WINTER– 18 EXAMINATION

Subject Name: Computer Graphics **Model Answer** Subject Code: 22318

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No.	Sub Q. N.	Answer	Marking Scheme
1		**Attempt any FIVE of the following:**	**10 M**
	a	**Define:** **(i)Pixel** **(ii)Frame Buffer**	2 M
	Ans	• Pixel Pixel or Pel is defined as "the smallest addressable screen element". **OR** A pixel may be defined as the smallest size object or color spot that can be displayed and addressed on a monitor. • 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.	1 M each for correct definition

b **Give the characteristics of display adaptor.** — 2 M

Ans

The characteristics of common display adapters are given in Table. The present-day display adapter supports all the modes of the preceding display adapters

2M for any relevant characteristics

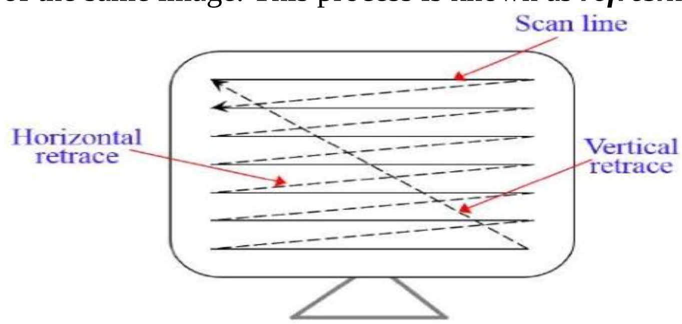
Driver selected	Mode constant	Display mode
CGA	CGAC0	320 × 200, 4 colour, palette 0
	CGAC1	320 × 200, 4 colour, palette 1
	CGAC2	320 × 200, 4 colour, palette 2
	CGAC3	320 × 200, 4 colour, palette 3
	CGSHI	640 × 200, 2 colour
EGA	EGALO	640 × 200, 16 colour
	EGAHI	640 × 350, 16 colour
VGA	VGALO	640 × 200, 16 colour
	VGAMED	640 × 350, 16 colour
	VGAHI	640 × 480, 16 colour

c **Explain Raster Scan** — 2 M

Ans

- In Raster scan, the electron beam from electron gun is swept horizontally across the phosphor one row at time from top to bottom.
- The electron beam sweeps back and forth from left to right across the screen. The beam is on, while it moves from left to right. The beam is off, when it moves back from right to left. This phenomenon is called the ***horizontal retrace.***
- As soon as the beam reaches the bottom of the screen, it is turned off and is rapidly retraced back to the top to start again. This is called the ***vertical retrace***.
- Raster scan displays maintain the steady image on the screen by repeating scanning of the same image. This process is known as ***refreshing of screen***.

2 M for correct explanation



Raster Scan CRT

d **State two line drawing algorithms.** — 2 M

Ans

Digital Differential Analyzer (DDA) Algorithm

Digital Differential Analyzer algorithm generates a line from differential equations of line

1 M for each Algorithm

and hence the name DDA.

Bresenham's Algorithm

The Bresenham algorithm is another line drawing algorithm which uses integer calculations for drawing line.

e **List types of Polygon** — 2 M

Ans Polygon can be of two types:- (1 M each)

- Convex polygon
- Concave polygon

f **List various polygon filling algorithms** — 2 M

Ans Various polygon filling algorithms are: (1 M each, Any two)

- Flood Fill Algorithm
- Boundary Fill Algorithm
- Scan Line Algorithm

g **Give matrix representation for 2D scaling** — 2 M

Ans Let us assume that the original co-ordinates are (X, Y), the scaling factors are (S_X, S_Y), and the produced co-ordinates are (X', Y'). This can be mathematically represented as shown below: (2 M for proper Matrix)

$$X' = X \cdot S_X \text{ and } Y' = Y \cdot S_Y$$

The scaling factor S_X, S_Y scales the object in X and Y direction respectively. The above equations can also be represented in matrix form as below:

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

2 **Attempt any THREE of the following:** — **12 M**

a **Differentiate between Random Scan and Raster Scan.** — 4 M

Ans (Any four points: 1 mark each)

Random Scan Display	Raster Scan Display
In vector scan display the beam is moved between the end points of the graphics primitives.	In raster scan display the beam is moved all over the screen one scan at a time, from top to bottom and then back to top.
Vector display flickers when the number of primitives in the buffer becomes too large.	In raster display, the refresh process is independent of the complexity of the image.

Scan conversion is not required.	Graphics primitives are specified in terms of their endpoints and must be scan converted into their corresponding pixels in the frame buffer.
Scan conversion hardware is not required.	Because each primitive must be scan converted real time dynamics is far more computational and requires separate scan conversion hardware.
Vector display draws continuous and smooth lines.	Raster display can display mathematically smooth lines, polygons and boundaries of curves primitives only by approximating them with pixels on the raster grid.
Mathematical functions are used to draw an image.	Screen points/pixels are used to draw an image.
It does not user interlacing.	It uses interlacing.
Editing is easy.	Editing is difficult.
Cost is more	Cost is low
Vector display only draws lines and characters.	Raster display has ability to display areas filled with solid colors or patterns.
Resolution is good because this system produces smooth lines drawings because CRT beam directly follows the line path.	Resolution is poor because raster system in contrast produces zigzag lines that are plotted as discrete point sets.
Picture definition is stored as a set of line drawing instructions in a display file.	Picture definition is stored as a set of intensity values for all screen points, called pixels in a refresh buffer area.
They are more suited to line drawing application e.g. CRO and pen plotter.	They are more suited to geometric area drawing applications e.g. monitors, TV
It uses beam-penetration method.	It uses shadow-mask method

b **Explain and write steps for DDA line drawing algorithm.** 4 M

Ans (Explanation 2M, Algorithm 2M)

- 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 methods.

Algorithm:

Steps 1: Read the end points of line (x1,y1) and (x2,y2).

