1.What is Computer Graphics and Explain graphics software packages.

Computer Graphics: Computer graphics is a field of study and technology that deals with creating, manipulating, and displaying visual images and animations using computers. It involves the use of algorithms, mathematics, and computer programming to generate graphical representations of objects and scenes.

Raster Graphics Software:

Adobe Photoshop: A powerful raster graphics editor used for photo editing, image manipulation, and graphic design. GIMP (GNU Image Manipulation Program): A free and open-source raster graphics editor with features similar to Photoshop.

Vector Graphics Software: Adobe Illustrator: A vector graphics editor used for creating illustrations, logos, and other vector-based graphics.

CorelDRAW: A vector graphics suite for illustration, page layout, and photo editing.

3D Modeling and Animation Software:

Blender: An open-source 3D modeling and animation software used for creating 3D graphics, animations, and simulations.

Autodesk Maya: A professional 3D computer graphics application for modeling, simulation, and animation.

CAD Software: AutoCAD: A widely used computer-aided design (CAD) software for creating 2D and 3D drawings and models. SolidWorks: A 3D CAD software used for mechanical design, simulation, and product development.

2.Explain Graphical input output devices.

Graphical input-output devices are hardware components that facilitate the interaction between a user and a computer in the context of graphics. Here are some common graphical input-output devices:

Graphics Tablets: Input: Graphics tablets, also known as digitizing tablets or drawing tablets, allow users to draw or write directly on a surface using a stylus or digital pen. Output: The output is displayed on a computer screen in real-time, enabling artists and designers to create digital illustrations or perform precise tasks.

Digital Cameras and Scanners: Input: Digital cameras capture real-world images and convert them into digital format. Output: The digital images captured by cameras or scanners can be processed, edited, and displayed on a computer monitor.

Monitors: Output: Monitors, also known as display screens or screens, are output devices that present visual information generated by the computer's graphics processing unit (GPU

Printers and Plotters: Output: Printers produce hard copies of digital images or documents on paper or other media. Input: Printers and plotters receive digital data from the computer, interpreting it to produce the desired output.

3D Printers: Output: 3D printers create three-dimensional objects layer by layer from digital models. They use materials such as plastic, metal, or resin to build physical representations of digital designs.

Input: The input for 3D printers is a 3D model created using computer-aided design (CAD) software or obtained from other sources.

3.What is inside outside test? Explain odd even method.

Inside-Outside Test: The Inside-Outside Test is a method used to determine if a point lies inside or outside a closed geometric region, typically a polygon. The basic idea is to draw a straight line (ray) from the test point to infinity in any direction and count how many times the line intersects with the edges of the polygon. The number of intersections is then used to classify the point as either inside or outside.

If the number of intersections is odd, the point is inside the polygon.

If the number of intersections is even, the point is outside the polygon.

Odd-Even Method: The Odd-Even Method is a specific implementation of the Inside-Outside Test. Instead of counting the number of intersections directly, it considers the parity (odd or even) of the crossings. Here's a simplified explanation: Start with an initial count of 0. For each edge of the polygon:

4.What is viewing transformation? What is difference between window and viewport?

Viewing Transformation:

Viewing Transformation, also known as View Transformation, is a crucial step in computer graphics that involves transforming 3D world coordinates into 2D screen coordinates for display. In simpler terms, it determines how a scene or object is viewed from a specific camera or observer's perspective. The viewing transformation encompasses operations such as translation, rotation, scaling, and perspective projection to convert the spatial information of 3D objects into a 2D representation. Difference between Window and Viewport:

Window: Definition: The window is a 2D region in world coordinates that represents the portion of the scene that needs to be displayed. It acts as a frame or boundary that defines what part of the overall scene should be visible on the screen.

Purpose: The window helps in specifying the area of interest and determines which part of the scene will be mapped onto the 2D display.

Viewport: Definition: The viewport is the actual rectangular area on the screen or display where the contents of the window are mapped. It is the region where the final image is displayed to the user.

Purpose: The viewport is responsible for scaling and mapping the window's contents onto the screen. It determines how the 2D representation of the scene is positioned and scaled within the actual display area.

5.Explain the types of 3-D projections.

