drawing algorithm.(2MARKS)

Answer

Line drawing algorithms:

- Digital Differential Analyzer (DDA) algorithm
- Bresenham's algorithm

Merits of DDA algorithms:

- It is the simplest algorithm and it does not require special skills for implementation.
- It is a faster method for calculating pixel positions than the direct use of equation y = mx + b. It eliminates the multiplication in the equation by making use of raster characteristics, so that appropriate increments are applied in the x or v direction to find the pixel positions along the line path
- Floating point Addition is still needed.

Merits of Bresenham's Algorithm:

- Bresenhams algorithm is faster than DDA algorithm
- Bresenhams algorithm is more efficient and much accurate than DDA algorithm.
- Bresenham's line algorithm is a highly efficient incremental method over DDA.
- Bresenhams algorithm can draw circles and curves with much more accuracy than DDA algorithm. It produces mathematically accurate results using only integer addition, subtraction, and multiplication by 2, which can be accomplished by a simple arithmetic shift operation.

Q9. Rephrase the Bresenham's algorithm to plot 1/8th of the circle and write the algorithm required to plot the same. (4MARKS)

OR Explain or Write Bresehnam's Circles Circle drawing Algorithm.(4marks/6marks)

Answer :

Raster Scan Graphics

2.2 CIRCLE GENERATING ALGORITHMS

2.2.1 Symmetry of Circle

- **Before** knowing circle generating algorithms, let's see the symmetry of circle. Very first of all circle **is a** symmetrical figure. The shape of circle is similar in each quadrant. After that circle is also **symmetrical** between octants.
- **Thus**, circle has eight way symmetry i.e. symmetrical in all octants. Once one point in one octant is **calculated**, it can be mapped into other seven circle points in other octants.
- **Fig.** 2.6 shows Symmetry of a Circle where **point** (x, y) in first octant gives circle points in **other** seven octants.
- **Now**, we will see the actual circle generating **algorithms** as,
  1. DDA Algorithm.
  2. Bresenham Algorithm.
  3. Mid-point Algorithm.

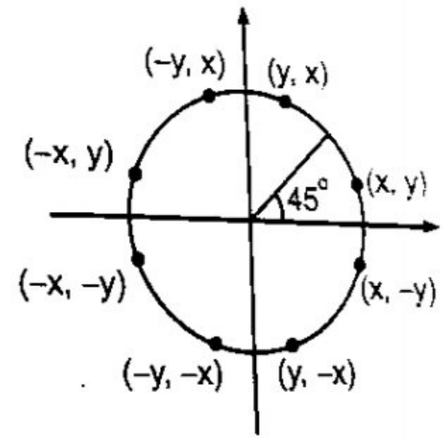


Fig. 2.6: Symmetry of Circle

Scanned with CamScanner

Pranjal Save (SY-COMPS)

Bresenham's Circle Generating Algorithm

This algorithm uses the symmetry of circle as seen before. Here Bresenham's line generating algorithm is adopted for generating circle, where at each sampling step decision parameter decides the closest pixel to the circumference of circle and then plot the nearest pixel on circumference to plot the circle. The shows how points on circle circumference are plotted.

Algorithm : Bresenham Circle Generating Algorithm.

Step 1 : Read radius (r) of circle.

Step 2 : Calculate initial decision variable P_i.

Step 3 : $x = 0$ and $y = r$

Step 4 : if ($P_i < 0$)

{

$x = x + 1$

$P_i = P_i + 4x + 6$

}

else if ($P_i \geq 0$)

{

$x = x + 1$

$y = y - 1$

$P_i = P_i + 4(x - y) + 10$

}

Step 5 : Plot pixels in all octants as,

Plot (x, y) (∵ Referring to Fig, 2.6)

Plot (y, x)

Plot (– y, x)

Plot (– x, y)

Plot (– x, – y)

Plot (– y, – x)

Plot (y, – x)

Plot (x, – y)

Step 6 : Stop

Generating Algorithm

Scanned with CamScanner

Pranjal Save (SY-COMPS)

Q10.Frame buffer(2marks)

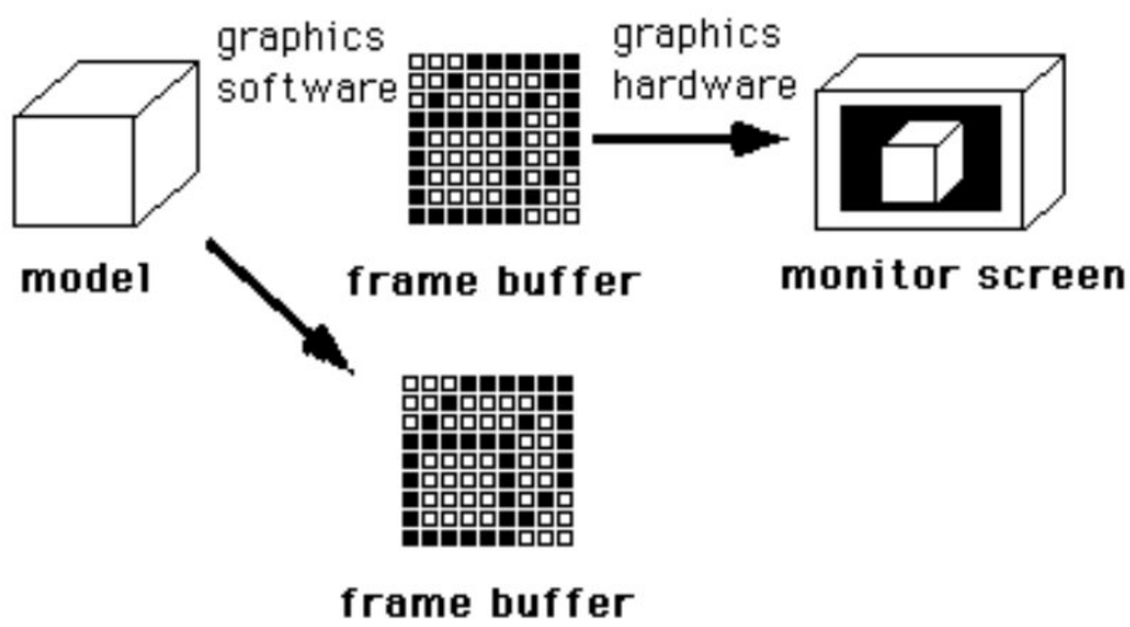
Frame Buffer The frame buffer is the video memory (RAM) that is used to hold or map the image displayed on the screen. OR

A framebuffer (frame buffer, or sometimes framestore) is a portion of RAM containing a bitmap that drives a video display.

It is also known as Refresh buffer

frame buffer An area of memory used to store information related to the pixels of a display. A frame buffer for a color display has a set of *planes* each defining one bit of the color information of each point. Typically a color display will have a frame buffer with up to 24 planes, 8 each for defining red, green, and blue

A memory area called refresh buffer or frame buffer stores picture definition. This memory area holds intensity values for all screen points. Stored intensity values are restored from frame buffer and painted on screen taking one row at a time. Each screen point is referred to as pixels.