Steps 2: Δx = abs $(x_2 - x_1)$ and
Δy = abs $(y_2 - y_1)$

Step 3: if $\Delta x \geq \Delta y$ then
length = Δx
else
length = Δy
end if

Step 4: $\Delta x = (x_2 - x_1)$/length

Step 5: $\Delta y = (y_2 - y_1)$/length

Step 6: x= x_1 + 0.5 * sign (Δx)
y= y_1 + 0.5 * sign (Δy)

Step 7: i = 1

```
while (i ≤ length)
{
    plot (integer (x), integer (y))
    x = x + Δx
    y = y + Δy
    i = i + 1
}
```

Step 8: End

c	**List out basic transformation techniques. Explain scaling transformation with respect to 2D.**		4 M
Ans			Listing 1M, Explanation 3M

Basic transformations techniques are:

- Translation
- Scaling
- Rotation

Scaling Transformation

- Scaling means to change the size of object. This change can either be positive or negative.
- To change the size of an object, scaling transformation is used. In the scaling process, you either expand or compress the dimensions of the object.
- Scaling can be achieved by multiplying the original co-ordinates of the object with the scaling factor to get the desired result.
- Let us assume that the original co-ordinates are (X, Y), the scaling factors are (S_X, S_Y), and the produced co-ordinates are (X', Y'). This can be mathematically represented as shown below:

$X' = X \cdot S_X$ and $Y' = Y \cdot S_Y$

- The scaling factor S_X, S_Y scales the object in X and Y direction respectively. The above equations can also be represented in matrix form as below:

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

OR

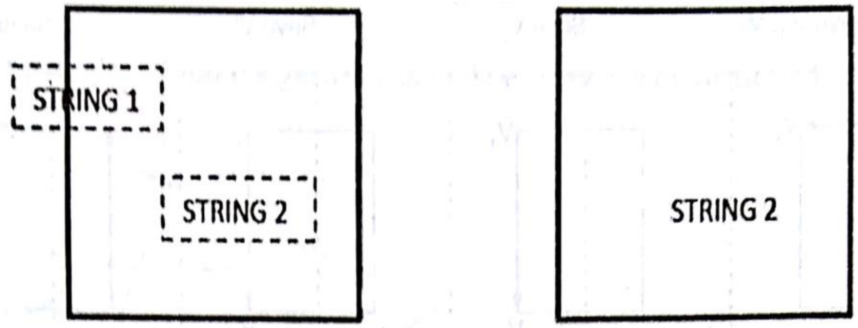
$$P' = P \cdot S$$

Where, S is the scaling matrix.

(a) Before Scaling **(b) After Scaling**

- If we provide values less than 1 to the scaling factor S, then we can reduce the size of the object. If we provide values greater than 1, then we can increase the size of the object.

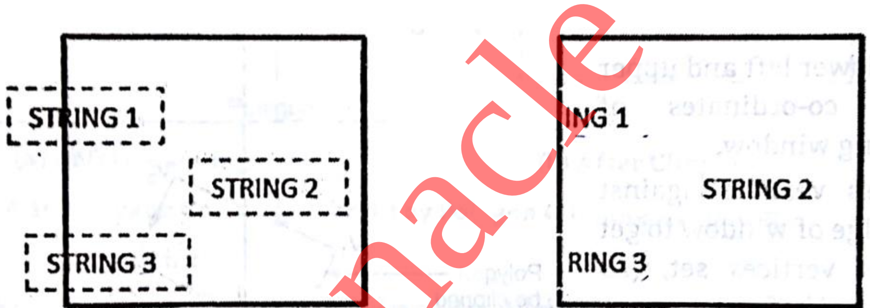
	d	**Explain differ types of Text clipping in brief.**	4 M
	Ans	Many techniques are used to provide text clipping in a computer graphics. It depends on the methods used to generate characters and the requirements of a particular application. There are three methods for text clipping which are listed below − 1) All or none string clipping 2) All or none character clipping 3) Text clipping The following figure shows all or none string clipping −	Explanation of 3 methods with diagrams 4 marks



(a) Before Clipping (b) After Clipping

In all or none string clipping method, either we keep the entire string or we reject entire string based on the clipping window. As shown in the above figure, Hello2 is entirely inside the clipping window so we keep it and Hello1 being only partially inside the window, we reject.

The following figure shows all or none character clipping –



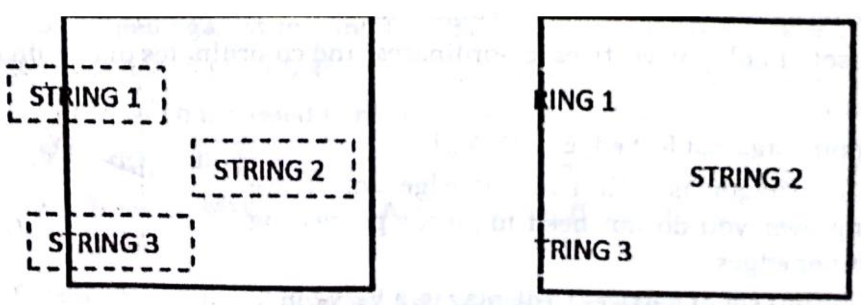
(a) Before Clipping (b) After Clipping

This clipping method is based on characters rather than entire string. In this method if the string is entirely inside the clipping window, then we keep it. If it is partially outside the window, then −

You reject only the portion of the string being outside

If the character is on the boundary of the clipping window, then we discard that entire character and keep the rest string.