In computer graphics, 3D projections are techniques used to represent three-dimensional objects and scenes in two dimensions for display on a screen or in print. There are several types of 3D projections, each with its own method of converting 3D spatial information into a 2D representation.

projections: Perspective Projection:

Description: Perspective projection simulates the way humans perceive the world, where objects appear smaller as they move farther away.
Types: One-Point Perspective: All lines converge to a single vanishing point on the horizon. Often used for scenes viewed head-on. Two-Point Perspective: Lines converge to two vanishing points on the horizon, suitable for scenes viewed at an angle.
Parallel (Orthographic) Projection: Description: In parallel projection, lines remain parallel, and there is no foreshortening. This results in a more straightforward representation of objects, but it lacks the depth cues provided by perspective projection.
Types: Isometric Projection: A specific type of orthographic projection where the object is tilted to show three faces equally.
Axonometric Projection: Description: Axonometric projections maintain equal scale along all three axes, allowing for a more realistic representation of 3D objects.
Types: Isometric Projection: A type of axonometric projection where all three axes are equally foreshortened.
Trimetric Projection: All three axes have different foreshortening.
Oblique Projection: Description: Oblique projection involves projecting the three-dimensional object onto a two-dimensional plane at an angle.

6.Explain in detail Random Scan Display and Raster Scan Display in computer grapics

Random Scan Display:

A Random Scan Display, also known as a vector display, is a type of display device used in computer graphics. In a random scan display, the electron beam directly traces the path of the displayed lines or curves. Unlike raster scan displays, random scan displays do not scan the entire screen in a fixed order; instead, they draw only the specified lines or curves as directed by the system.
Key Features of Random Scan Displays:
Point Plotting: Random scan displays are well-suited for drawing points, lines, and curves based on mathematical equations or user input.
Coordinate-Based Drawing: Graphics commands are used to specify the coordinates of endpoints, and the display hardware draws lines connecting these points.
Drawing Primitives: Primitives like circles, arcs, and other complex shapes are drawn using algorithms that calculate the points on the shape.
Interactive Graphics: Random scan displays are often used in interactive environments where the user can control the drawing process dynamically.

Raster Scan Display: A Raster Scan Display is a type of display device commonly used in computer graphics. In a raster scan display, the electron beam systematically scans the entire screen in a fixed order, from left to right and top to bottom, creating a grid of pixels.
Key Features of Raster Scan Displays:
Pixel-Based Drawing: Graphics primitives are drawn pixel by pixel, with the electron beam scanning each row of pixels sequentially.
Framebuffer: Raster scan displays use a framebuffer to store the pixel values of the entire screen.
Refresh Rate: The entire screen is refreshed multiple times per second, and the refresh rate is a critical factor in determining the display's quality and flicker perception.
Well-Suited for Raster Images: Raster scan displays are well-suited for displaying raster images, where each pixel has a specific color value.

7.Explain Types of Graphics Devices.

Graphics devices refer to hardware components that enable the creation and display of graphical images on a computer.
Graphics Cards (Video Cards or GPU): Functionality: Graphics cards, or Graphics Processing Units (GPUs), are specialized processors designed to handle graphics-related tasks. Use Cases: GPUs are crucial for gaming, video playback, 3D rendering, and other graphics-intensive applications
Monitors: Functionality: Monitors are display devices that present visual information generated by the computer's graphics card. They come in various types, such as CRT (Cathode Ray Tube), LCD (Liquid Crystal Display), LED (Light-Emitting Diode), and OLED (Organic Light-Emitting Diode).
Printers and Plotters:
Functionality: Printers and plotters produce hard copies of digital images and graphics on paper or other media. Use Cases: Printers are commonly used for graphic design, document printing, and photo reproduction.
Scanners: Functionality: Scanners capture physical documents, images, or graphics and convert them into digital format. Use Cases: Scanners are used for tasks such as document digitization, image editing, and incorporating physical drawings into digital designs.
Graphics Tablets: Functionality: Graphics tablets consist of a pressure-sensitive surface and a stylus (digital pen). They allow users to draw or write directly on the tablet, with the movements and pressure translated into digital data.
Digital Cameras: Functionality: Digital cameras capture still images or video in digital format. They use sensors to record light and color information and store it as digital data.
3D Printers: Functionality: 3D printers create three-dimensional objects layer by layer based on digital 3D models. They use materials such as plastic, metal, or resin.