Scanned with CamScanner

6marks

DDA Circle Generating Algorithm

Step1 : Read radius of circle (r)
calculate value of epsilon ε

Step2 : $x = 0$
$y = r$

Step3 : $x_1 = x$
$y_1 = y$

Step4 : while $((y_1 - y) < \varepsilon \,||\, (x - x_1) > \varepsilon)$
{
$x_1 = x_1 + \varepsilon y_1$
$y = y_1 + \varepsilon x_1$
plot(int(x), int(y))
}

Step5 : Stop

Scanned with CamScanner

ider the line from A(10,2) to B(6,8). Use DDA
gorithm to rastorize this line — (4 marks)

Step 1: Read the points on line

$(x_1, y_1) = A(10,2)$ $\therefore x_1 = 10$ $y_1 = 2$

$(x_2, y_2) = B(6,8)$ $x_2 = 6$ $y_2 = 8$

Step 2: $\Delta x = x_2 - x_1$
$= 6 - 10$
$\boxed{\Delta x = -4}$

& $\Delta y = y_2 - y_1$
$= 8 - 2$
$\boxed{\Delta y = 6}$

Step 3: if $\Delta x > \Delta y$
length $= \Delta x$ or length Δy.

So, $-4 > 6$
So, length $= \Delta y$
$\boxed{\text{length} = 6}$

Step 4: $\Delta x = (x_2 - x_1)/\text{length}$
$-4/6$
$\boxed{\Delta x = -0.6666}$

$\Delta y = (y_2 - y_1)/\text{length}$
$= 6/6$
$\boxed{\Delta y = 1}$

Step 5:

$x = x_1 + 0.5 * \text{sign}(\Delta x)$
$x = 10 + 0.5 * (-0.660)$
$x = 9.6667$

$y = y_1 + 0.5 * \text{sign}(\Delta y)$
$2 + 0.5 * (1)$
$y = 2.5$

Scanned with CamScanner

Step 6 : i=1

```
while (i≤length)
 {
   int(x); int(y)
     x=x+Δx;
     y=y+Δy;
     i=i+1
 }
```

$1 \leq 6$

$x = -0.6666 + 9.666$

$\boxed{x = 9.0001}$

$y = 2.5 + 1$

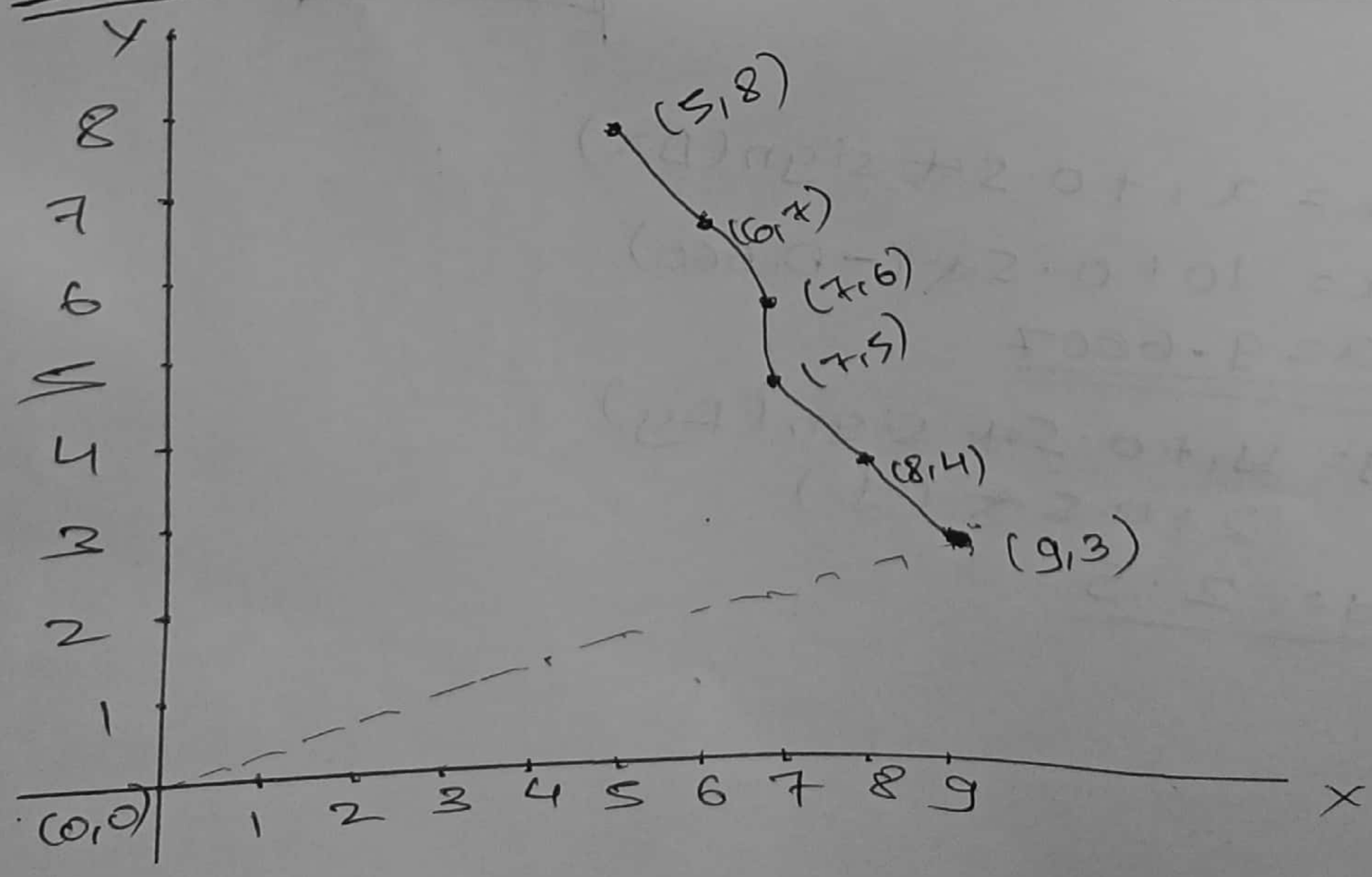
$\boxed{y = 3.5}$

$(x, y) = (9.0001, 3.5)$

Iterations

i	x	y	Plot Pixel (x, y)
1	9.0001	3.5	(9,3)
2	8.3335	4.5	(8,4)
3	7.6669	5.5	(7,5)
4	7.0003	6.5	(7,6)
5	6.3337	7.5	(6,7)
6	5.6671	8.5	(5,8)

Graph



Q. 6 Consider the line from (0,0) to (4,6). Use the simple DA algorithm to rasterlize this line.

Step 1: Read the end points.