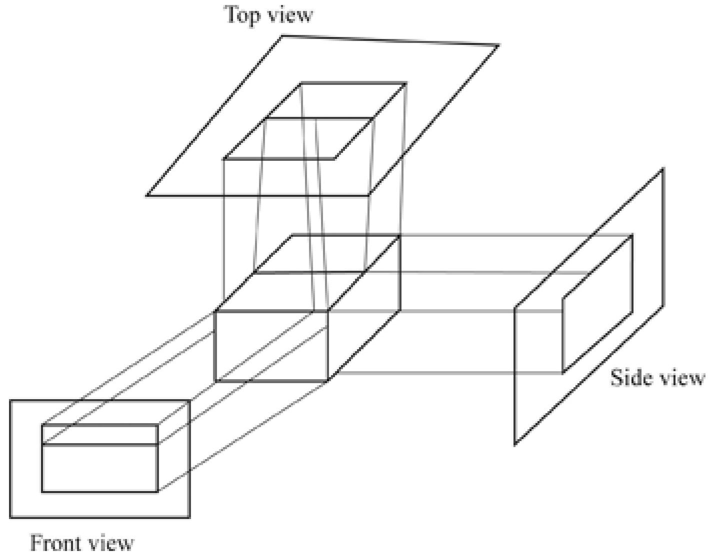
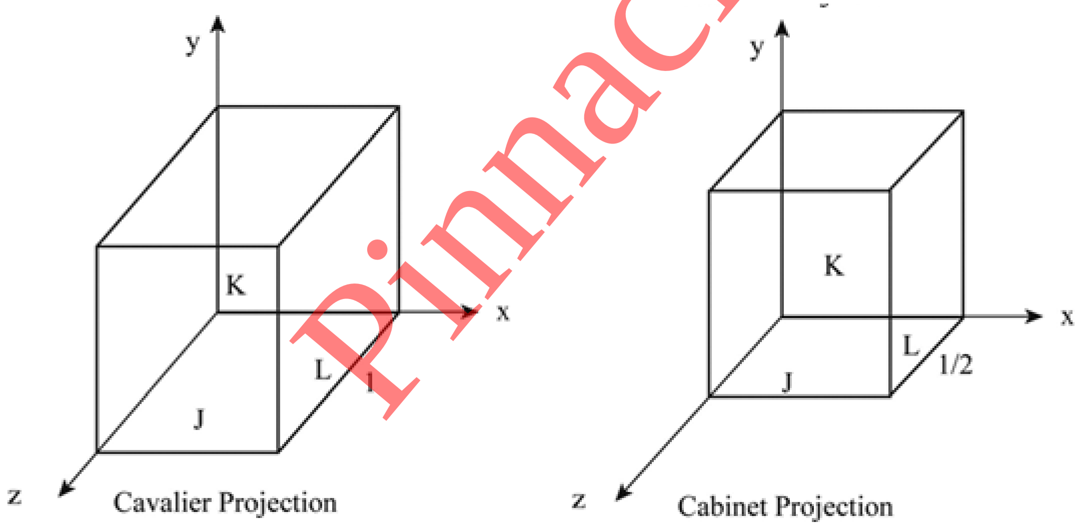
The following figure shows text clipping –



(a) Before Clipping (b) After Clipping

This clipping method is based on characters rather than the entire string. In this method if the string is entirely inside the clipping window, then we keep it. If it is partially outside the window, then you reject only the portion of string being

		outside. If the character is on the boundary of the clipping window, then we discard only that portion of character that is outside of the clipping window.	
3		**Attempt any THREE of the following:**	**12 M**
	a	**Explain stroke method and Bitmap method with example.**	4M
	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. **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 bitmatrix 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.	Stroke Method 2 Marks; Bitmap Method 2 Marks
	b	**Explain types of Parallel Projection with example.**	4M
	Ans	• Orthographic projection – the projection direction is a normal one to the plane and it is categorized as ○ Top projection ○ Front projection ○ Side projection	Orthographic projection 2 marks; Oblique projection 2 Marks

		 • Oblique projection – the projection direction is not a normal one to the plane; it gives a better view and it is categorized as ○ Cavalier projection ○ Cabinet projection 	
	c	**Write down Cohen-Sutherland Line clipping algorithm.**	4M
	Ans	Step 1: Scan end points for the line P1(x1, y1) and P2(x2, y2) Step 2: Scan corners for the window as (Wx1, Wy1) and (Wx2, Wy2) Step 3: Assign the region codes for endpoints P1 and P2 by Bit 1 - if (x < Wx1) Bit 2 - if (x < Wx2) Bit 3 - if (x < Wy2)	Correct algorithm 4 Marks

Bit 4 - if (x < Wy1)

Step 4: Check for visibility of line P1, P2

- If region codes for both end points are zero then the line is visible, draw it and jump to step 9.
- If region codes for end points are not zero and the logical and operation of them is also not zero then the line is invisible, reject it and jump to step 9.
- If region codes for end points does not satisfies the condition in 4(i) and 4(ii) then line is partly visible.

Step 5: Determine the intersecting edge of the clipping window by inspecting the region codes for endpoints.

- If region codes for both the end points are non-zero, find intersection points P1 and P2 with boundary edges of clipping window with respect to point P1 and P2.
- If region code for any one end point is non zero then find intersection point P1 or P2 with the boundary edge of the clipping window with respect to it.

Step 6: Divide the line segments by considering intersection points.

Step 7: Reject the line segment if any of the end point of it appear outside the window.

Step 8: Draw the remaining line.

Step 9: Exit

d **Explain Koch curve with diagram.** 4M

Ans Koch Curve: - In Koch curve, begin at a line segment. Divide it into third and replace the center by the two adjacent sides of an equilateral triangle as shown below.