8.Explain Inverse Transformation with example in computer graphics.

Inverse transformation in computer graphics refers to the process of reversing a previously applied transformation. When objects are transformed in a 2D or 3D space, such as through translation, rotation, scaling, or a combination of these, the inverse transformation allows you to undo those changes and revert the object to its original state.

Let's consider a simple example of a 2D transformation, such as translation, to illustrate the concept of inverse transformation.

Suppose you have a 2D point P(x, y) that you want to translate by a vector (dx, dy). The translation is represented by the following equations:

$x'=x+dx$, $y'=y+dy$

Here, (x', y') are the coordinates of the transformed point. Now, to find the inverse transformation to revert this translation, you can subtract the translation vector from the transformed coordinates:

$x=x'-dx$, $y=y'-dy$

These equations represent the inverse transformation for translation.

Let's go through an example:

Original point: P(2, 3), Translation vector: (dx, dy) = (4, -1) Apply the translation:

$x'=2+4=6$ $=3-1=2$

$y'=3-1=2$

Transformed point: P'(6, 2)

Apply the inverse transformation to revert the translation:

$x=6-4=2$ $=2+1=3$, $y=2+1=3$

You can see that applying the inverse transformation brings you back to the original point (2, 3), undoing the effect of the translation.

9.Explain Bitmap Method with example.

The term "Bitmap" typically refers to a method of representing images in computer graphics. A bitmap is a two-dimensional array of pixels, where each pixel corresponds to a specific location in the image and has its own color information.

Here's an explanation of the Bitmap method with an example:

Bitmap Method: Pixel Grid: A bitmap image is essentially a grid of pixels. Each pixel represents the smallest unit of the image and contains color information.

Color Representation: Colors are represented using a combination of color channels. Common color channels include Red (R), Green (G), and Blue (B) in RGB color space.

Resolution: The resolution of a bitmap image is determined by the number of pixels in the horizontal and vertical directions. Higher resolution images have more pixels and, therefore, can represent more detail.

File Size: Bitmap images can result in large file sizes, especially at higher resolutions, as each pixel stores color information independently. Compression techniques are often used to reduce file sizes.

Example: Let's consider a small example of a simple 3x3 pixel bitmap image in grayscale. Each pixel is represented by a numerical value indicating its intensity (brightness):

2 5 8
1 4 7
3 6 9

In this example, each number represents the intensity of a pixel.

(R:2, G:2, B:2) (R:5, G:5, B:5) (R:8, G:8, B:8)
(R:1, G:1, B:1) (R:4, G:4, B:4) (R:7, G:7, B:7)
(R:3, G:3, B:3) (R:6, G:6, B:6) (R:9, G:9, B:9)

10.What are different types of printer? Explain its working.

Printers are output devices that produce a hard copy of electronic or digital documents. There are several types of printers, each with its own technology and working principles. Here are some common types of printers:

Inkjet Printers: Working Principle: Inkjet printers create images by propelling droplets of ink onto paper. The print head contains tiny nozzles that spray the ink in a controlled manner, forming characters and images.

Laser Printers: Working Principle: Laser printers use a laser beam to create an electrostatic image on a drum. Toner (powdered ink) is then attracted to the charged areas on the drum and transferred to paper, where it is fused using heat.

Dot Matrix Printers: Working Principle: Dot matrix printers use a matrix of tiny pins to strike an inked ribbon, creating patterns of dots that form characters and images on paper.

Thermal Printers: Working Principle: Thermal printers use heat to produce an image on specially coated paper. There are two types: direct thermal (heat activates the paper's coating) and thermal transfer (heat transfers ink from a ribbon to the paper).

3D Printers: Working Principle: 3D printers build objects layer by layer from digital models. They use materials like plastic, metal, or resin, and the printing process depends on the specific technology (e.g., Fused Deposition Modeling, Stereolithography).

Dye-Sublimation Printers: Working Principle: Dye-sublimation printers use heat to transfer dye onto various materials, such as paper or fabric. The dye transitions from a solid to a gas without becoming a liquid.

Plotters: Working Principle: Plotters are used for large-format printing and use pens or markers to draw continuous lines on paper based on vector graphics.

11.Explain Plotters in computer graphics.