$(x_1, y_1) = (0,0)$

$(x_2, y_2) = (4,6)$

$x_1 = 0 \quad x_2 = 4$

$y_1 = 0 \quad y_2 = 6$

Step 2: $\Delta x = x_2 - x_1$

$= 4 - 0$

$\Delta x = 4$

$\Delta y = y_2 - y_1$

$= 6 - 0$

$\Delta y = 6$

Step 3: length ($\Delta x > \Delta y$)

then Δx = length or Δy = length

$\therefore 4 > 6$

So, Δy = length

length = 6

Step 4: $\Delta x = 4/6$

$\Delta x = 0.6666$

$\Delta y = 6/6$

$\Delta y = 1$

Step 5: $x = x_1 + 0.5 * \text{sign}(\Delta x)$

$= 0 + 0.5 * 0.6666$

$x = 0.3333$

$y = y_1 + 0.5 * \text{sign}(\Delta y)$

$= 0 + 0.5 * \text{sign}(1)$

$y = 0.5$

Scanned with CamScanner

Step 6 :

```
i=1
while(i ≤ length)
{
 x=x+Dx;
 y=y+Dy;
 i++;
}
```

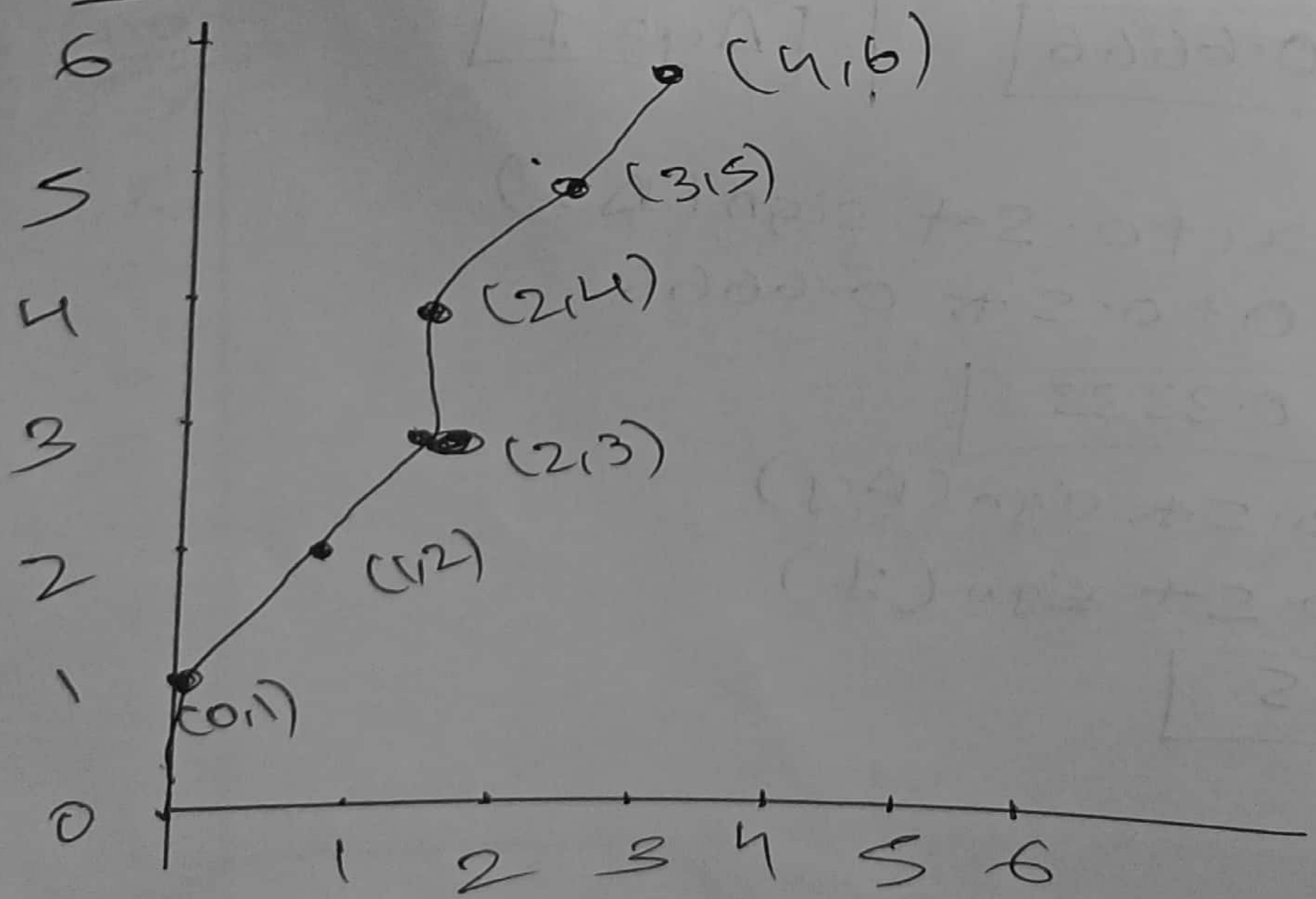
```
while (1 ≤ 6)
{
x= 0.3333 + 0.6666
y= 0.5 + 1
```

$\boxed{y=1.5}$ $\Rightarrow$ $\boxed{x=0.9999}$ $\boxed{x=1}$

```
}
```

i	x	y	(x, y)
1	0.9999	1.5	(0,1)
2	1.6665	2.5	(1,2)
3	2.3331	3.5	(2,3)
4	2.9997	4.5	(2,4)
5	3.6663	5.5	(3,5)
6	4.3329	6.5	(4,6)

Graph



A Line generating algorithm

ine = (0,0) to (5,6)

Step1 : Read points. $x_1 = 0$ $y_1 = 0$
$x_2 = 5$ $y_2 = 6$

Step2 : $\Delta x = x_2 - x_1$
$= 5 - 0$
$\boxed{\Delta x = 5}$

$\Delta y = y_2 - y_1$
$= 6 - 0$
$\boxed{\Delta y = 6}$

Step3 : $\boxed{\Delta x < \Delta y}$
length = Δy
$\boxed{\text{length} = 6}$

Step 4 : $\Delta x = (x_2 - x_1) / \text{length}$
$= 5/6 = 0.833$
$\boxed{\Delta x = 0.833}$

$\Delta y = (y_2 - y_1) / \text{length}$
$= 6/6 = 1$
$\boxed{\Delta y = 1}$

Step 5 :
$x = x_1 + 0.5 * \text{sign}(\Delta x)$
$= 0 + 0.5 \times 0.833$
$= 0.4165$
$\boxed{x = 0.5}$

$\boxed{y = 0.5}$ $y = 0.5 \times 1$

=1

while (1 ≤ length)

{

x = x + Δx

y = y + Δy

i = i++

}

while (1 ≤ 6)