(Description 3 Marks; Diagram 1 Mark)

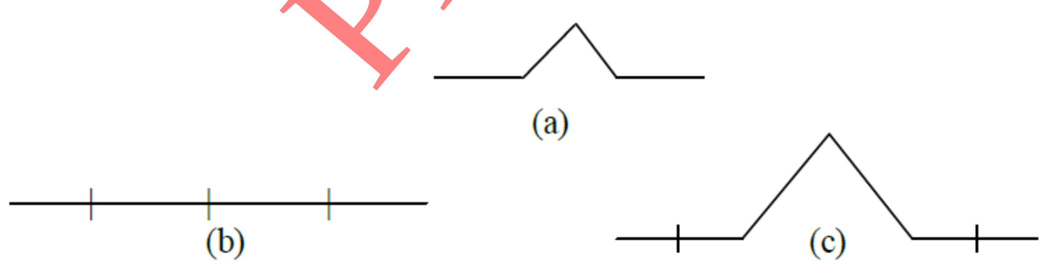


Fig 6.3 Replacement of Line Segment for Koch Curve

This will give the curve which starts and ends at same place as the original segment but is built of 4 equal length segments, with each 1/3rd of the original length. So the new curve has 4/3 the length of original segments. Repeat same process for each of the 4 segment which will give curve more wiggles and its length become 16/9 times the original. Suppose repeating the replacements indefinitely, since each repetition increases the length by a factor of 4/3, the length of the curve will be infinite but it is folded in lots of tiny wiggles.

4 **Attempt any THREE of the following:** **12 M**

a **Compare Bitmap Graphics and Vector based graphics.** 4 M

Ans

Bitmap Graphics	Vector Based Graphic
It is pixel based image	It is Mathematical based image
Images are resolution dependent.	Images are formula based / dependent.
These images are not easily scalable.	Easily scalable with the help of formula.
Poor quality of image as oppose to Vector based Graphics.	Better image quality as compare to Bitmap Graphics.
Size of image is high.	Size of image is low.

Any 4 Points of comparison; 1 Mark each

b **Consider line from (4, 4) to (12 9). Use Bresenham's algorithm to rasterize this line.** 4 M

Ans

x1 = 4 | y1 = 4 | & | x2 = 12 | y2 = 9	
Calculation	**Result**
dx = abs(x1 - x2)	8 = abs(4 - 12)
dy = abs(y1 - y2)	5 = abs(4 - 9)
p = 2 * (dy - dx)	-6 = 2 * (5 - 8)
ELSE	x = x1 | y = y1 | end = x2
	x = 4 | y = 4 | end = 12

STEP	while(x < end)	x = x + 1	if(p < 0) { p = p + 2 * dy } else{ p = p + 2 * (dy - dx) }	OUTPUT
1	5 < 12	5 = 4 + 1	IF 4 = -6 + 2 * 5	x = 5 | y = 4
2	6 < 12	6 = 5 + 1	ELSE -2 = 4 + 2 * (5 - 8)	x = 6 | y = 5
3	7 < 12	7 = 6 + 1	IF 8 = -2 + 2 * 5	x = 7 | y = 5

Any Suitable method can be consider

Correct steps and result: 4 Marks

4	8 < 12	8 = 7 + 1	ELSE 2 = 8 + 2 * (5 - 8)	x = 8 | y = 6
5	9 < 12	9 = 8 + 1	ELSE -4 = 2 + 2 * (5 - 8)	x = 9 | y = 7
6	10 < 12	10 = 9 + 1	IF 6 = -4 + 2 * 5	x = 10 | y = 7
7	11 < 12	11 = 10 + 1	ELSE 0 = 6 + 2 * (5 - 8)	x = 11 | y = 8
8	12 < 12	12 = 11 + 1	ELSE -6 = 0 + 2 * (5 - 8)	x = 12 | y = 9

c **Use Cohen-Sutherland algorithm to clip two lines PI (40, 15) -- P2 (75. 45) and P3 (70, 20) — P4 (100, 10) against a window A (50, 10), B (80, 10). C(80, 40) & D(50,40)** — 4 M

Ans

Solution :

Line 1 : P1 (40, 15) - P2 (75, 45) Wxi = 50 Wy2 = 40 Wx2 = 80 Wy2 = 10

Point	Endeode	ANDing	
P1	0001	0000	(Partially visible)
P2	0000		

$y_1 = m(x_L – x) + y = \frac{6}{7}(50\text{-}40)+15$ $\quad m = \frac{45-15}{75-40}$

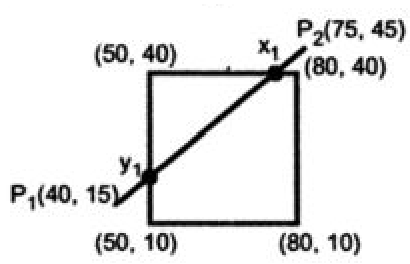
$= 23.57$

$x_1 = \frac{1}{m} (y_T – y) + x = \frac{7}{6}(40\text{-}50)+40 = 69.16$

$y_2 = m(x_R – x) + y = \frac{6}{7}(80\text{-}40)+15 = 49.28$

$x_2 = \frac{1}{m} (y_B – y) + x = \frac{7}{6}(10\text{-}15)+40 = 34.16$

Hence:



Any suitable method can be consider

Computation for Line 1: 2 Marks; Computation for Line 2 : 2 Marks

Line 2 : P3 (70,20) – P4 (100,10) Wxi = 50 Wy2 = 40 Wx2 = 80 Wy2 = 10

Point	Endeode	ANDing	
P3	0000	0000	(Partially visible)
P4	0010		

Slope $m` = \frac{10-20}{100-70} = \frac{-10}{30} = \frac{-1}{3}$

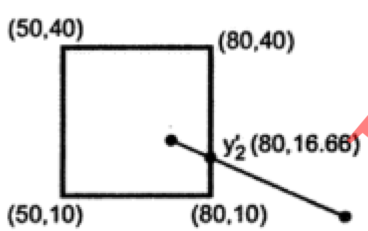
$y`_1 = m(x_L - x) + y = \frac{-1}{3}(50-70)+20 = 26.66$

$x`_1 = \frac{1}{m}(y_T - y) + x = -3\,(40-20)+70 = 10$

$y`_2 = m(x_R - x) + y = \frac{-1}{3}(80-70)+20 = 16.66$

$x`_2 = \frac{1}{m}(y_B - y) + x = -3\,(10-20)+70 = 100$

Hence:



d **Consider the square A (1, 0), B (0, 0), C (O, 1), D (1, I). Rotate the square ABCD by 45° anticlockwise about point A (1. 0).** 4 M