In computer graphics, a plotter is a specialized output device used to produce high-quality, large-format graphics such as engineering drawings, architectural blueprints, and detailed illustrations.

Vector Graphics: Plotters work with vector graphics, which define images using mathematical equations to represent lines and shapes.

Pen or Marker System:Plotters have a mechanism that holds one or more pens or markers. These pens can be of different colors and line thicknesses.

Coordinate System: Plotters use a coordinate system to the position of the pen on the drawing surface. The computer sends specific commands to the plotter, instructing it to move the pen to precise coordinates to create lines and shapes.

High Precision: Plotters are known for their high precision and accuracy. They can produce detailed and intricate drawings with consistent line quality.

Large Format: Plotters are designed to handle large sheets of paper or other drawing surfaces. This makes them ideal for creating large-scale drawings or blueprints.

Speed and Efficiency: Plotters can be relatively slow compared to some other printing technologies, especially when producing complex drawings.

Applications: Plotters are commonly used in various fields, including architecture, engineering, cartography, and graphic design. They are particularly well-suited for tasks that require detailed line drawings, such as architectural plans, circuit diagrams, and GIS maps.

Types of Plotters: There are different types of plotters, including drum plotters, flatbed plotters, and electrostatic plotters.

12. Explain translation, rotation, scaling.

Translation, rotation, and scaling are fundamental transformations used in computer graphics to manipulate the position, orientation, and size of objects in a two-dimensional (2D) or three-dimensional (3D) space. These transformations play a crucial role in tasks such as rendering, animation, and computer-aided design. Let's explore each transformation individually: 1. Translation: Definition: Translation involves moving an object from one position to another by a specified distance along a specified direction.

Mathematical Representation (2D):
$x'=x+\Delta x$ $y'=y+\Delta y$

Mathematical Representation (3D):
$x'=x+\Delta x$, $y'=y+\Delta y$, $z'=z+\Delta z$

Here,(x,y,z) are the original coordinates, (x',y',z') are the translated coordinates, and $(\Delta x,\Delta y,\Delta z)$ represent the translation distances along the respective axes.

2. Rotation: Definition: Rotation involves changing the orientation of an object by a specified angle around a fixed point, known as the center of rotation.

Mathematical Representation (2D):
$x'=x\cos(\theta)-y\sin(\theta)$,$y'=x\sin(\theta)+y\cos(\theta)$

Mathematical Representation (3D - Rotation around Z-axis):
$x'=x\cos(\theta)-y\sin(\theta)$, $y'=x\sin(\theta)+y\cos(\theta)$, $z'=z$.

3. Scaling: Definition: Scaling involves changing the size of an object by multiplying its coordinates by scaling factors along each axis.

Mathematical Representation (2D):
$x'=sx\cdot x$, $y'=sy\cdot y$

Mathematical Representation (3D):
$x'=sx\cdot x$, $y'=sy\cdot y$, $z'=sz\cdot z$

Here, (x,y,z) are the original coordinates,(x',y',z') are the scaled coordinates, and (sx,sy,sz) represent the scaling factors along the respective axes.

25.Explain 2D reflection in details.

In computer graphics, a 2D reflection is a transformation that flips or mirrors an object over a line. This line is often referred to as the "mirror line" or "reflection axis." The reflection process involves changing the positions of points in the object to create a mirrored version.

Mathematical Representation: Suppose you have a point $(P(x,y)$ in the original object and you want to reflect it over a line L. The reflected point $P'(x',y')$ can be computed using the following formulas:

1.For a Horizontal Line (y-axis reflection): $P'(x',y')=(x,-y)$

2.For a Vertical Line (x-axis reflection): $P'(x',y')=(-x,y)$

1.For an Oblique Line ($y = mx + b$): $P'(x',y')=(1+m2x-m(y-b),1+m2y-b-mx)$

Steps for 2D Reflection: 1.Define the Reflection Axis: Choose the line over which you want to reflect the object. This line can be vertical, horizontal, or oblique.

2.Compute Reflected Coordinates:•For each point in the original object, apply the reflection formulas based on the chosen axis to obtain the corresponding reflected coordinates.

13.Explain Midpoint methods for line.