{

x = 0.5 + 0.833 $\boxed{x = 1.33}$

y = 0.5 + 1 $\boxed{y = 1.5}$

i = 2

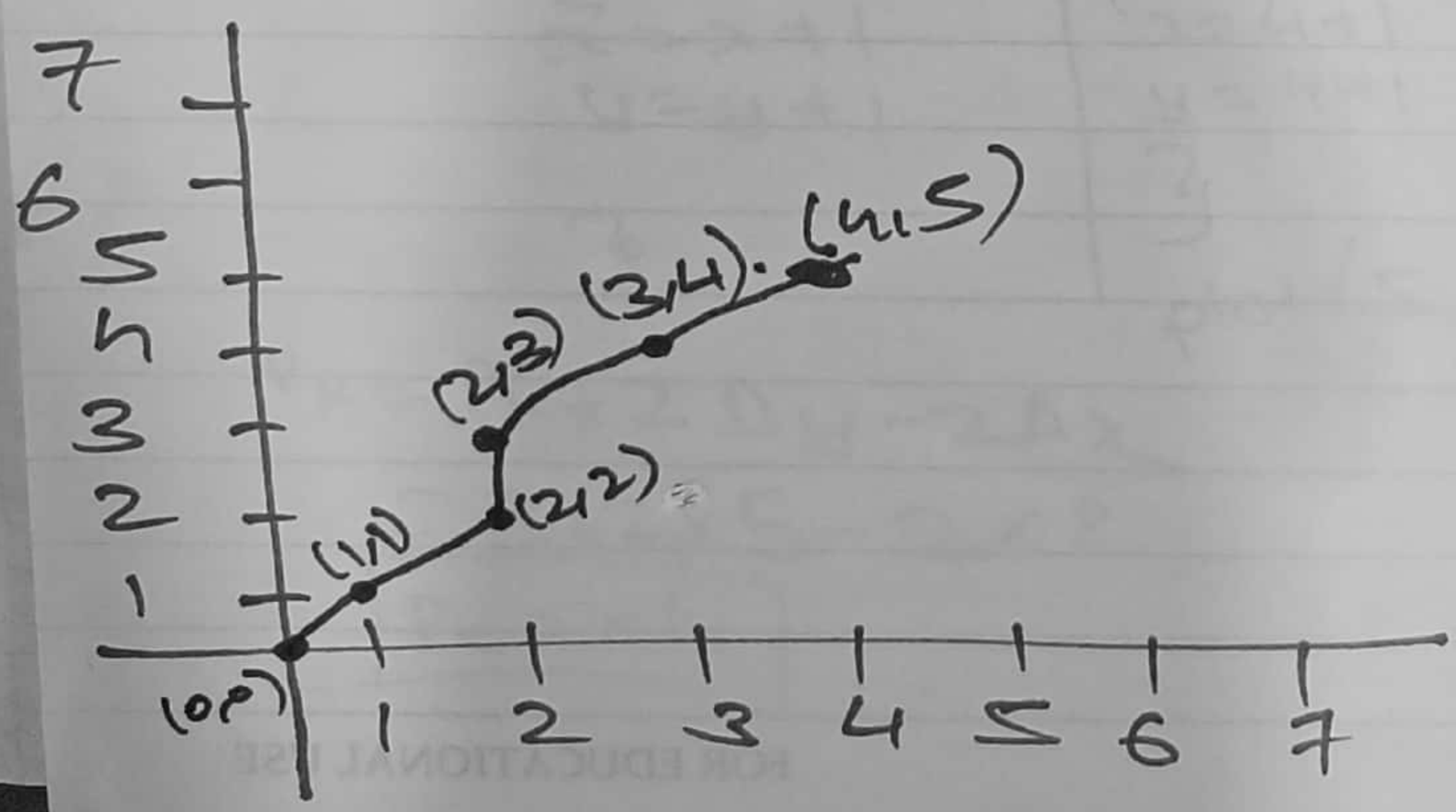
}

Repeat & continue till loop.

Step 7 :

i	x	y	Plot
1	0.5	0.5	(0,0)
2	1.33	1.5	(1,1)
3	2.166	2.5	(2,2)
4	2.99	3.5	(2,3)
5	3.832	4.5	(3,4)
6	4.665	5.5	(4,5)

Step 7 : plot graph.



Step 8 : End

amCS

Pranjal Save (SY-COMPS)

Winter 18
Summer 22

sider line from (4,4) to (12,9). Use Bresenham's gorithm to rasterize line. (6marks) 4marks)

Step 1: Read line end point

$(x_1, y_1) = (4,4)$

$(x_2, y_2) = (12,9)$

$x_1 = 4 \quad x_2 = 12$

$y_1 = 4 \quad y_2 = 9$

Step 2: ~~initializ~~

$\Delta x = x_2 - x_1 = 12 - 4$ $\boxed{\Delta x = 8}$

$\Delta y = y_2 - y_1 = 9 - 4$ $\boxed{\Delta y = 5}$

Step 3: initalize

$x = x_1$ & $y = y_1$

$\boxed{x = 4}$ $\boxed{y = 4}$

Step 4: plot (x, y) so (4,4) — (1)

Step 5

$P_K = 2\Delta y - \Delta x$

$P_K = 2 \times 5 - 8$

$10 - 8$

$\boxed{P_K = 2}$

Step 6: Check P_K condition 2

$P_K \geq 0$	$2 \geq 0$
{	{
$x = x + 1$	$x = 4 + 1$
$y = y + 1$	$y = 4 + 1$
}	}
	plot (5, 5) — (2)

$\boxed{x = 5 \quad y = 5}$

$P_K = P_K + 2\Delta y - 2\Delta x$

$= 2 + 2 \times 5 - 2 \times 8$

$\boxed{P_K = -4}$

Scanned with CamScanner

check condition I

$P_k < 0$

$-4 < 0$

$x = x+1$

$y = y$

$P_k = P_k + 2\Delta y$

$x = 5+1$ $\boxed{x=6}$

$y = y$ $\boxed{y=5}$

So, $(x,y) = (6,5)$ — (3)

$P_k = P_k + 2\Delta y$

$= -4 + 2\times 5$

$\boxed{P_k = +6}$

check condition; (II)

$P_k \geq 0$

$6 \geq 0$

$x = x+1$

$y = y+1$

$P_k = P_k + 2\Delta y - 2\Delta x$

$x = 6+1$ $\boxed{x=7}$

$y = 5+1$ $\boxed{y=6}$

So $(x,y) = (7,6)$ — (4)

$P_k = P_k + 2\Delta y - 2\Delta x$

$= 6 + 2\times 5 - 2\times 8$

$= 6 + 10 - 16$

$= 16 - 16$ $\boxed{P_k = 0}$

Check condition,

$P_k \geq 0$ ..... Condition II

$0 \geq 0$

$x = x+1$

$y = y+1$

$P_k = P_k + 2\Delta y - 2\Delta x$

$x = 7+1 = 8$ $\boxed{x=8}$

$y = 6+1 = 7$ $\boxed{y=7}$

So, $(8,7)$ — (5)