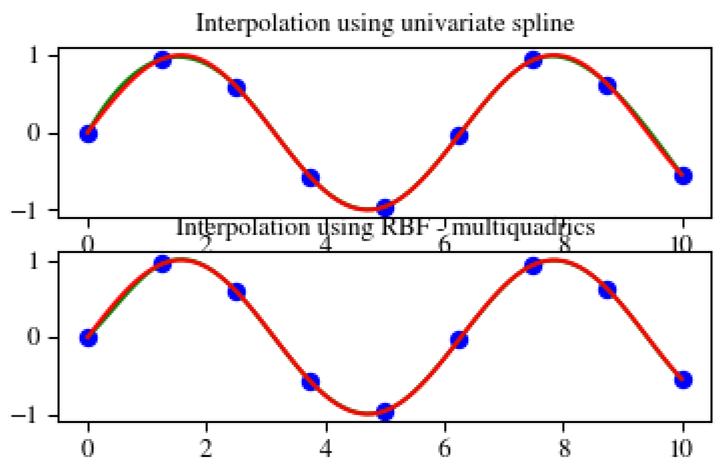
Ans

$$\begin{bmatrix} \cos\theta & \sin\theta\theta & 0 \\ -\sin\theta\theta & \cos\theta & 0 \\ -Xp\cos\theta + Yp\sin\theta + Xp & -Xp\sin\theta - Yp\cos\theta + Yp & 1 \end{bmatrix}$$

Matrix formation 2 Marks; Matrix calculation 2 Marks

Here, θ = 45°, $X_p = 1$ $Y_p = 0$

$$[T_1.R.T_2] = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} + 1 & -\frac{1}{\sqrt{2}} & 1 \end{bmatrix}$$

		$\begin{bmatrix} A' \\ B' \\ C' \\ D' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ \frac{-1}{\sqrt{2}}+1 & -\frac{1}{\sqrt{2}} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ \frac{-1}{\sqrt{2}}+1 & -\frac{1}{\sqrt{2}} & 1 \\ 1-\sqrt{2} & 0 & 1 \\ 1-\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1 \end{bmatrix}$	
	e	**Explain curve generation using Interpolation technique.**	4 M
	Ans	Specify a spline curve by giving a set of coordinate positions, called control points, which indicates the general shape of the curve These, control points are then fitted with piecewise continuous parametric polynomial functions in one of two ways. When polynomial sections are fitted so that the curve passes through each control point, the resulting curve is said to interpolate the set of control points. On the other hand, when the polynomials are fitted to the general control-point path without necessarily passing through any control point, the resulting curve is said to approximate the set of control points interpolation curves are commonly used to digitize drawings or to specify animation paths. Approximation curves are primarily used as design tools to structure object surfaces an approximation spline surface credited for a design application. Straight lines connect the control-point positions above the surface. 	Description 2 Marks; Example/Diagram 2 Marks

5		**Attempt any two of the following:**	**12 M**
	a	**Rotate a triangle defined by A(0,0), B(6,0), & C(3,3) by 90^0 about origin in anti-clock wise direction**	6 M

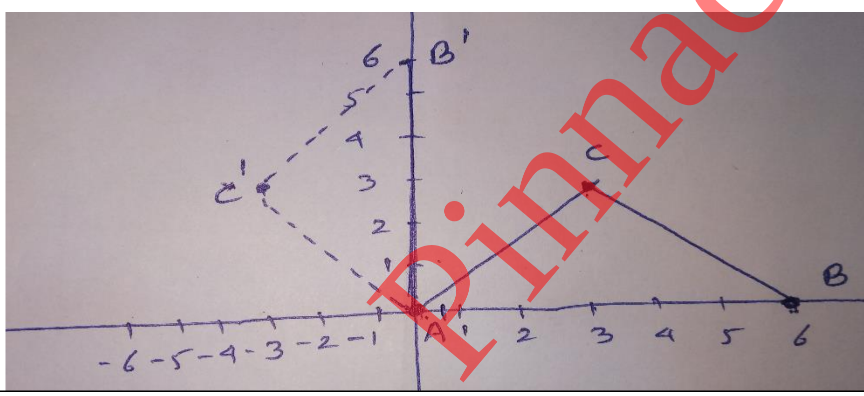
Ans

The new position of point A (0, 0) will become A' (0,0)
The new position of point B (6,0) will become B' (0, 6)
The new position of point C (3, 3) will become C' (-3, 3)

$$\begin{bmatrix} x' \\ y' \\ \omega' \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} x \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 \\ 6 & 0 & 1 \\ 3 & 3 & 1 \end{bmatrix} x \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 6 & 1 \\ -3 & 3 & 1 \end{bmatrix}$$



Matrix 2 Marks

Correct answer 4 marks

	b	**Explain boundary fill algorithm with pseudo code. Also mention its limitations if any.**	6 M

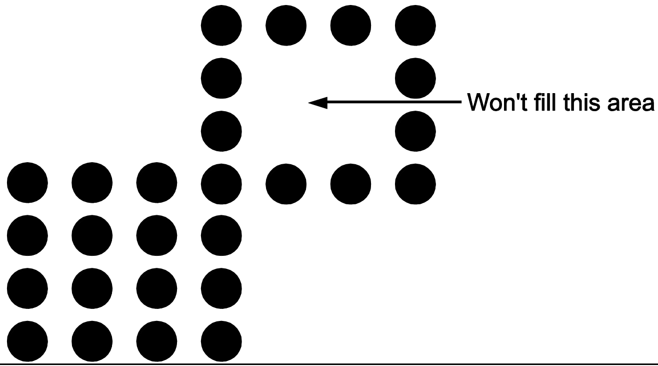
Ans

```
Procedure : boundary_fill (x, y, f_colour, b_colour)
        {
          if (getpixel (x,y) ! = b_colour && getpixel (x, y) ! = f_colour)
            {       putpixel (x, y, f_colour)
                    boundary_fill (x + 1, y, f_colour, b_colour);
                    boundary_fill (x, y + 1, f_colour, b_colour);
                    boundary_fill (x – 1, y, f_colour, b_colour);
                    boundary_fill (x, y – 1, f_colour, b_colour);
            }
        }
```