Mid Point Line Drawing Algorithm-

Procedure- Given- Starting coordinates = (X_0, Y_0)

Ending coordinates = (X_n, Y_n)

The points generation using Mid Point Line Drawing Algorithm involves the following steps-

Step-01: Calculate ΔX and ΔY from the given input.

These parameters are calculated as-

$\Delta X = X_n - X_0$, $\Delta Y = Y_n - Y_0$

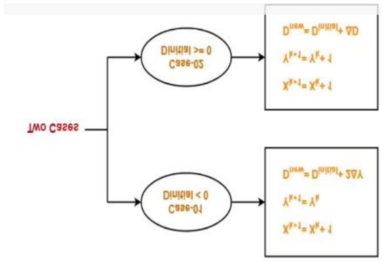
Step-02: Calculate the value of initial decision parameter and ΔD.

These parameters are calculated as-

$D_{initial} = 2\Delta Y – \Delta X, \Delta D = 2(\Delta Y – \Delta X)$

Step-03: The decision whether to increment X or Y coordinate depends upon the flowing values of $D_{initial}$.

Follow the below two cases-



Step-04: Keep repeating Step-03 until the end point is reached.

For each D_{new} value, follow the above cases to find the next coordinates.

(20, 10) and ending coordinates (30, 18)

Advantages of Mid Point Line Drawing Algorithm-

Accuracy of finding points is a key feature of this algorithm.

It is simple to implement.

It uses basic arithmetic operations.

It takes less time for computation.

The resulted line is smooth as compared to other line drawing algorithms.

Disadvantages of Mid Point Line Drawing Algorithm-

This algorithm may not be an ideal choice for complex graphics and images.

In terms of accuracy of finding points, improvement is still needed.

There is no any remarkable improvement made by this algorithm.

14.Explain the steps in Bresenham's circle drawing algorithm with example.

Bresenham Line Drawing Algorithm-

Procedure- Given-Starting coordinates = (X_0, Y_0)

Ending coordinates = (X_n, Y_n)

The points generation using Bresenham Line Drawing Algorithm involves the following steps-

Step-01: Calculate ΔX and ΔY from the given input.

These parameters are calculated as-

$\Delta X = X_n – X_0$, $\Delta Y = Y_n – Y_0$

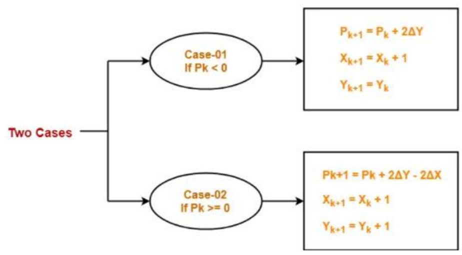
Step-02: Calculate the decision parameter P_k.

It is calculated as- $P_k = 2\Delta Y – \Delta X$

Step-03: Suppose the current point is (X_k, Y_k) and the next point is (X_{k+1}, Y_{k+1}).

Find the next point depending on the value of decision parameter P_k.

Follow the below two cases-



Step-04: Keep repeating Step-03 until the end point is reached or number of iterations equals to (ΔX-1) times.

Advantages of Bresenham Line Drawing Algorithm-

The advantages of Bresenham Line Drawing Algorithm are-

- It is easy to implement.
- It is fast and incremental.
- It executes fast but less faster than DDA Algorithm.
- The points generated by this algorithm are more accurate than DDA Algorithm.
- It uses fixed points only.

Disadvantages of Bresenham Line Drawing Algorithm-

The disadvantages of Bresenham Line Drawing Algorithm are-

Though it improves the accuracy of generated points but still the resulted line is not smooth.

This algorithm is for the basic line drawing.

It can not handle diminishing jaggies.

15.Explain DDA Line drawing algorithm with example.

DDA Algorithm- DDA Algorithm is the simplest line drawing algorithm.

procedure- Given-

Starting coordinates = (X_0, Y_0)

Ending coordinates = (X_n, Y_n)

The points generation using DDA Algorithm involves the following steps-

Step-01: Calculate ΔX, ΔY and M from the given input.