$P_k = P_k + 2\Delta y - 2\Delta x$

$= 0 + 2\times 5 - 2\times 8$

$\boxed{P_k = -6}$

Condition,

$k < 0$ ... Condition ①

$6 < 0$

{ $x = x+1$
$y = y$
$P_k = P_k + 2\Delta y$ }

$x = 8+1$ $\boxed{x = 9}$
$y = 7$ $\boxed{y = 7}$
So, $(9, 7)$

~~check condition~~, $P_k = P_k + 2\Delta y$ ——— ⑥

$= -6 + 2 \times 8$

$\boxed{P_k = 10}$

check condition

$P_k \geq 0$

$10 \geq 0$

{
$x = x+1$
$y = y+1$
} $P_k = P_k + 2\Delta y - 2\Delta x$

$x = 9+1$ $\boxed{x = 10}$
$y = 7+1$ $\boxed{y = 8}$
So, $(10, 8)$ ——— ⑦

$P_k = 10 + 2.5 - 2.8$

$\boxed{P_k = 4}$

Check condition,

$P_k \geq 0$

{
$x = x+1$
$y = y+1$
}

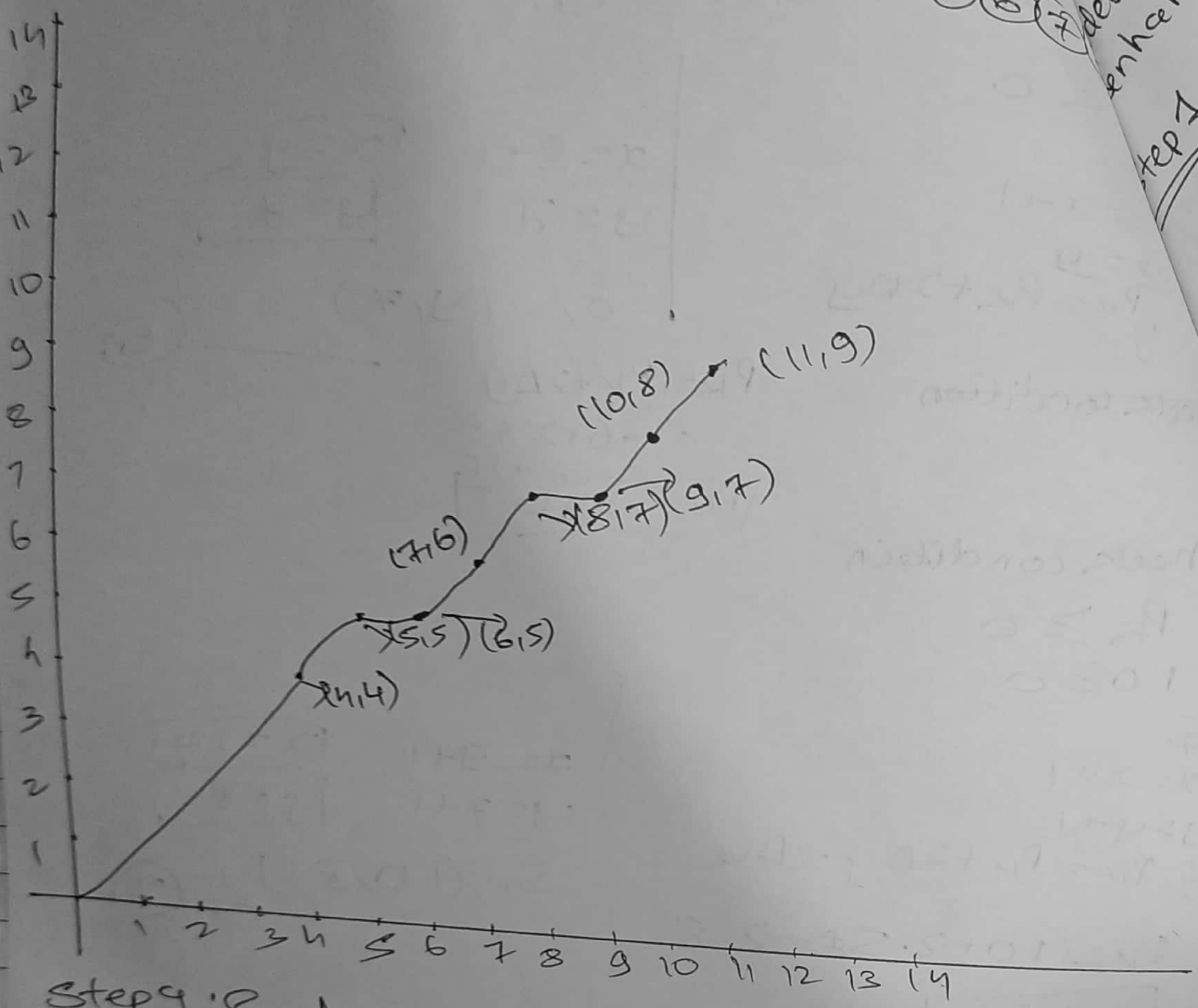
$x = 10+1$ $\boxed{x = 11}$
$y = 8+1$ $\boxed{y = 9}$

So, $(11, 9)$ ..... ⑧

<u>Step 7</u>: Repeat (Δx) times

$= 8$ times

Step8 : Plot from -- (1) (2) (3) (4) (5) (6) (7)



Step9 : End

*der the line from (5,5) to (13,9). Use the enhams algorithm to rasterize line.

(6 marks)

tep 1: Read line end points

$(x_1, y_1) = (5,5)$

$(x_2, y_2) = (13,9)$

$x_1 = 5$ $x_2 = 13$

$y_1 = 5$ $y_2 = 9$

Step2: $\Delta x = x_2 - x_1 = 13 - 5$ $\boxed{\Delta x = 8}$

$\Delta y = y_2 - y_1 = 9 - 5$ $\boxed{\Delta y = 4}$

Step 3: Initialize,

$x = x_1$ & $y = y_1$

$\boxed{x = 5}$ & $\boxed{y = 5}$

step4: plot (x, y) So $(5,5)$ on graph

1

Step5: $\boxed{P_k = 2\Delta y - \Delta x}$

$= 2 \times 4 - 8$

$= 8 - 8$

$\boxed{P_k = 0}$

Step6: Check P_k and Accordingly put in condition

$P_k \geq 0$

{

$x = x + 1$

$y = y + 1$

$P_k = P_k + 2\Delta y - 2\Delta x$

}

$0 \geq 0$

$x = 5 + 1$ $\boxed{x = 6}$

$y = 5 + 1$ $\boxed{y = 6}$

$(6,6)$

2

$P_k = 0 + 2 \times 4 - 2(8)$

$= 8 - 16$

$\boxed{P_k = -8}$

check condition:

$P_k < 0$

$\{$
$x = x+1$
$y = y$
$P_k = P_k + 2D_y$
$\}$

$-8 < 0$

$x = 6+1$ $\quad x = 7$

$y = 6$ $\quad y = 6$

$\therefore (7,6)$ —— (3)

$P_k = P_k + 2D_y$

$P_k = -8 + 2 \times 4$

$= -8 + 8$

$\boxed{P_k = 0}$

Check condition:

$P_k \geq 0$

$\{$
$x = x+1$
$y = y+1$
$P_k = P_k + 2D_y - 2D_x$
$\}$

$x = 7+1$ $\quad \boxed{x = 8}$

$y = 6+1$ $\quad \boxed{y = 7}$

$\therefore (8,7)$ —— (4)

$P_k + 2D_y - 2D_x$

$0 + 2 \times 4 - 2 \times 8$

$= 8 - 16$

$\boxed{P_k = -8}$

$P_k < 0$

$\{$
$x = x+1$
$y = y$
$P_k = P_k + 2D_y$
$\}$

$x = 8+1$ $\quad \boxed{x = 9}$

$\boxed{y = 7}$

$(9,7)$ —— (5)

$P_k = -8 + 2(4)$

$\boxed{P_k = 0}$

condt(ii)

≥ 0

$x = x+1$
$y = y+1$
$P_k = P_k + 2\Delta y - 2\Delta x$
}

$P_k = 0 + 2\times 4 - 2(8)$
$= 8 - 16$
$P_k = -8$

$x = 9+1$ $\boxed{x = 10}$
$y = 7+1$ $\boxed{y = 8}$
$(10, 8)$ — 6

So,
$P_k < 0$
{ $x = x+1$
$y = y$
$P_k = P_k + 2\Delta y$
}

$x = 11$
$y = 8$

$(11, 8)$ — 7

$P_k = 0 + 2 \cdot 4$
$\boxed{P_k = -8}$

$P_k > 0$
{ $x = x+1$
$y = y+1$
}

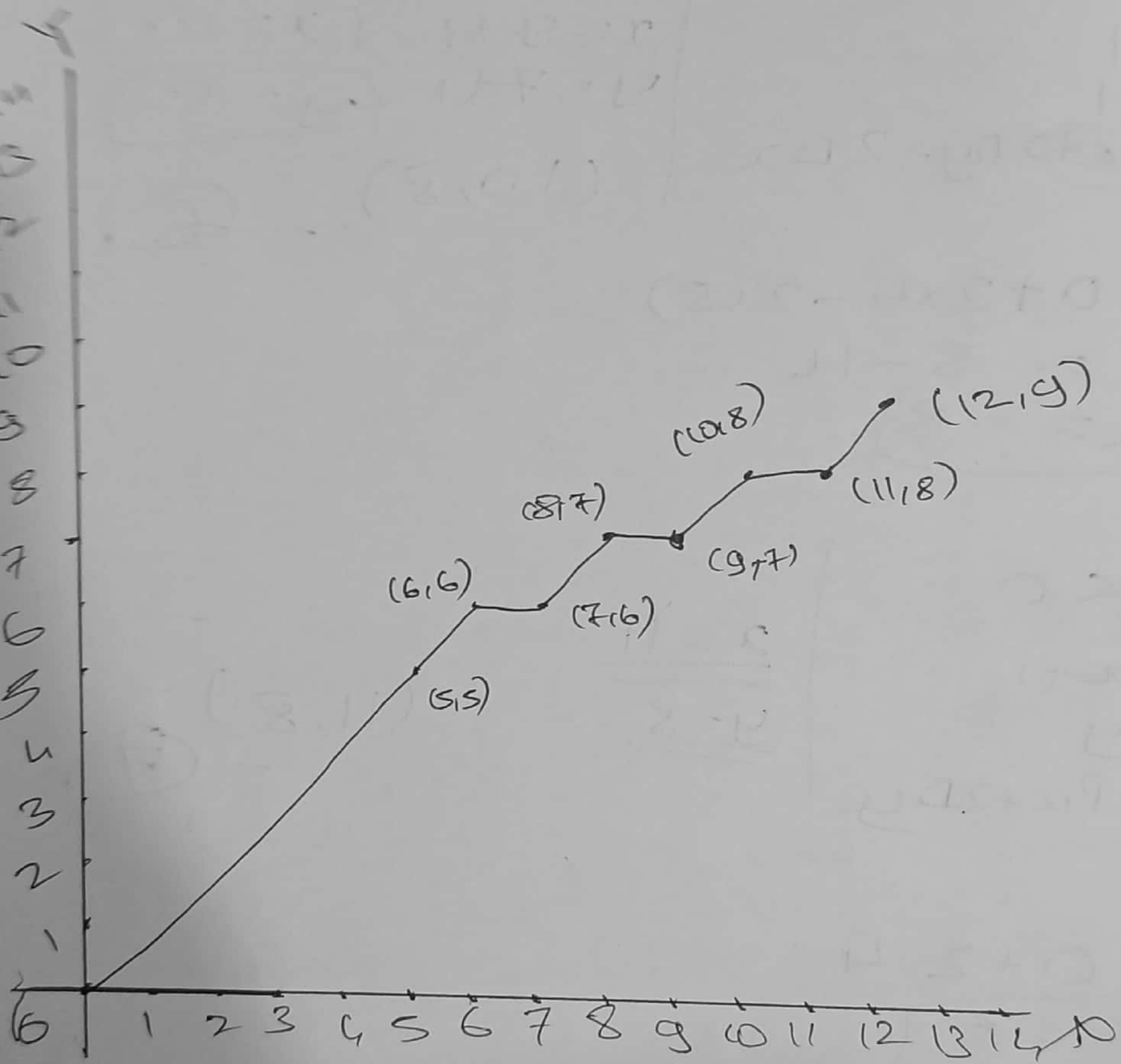
$\boxed{x = 12}$
$\boxed{y = 9}$

$(12, 9)$ — 8

Step 7: Repeat (Δx) times
= 8 times.

Scanned with CamScanner

Step 8 : Plot graph. from (1), (2), (3) (4) (8)



(4) ...ns

...resenham's line drawing algorithm to rasterize from (6,5) to (15,10). ..... (6 marks)

Step 1: Read the end points.

$(x_1, y_1) = (6,5)$ | $x_1 = 6$ $y_1 = 5$

$(x_2, y_2) = (15,10)$ | $x_2 = 15$ $y_2 = 10$

Step 2:

$\Delta x = x_2 - x_1$ $\Delta x = 15 - 6$ $\boxed{\Delta x = 9}$

$\Delta y = y_2 - y_1$ $\Delta y = 10 - 5$ $\boxed{\Delta y = 5}$

Step 3: Initalize

$x = x_1$ & $y = y_1$

$\boxed{x = 6}$ $\boxed{y = 5}$

Step 4: plot(x,y) i.e (6,5) on graph

Step 5: $\boxed{P_k = 2\Delta y - \Delta x}$ ..... (1)

$P_k = 2 \times 5 - 9$

$= 10 - 9$

$\boxed{P_k = 1}$.

Step 6: Check P_k and accordingly put into Condition.

$P_k \geq 0$ ... Condition 2

$1 \geq 0$

{
$x = x + 1$
$y = y + 1$
$P_k = P_k + 2\Delta y - 2\Delta x$
}

$x = 6 + 1$ $\boxed{x = 7}$

$y = 5 + 1$ $\boxed{y = 6}$

$\therefore (7,6)$ ..... (2)

$P_k + 2\Delta y - 2\Delta x$

$1 + 2 \times 5 - 2 \times 9$

$21 - 18$

~~$P_k = 3$~~ $\boxed{P_k = -7}$

Scanned with CamScanner

Check condition.