Limitations:

- There is a problem with this technique. Consider the case following, where we tried to fill the

4m algorithm, 2m for limitations

		entire region. Here, the image is filled only partially. In such cases, 4-connected pixels technique cannot be used. 	
	c	**obtain the curve parameters for drawing a smooth Bezier curve for the following points A(0,10), B(10,50), C(70,40) &D(70,-20)**	6 M

Ans

$A(0,10),\ B(10,50),\ C(70,40),\ D(70,-20)$

$P(u) = (1-u^3)P_1 + 3u(1-u^2)P_2 + 3u^2(1-u)P_3 + u^3P_4$

$u = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$

$P(0) = P_1 = (0,10)$

$P(\frac{1}{4}) = (1-\frac{1}{4})^3P_1 + 3\frac{1}{4}(1-\frac{1}{4})^2 + 3(\frac{1}{4})^2(1-\frac{1}{4})P_3 + (\frac{1}{4})^3P_4$

$= \frac{27}{64}(0,10) + \frac{27}{64}(10,50) + \frac{9}{64}(70,40) + \frac{1}{64}(70,-20)$

$= [\frac{27}{64}\times 0 + \frac{27}{64}\times 10 + \frac{9}{64}\times 70 + \frac{1}{64}\times 70 ; \frac{27}{64}\times 10 + \frac{27}{64}\times 50 + \frac{9}{64}\times 40 + \frac{1}{64}\times -20]$

$= [0 + \frac{270}{64} + \frac{630}{64} + \frac{70}{64},\ \frac{270}{64} + \frac{135}{64} + \frac{360}{64} - \frac{20}{64}]$

$= [\frac{970}{64}, \frac{745}{64}] = (15.15, 11.64)$

$P(\frac{1}{2}) = (1-\frac{1}{2})^3P_1 + 3\frac{1}{2}(1-\frac{1}{2})^2P_2 + 3(\frac{1}{2})^2(1-\frac{1}{2})P_3 + (\frac{1}{2})^3P_4$

$= (\frac{1}{8})(0,10) + \frac{3}{8}(10,50) + \frac{3}{8}(70,40) + \frac{1}{8}(70,-20)$

$= (\frac{1}{8}\times 0 + \frac{3}{8}\times 10 + \frac{3}{8}\times 70 + \frac{1}{8}\times 70,$
$\frac{1}{8}\times 10 + \frac{3}{8}\times 50 + \frac{3}{8}\times 40 + \frac{1}{8}\times 40)$

$= (\frac{30}{8} + \frac{210}{8} + \frac{70}{8},\ \frac{10}{8} + \frac{150}{8} + \frac{120}{8} + \frac{-20}{8})$

$= (\frac{310}{8}, \frac{260}{8}) = (38.7, 32.5)$

Any correct method can be consider.

Calculation 3 Marks

Diagram 3Marks

$$P\left(\frac{3}{4}\right) = \left(1-\frac{3}{4}\right)^3 P_1 + 3\,\frac{3}{4}\left(1-\frac{3}{4}\right)^2 P_2 + 3\left(\frac{3}{4}\right)^2\left(1-\frac{3}{4}\right) P_3 + \left(\frac{3}{4}\right)^3 P_4$$

$$= \frac{1}{64}(0,10) + \frac{9}{64}(10,50) + \frac{27}{64}(70,40) + \frac{27}{64}(70,-20)$$

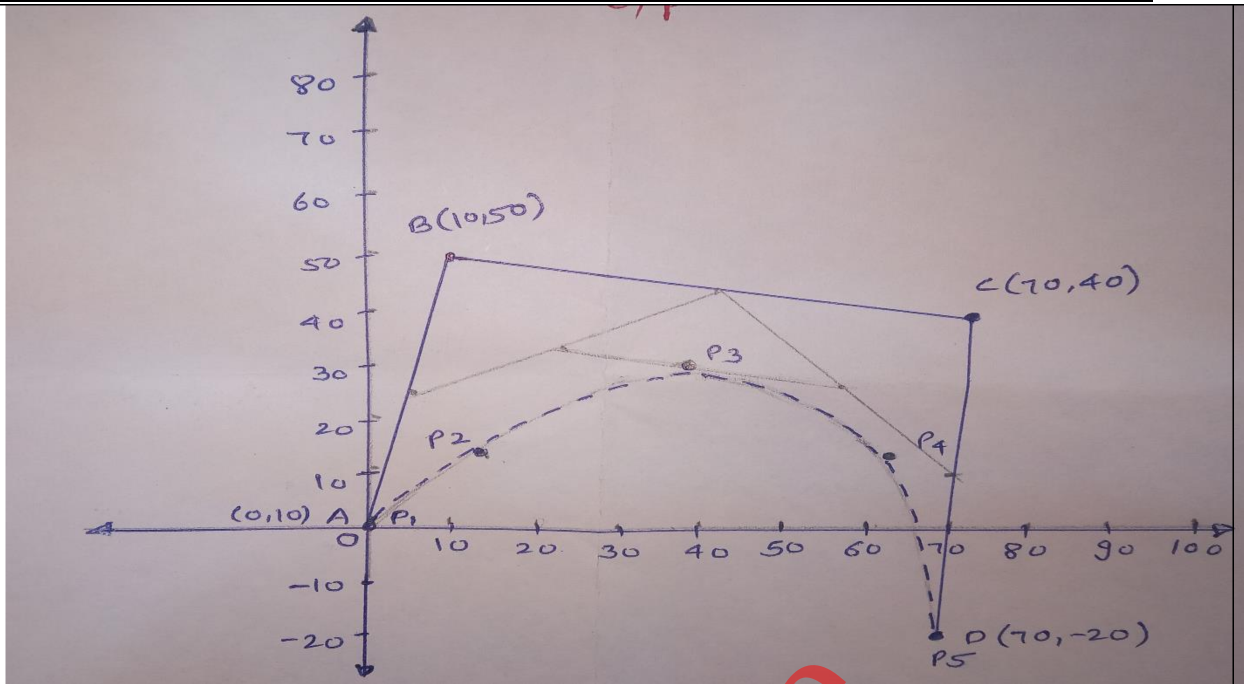
$$= \left(\frac{1}{64}\times 0 + \frac{9}{64}\times 10 + \frac{27}{64}\times 70 + \frac{27}{64}\times 70,\right.$$