$P_k \geq 0$

$3 \geq 0$ ... condition 2

so,

$x = x+1$

$y = y+1$

$P_k = P_k + 2\Delta y - 2\Delta x$

$3 + 2 \times 5 - 2(9)$

$x = 7+1$ $\boxed{x=8}$

$y = 6+1$ $\boxed{y=7}$

so, $(8, 7)$ — 3

Check condition

$P_k \leq 0$ ... C1

$-7 < 0$

$x = x+1$

$y = y$

$P_k = P_k + 2\Delta y$

$-7 + 2 \times 5$

$\boxed{P_k = 3}$

$x = 7+1$

$\boxed{y=6}$

$\therefore (8, 6)$

Check condition

$P_k \geq 0$ ... C2

$3 \geq 0$

so

$x = x+1$

$y = y+1$

$P_k = P_k + 2\Delta y - 2\Delta x$

$3 + 2(5) - 2(9)$

$3 + 10 - 18$

$13 - 18$

$\boxed{P_k = -5}$

$\boxed{x=9}$

$\boxed{y=7}$

$\therefore (9, 7)$ — 4

Check P_k

$P_k < 0$

$P_k (-5) < 0$ ... C1

$x = x+1$

$y = y$

$P_k = P_k + 2\Delta y$

$-5 + 2(5)$

$\boxed{P_k = 0}$

$\boxed{x=10}$

$\boxed{y=7}$

$(10, 7)$ — 5

Check P_k

$P_k \geq 0$ — C2

$0 \geq 0$

$x = x+1$

$y = y+1$

$P_k = P_k + 2\Delta y - 2\Delta x$

$0 + 2(5) - 2(9)$

$= 10 - 18$

$\boxed{P_k = -8}$

$\boxed{x=11}$

$\boxed{y=8}$

$(11, 8)$ — 6

Check P_k,

$P_k < 0$ — C1

$x = x+1$

$y = y$

$P_k = P_k + 2\Delta y$

$-8 + 2(5)$

$-8 + 10$

$\boxed{P_k = 2}$

$\boxed{x=12}$

$\boxed{y=8}$

$(12, 8)$ — 7

Scanned with CamScanner

condition.

≥ 0

≥ 0

$x = x+1$
$y = y+1$
$P_k = P_k + 2\Delta y - 2\Delta x$
$\}$

$P_k = 2 + 2(5) - 2(9)$
$= 2 + 10 - 18$
$= 12 - 18$
$\boxed{P_k = -6}$

$\boxed{x = 13}$
$\boxed{y = 9}$

(13, 9)

8

check C,
$P_k < 0$
$\}$
$x = x-1$
$y = y$
$\}$

$\boxed{x = 14}$
$\boxed{y = 9}$

$P_k + 2\Delta y$ (14, 9)
$-6 + 2(5)$
$-6 + 10$
$\boxed{P_k = 4}$

9

$P_k > 0$

$\boxed{x = 15}$
$\boxed{y = 10}$

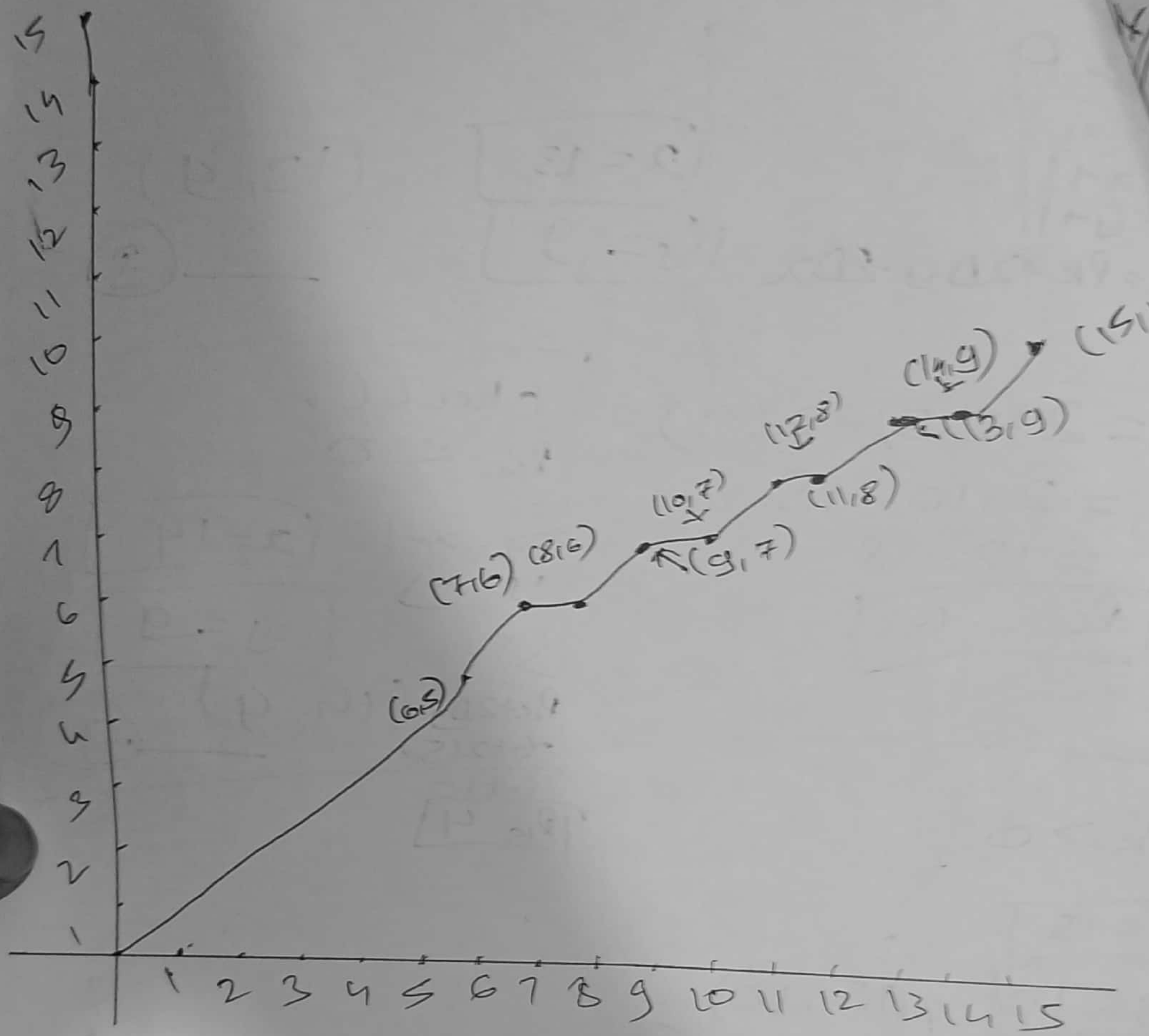
(15, 10)

10

Step 7: Repeat step (Dx) times

= 9 times

Step8 : ~~Step~~ Plot graph from (1), (2), (3)



(12, 9)

: $x_1 = 6$ $\quad x_2 = 12$

$y_1 = 6$ $\quad y_2 = 9$

Step 2: $\Delta x = 12 - 6 = 6$

$\Delta y = 9 - 6 = 3$

$\boxed{\Delta x = 6}$

$\boxed{\Delta y = 3}$

$(\Delta x = x_2 - x_1, \Delta y = y_2 - y_1)$

Step 3: $x = 6, y = 6$

Step 4: $\text{plot}(x = x_1, y = y_1)$

Plot (x, y) i.e $(6, 6)$ ..... (1)

Step 5: $\boxed{P_k = 2\Delta y - \Delta y}$

$P_k = 2(3) - 6$

$= 6 - 6$

$\boxed{P_k = 0}$

Step 6: Check condition, perform ~~addor~~ accordingly.