$$\left.\frac{1}{64}\times 10 + \frac{9}{64}\times 50 + \frac{27}{64}\times 40 + \frac{27}{64}\times -20\right)$$

$$= \left(\frac{90}{64} + \frac{1890}{64} + \frac{1890}{64},\ \frac{10}{64} + \frac{450}{64} + \frac{1080}{64} - \frac{540}{64}\right)$$

$$= \underline{(60.46,\ 15.62)}$$

$$P(1) = \underline{(70,-20)}$$



OR

ITERATION 1:

Mid of AB = AB'

AB' = [(Ax + Bx)/2, (Ay + By)/2)]

= [(0+10)/2, (10+50)/2]

= [(10)/2, (60)/2]

= (5, 30)

Mid of BC = BC'

BC' = [(Bx + Cx)/2, (By + Cy)/2)]

= [(10+70)/2, (50+40)/2]

= [(80)/2, (90)/2]

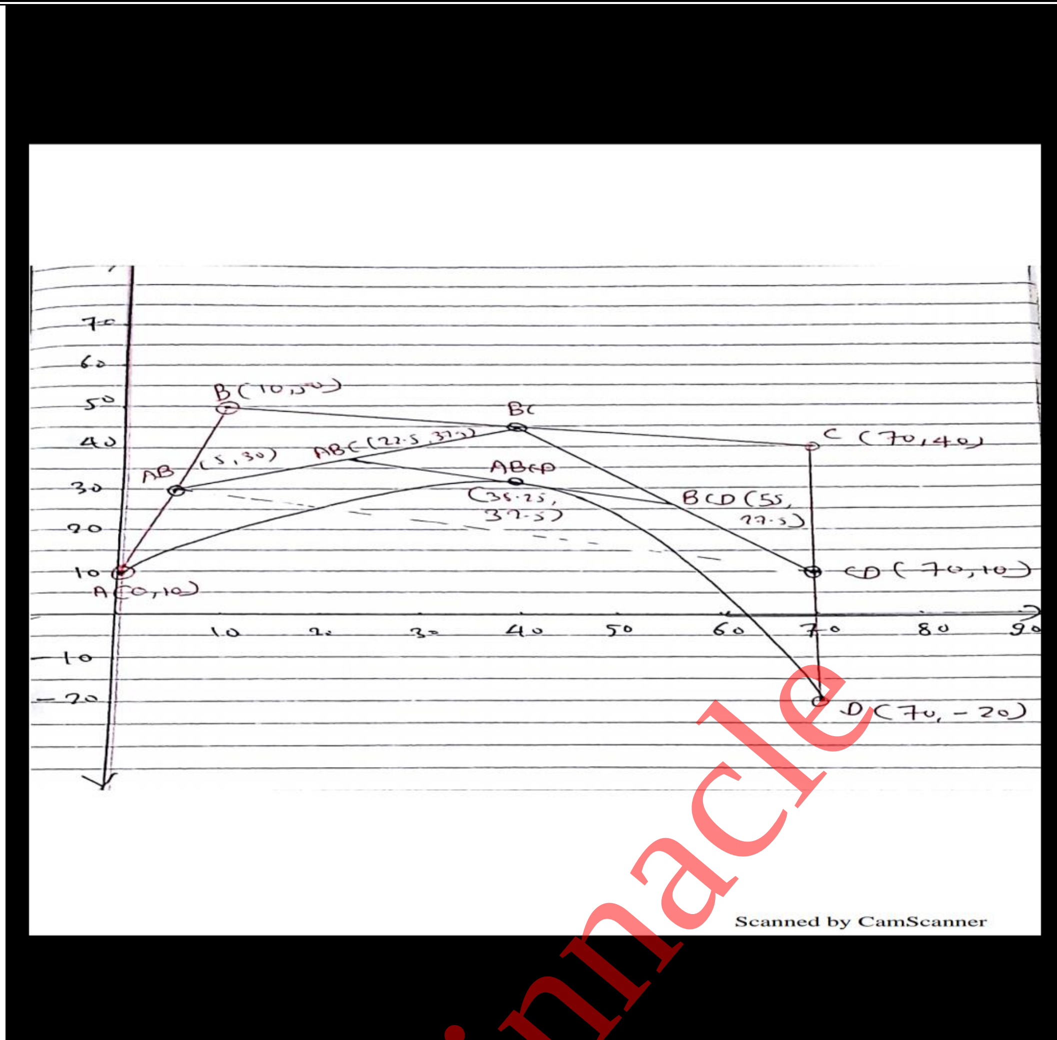
= (40, 45)

Mid of CD = CD'

CD' = [(Cx + Dx)/2, (Cy + Dy)/2)]

= [(70+70)/2, (40+(-20))/2]

		= [(140)/2, (20)/2] = (70, 10) ITERATION 2: Mid of ABC = ABC' ABC' = [(ABx + BCx)/2, (ABy + BCy)/2)] = [(5+40)/2, (30+45)/2] = [(45)/2, (75)/2] = (22.5, 37.5) Mid of BCD = BCD' BCD' = [(BCx + CDx)/2, (BCy + CDy)/2)] = [(40+70)/2, (45+10)/2] = [(110)/2, (55)/2] = (55, 27.5) ITERATION 3: Mid of ABCD = ABCD' ABCD' = [(ABCx + BCDx)/2, (ABCy + BCDy)/2)] = [(22.5+55)/2, (37.5+27.5)/2] = [(77.5)/2, (65)/2] = (38.25, 32.5)	