$P_k \geq 0$ ... condition (2)

$x = x + 1$ $\quad x = 6 + 1$ $\quad \boxed{x = 7}$

$y = y + 1$ $\quad y = 6 + 1$ $\quad \boxed{y = 7}$

$\therefore$ plot $(7, 7)$ ..... (2)

$P_k = P_k + 2\Delta y - 2\Delta x$

$= 0 + 2(3) - 6 \; (2 \times 6)$

$= 6 - 12$

$\boxed{P_k = -6}$

Scanned with CamScanner

< 0 .... Condition 1

$x = x+1$ $\quad x = 7+1$ $\quad \boxed{x = 8}$

$y = y$ $\quad y = 7$ $\quad \boxed{y = 7}$

plot(8,7) 3

$P_K = P_K + 2\Delta y$

$= -6 + 2(3)$

$= -6 + 6$

$\boxed{P_K = 0}$

Check,

$P_K \geq 0$ .... Condition 2

$x = x+1$ $\quad x = 8+1$ $\quad \boxed{x = 9}$

$y = y+1$ $\quad y = 7+1$ $\quad \boxed{y = 8}$

plot(9,8) 4

$P_K = P_K + 2\Delta y - 2\Delta x$

$= 0 + 2(3) - 2(6)$

$= 6 - 12$

$\boxed{P_K = -6}$

Check,

$P_K < 0$ ... Condition 1

$x = x+1$ $\quad x = 9+1$ $\quad \boxed{x = 10}$

$y = y$ $\quad y = 8$ $\quad \boxed{y = 8}$

plot(10,8) 5

$P_K = P_K + 2\Delta y$

$= -6 + 2(3)$

$\boxed{P_K = 0}$

Scanned with CamScanner

$P_k \geq 0$

$x = x+1$ .... Condition 2

$y = y+1$

$x = 10+1$

$y = 8+1$

$\boxed{x = 11}$

$\boxed{y = 9}$

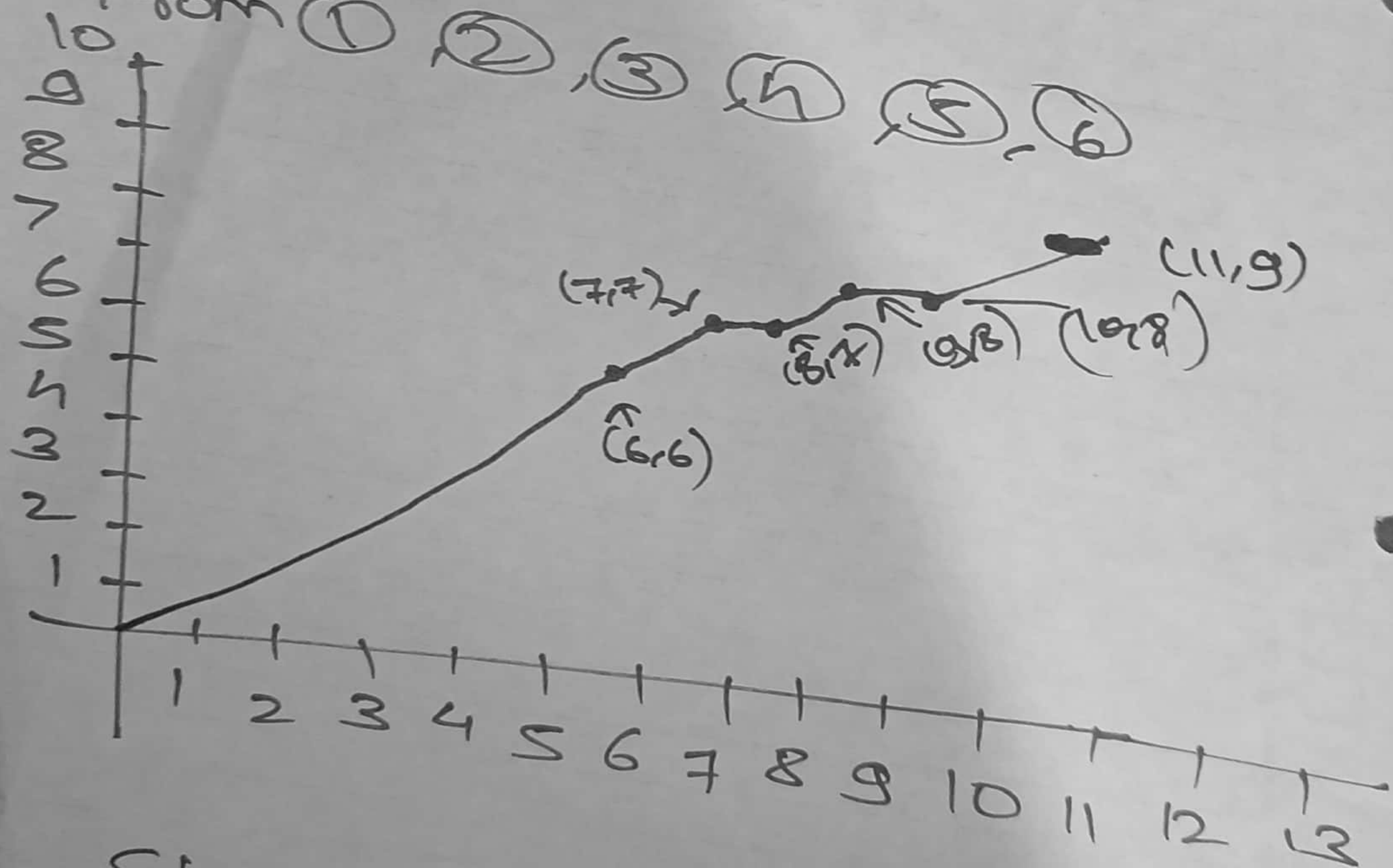
plot (11, 9) ..... (6)

Step 7: Repeat (Δx) times

$\therefore$ 6 times.

Step 8: plot graph

From (1), (2), (3), (4), (5), (6)



Step 9: End