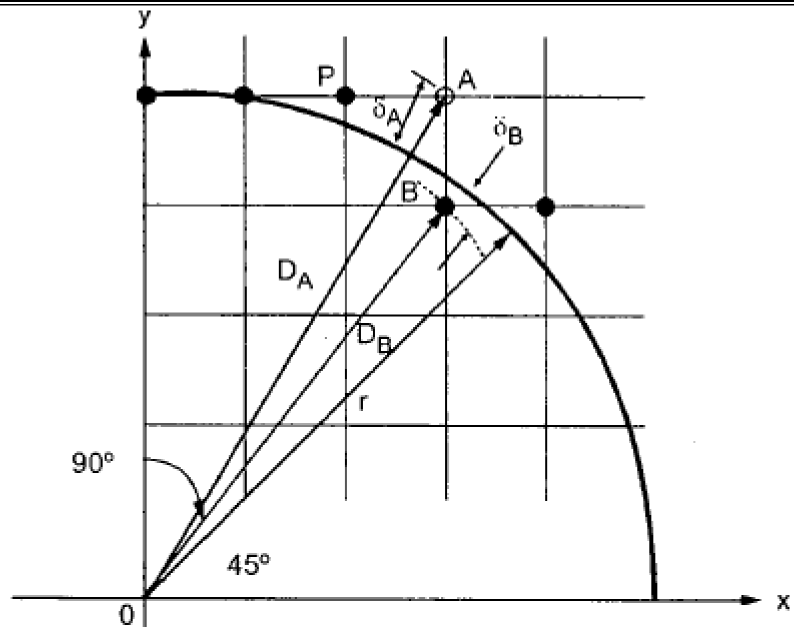
6		**Attempt any two of the following:**	**12 M**
	a	**Write matrices in homogeneous co-ordinates system for 3D scaling transformation.**	6M
	Ans	3D transformation matrix for scaling is as follows: $$S=\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$	Correct matrix 6 Marks

		It specifies three co-ordinates with their own scaling factors. If scale factors, $S_x = S_y = S_z = S > 1$ then the scaling is called as magnification. $S_x = S_y = S_z = S < 1$ then the scaling is called as reduction. Therefore, point after scaling with respect to origin can be calculated as, $\therefore \quad P = P \cdot S$	
	b	**Write down Cyrus-Beck line clipping algorithm.**	6M

	Ans	**Step 1:** Read end points of line P_1 and P_2.
		Step 2: Read vertex coordinates of clipping window.
		Step 3: Calculate $D = P_2 - P_1$.
		Step 4: Assign boundary point b with particular edge.
		Step 5: Find inner normal vector for corresponding edge.
		Step 6: Calculate D.n and $W = P_1 - b$
		Step 7: If D.n > 0
		$t_L = -(W.n)/(D.n)$
		else
		$t_U = -(W.n)/(D.n)$
		end if
		Step 8: Repeat steps 4 through 7 for each edge of clipping window.
		Step 9: Find maximum lower limit and minimum upper limit.
		Step 10: If maximum lower limit and minimum upper limit do not satisfy condition $0 \le t \le 1$ then ignore line.
		Step 11: Calculate intersection points by substituting values of maximum lower limit and minimum upper limit in parametric equation of line P_1P_2.
		Step 12: Draw line segment $P(t_L)$ to $P(t_U)$.
		Step 13: Stop.
			Correct algorithm 6 marks

	c	**Derive the expression for decision parameter used in Bresenhaum's circle drawing algorithm.**	6M

Ans



The distances of pixels A and B from the origin are given as

$$D_A = \sqrt{(x_{i+1})^2 + (y_i)^2} \quad \text{and}$$
$$D_B = \sqrt{(x_{i+1})^2 + (y_i - 1)^2}$$

Now, the distances of pixels A and B from the true circle are given as

$$\delta_A = D_A - r \text{ and } \delta_B = D_B - r$$

However, to avoid square root term in derivation of decision variable, i.e. to simplify the computation and to make algorithm more efficient the δ_A and δ_B are defined as

$$\delta_A = D_A^2 - r^2 \quad \text{and}$$
$$\delta_B = D_B^2 - r^2$$

From Fig. , we can observe that δ_A is always positive and δ_B always negative. Therefore, we can define **decision variable** d_i as

$$d_i = \delta_A + \delta_B$$

and we can say that, if $d_i < 0$, i.e., $\delta_A < \delta_B$ then only x is incremented; otherwise x is incremented in positive direction and y is incremented in negative direction. In other words we can write,

For $d_i < 0$, $\quad x_{i+1} = x_i + 1$ and

For $d_i \geq 0$, $\quad x_{i+1} = x_i + 1$ and $\quad y_{i+1} = y_i - 1$

The equation for d_i at starting point, i.e. at x = 0 and y = r can be simplified as follows

$$\begin{aligned} d_i &= \delta_A + \delta_B \\ &= (x_i + 1)^2 + (y_i)^2 - r^2 + (x_i + 1)^2 + (y_i - 1)^2 - r^2 \\ &= (0 + 1)^2 + (r)^2 - r^2 + (0 + 1)^2 + (r - 1)^2 - r^2 \\ &= 1 + r^2 - r^2 + 1 + r^2 - 2r + 1 - r^2 \\ &= 3 - 2r \end{aligned}$$

Similarly, the equations for d_{i+1} for both the cases are given as

For $d_i < 0$, $\quad d_{i+1} = d_i + 4x_i + 6 \quad$ and

For $d_i \leq 0$, $\quad d_{i+1} = d_i + 4(x_i - y_i) + 10$

Correct method and correct equation 6 Marks