These parameters are calculated as-

$\Delta X = X_n - X_0$

$\Delta Y = Y_n - Y_0$

$M = \Delta Y / \Delta X$

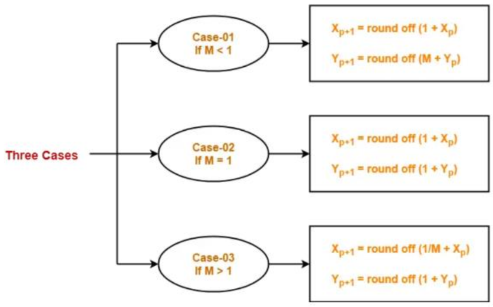
Step-02: Find the number of steps or points in between the starting and ending coordinates.

if (absolute (ΔX) > absolute (ΔY))
Steps = absolute (ΔX);

else
Steps = absolute (ΔY);

Step-03: Suppose the current point is (X_p, Y_p) and the next point is (X_{p+1}, Y_{p+1}).

Find the next point by following the below three cases-



Step-04: Keep repeating Step-03 until the end point is reached or the number of generated new points (including the starting and ending points) equals to the steps count.

16.Explain Midpoint methods for line and circle.

Mid Point Circle Drawing Algorithm-

The points for other octacts are generated using the eight symmetry property.

Procedure-

Given- Centre point of Circle = (X0, Y0)

Radius of Circle = R

The points generation using Mid Point Circle Drawing Algorithm involves the following steps-

Step-01: Assign the starting point coordinates (X0, Y0) as-

X0 = 0, Y0 = R

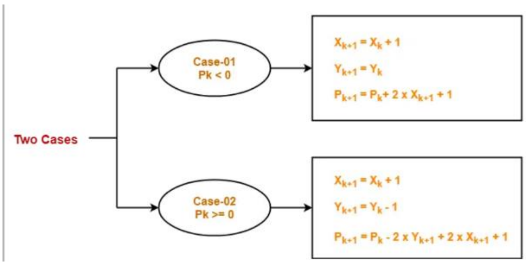
Step-02: Calculate the value of initial decision parameter P0 as-

P0 = 1 – R

Step-03: Suppose the current point is (Xk, Yk) and the next point is (Xk+1, Yk+1).

Find the next point of the first octant depending on the value of decision parameter Pk.

Follow the below two cases-



Step-04: If the given centre point (X_0, Y_0) is not (0, 0), then do the following and plot the point- $X_{plot} = X_c + X_0$, $Y_{plot} = Y_c + Y_0$

Here, (X_c, Y_c) denotes the current value of X and Y coordinates.

Step-05: Keep repeating Step-03 and Step-04 until X_{plot} >= Y_{plot}.

Step-06: Step-05 generates all the points for one octant.

17.Explain the steps in Bresenham's circle drawing algorithm with example.

Bresenham Circle Drawing Algorithm-

The points for other octacts are generated using the eight symmetry property.

Also Read- Mid Point Circle Drawing Algorithm

Procedure- Given- Centre point of Circle = (X0, Y0)

Radius of Circle = R

The points generation using Bresenham Circle Drawing Algorithm involves the following steps-

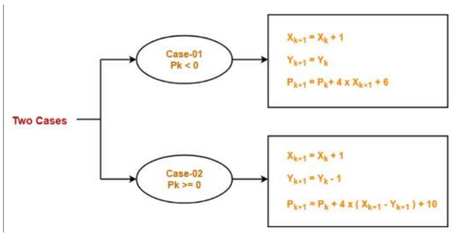
Step-01: Assign the starting point coordinates (X0, Y0) as- X0 = 0, Y0 = R

Step-02: Calculate the value of initial decision parameter P0 as- P0 = 3 – 2 x R

Step-03: Suppose the current point is (Xk, Yk) and the next point is (Xk+1, Yk+1).

Find the next point of the first octant depending on the value of decision parameter Pk.

Follow the below two cases-



Step-04: If the given centre point (X_0, Y_0) is not (0, 0), then do the following and plot the point-

$X_{plot} = X_c + X_0$, $Y_{plot} = Y_c + Y_0$

Here, (X_c, Y_c) denotes the current value of X and Y coordinates.

Step-05: Keep repeating Step-03 and Step-04 until X_{plot} => Y_{plot}. Step-06: Step-05 generates all the points for one octant.

PRACTICE PROBLEMS BASED ON BRESENHAM CIRCLE DRAWING ALGORITHM-

Problem-01: Given the centre point coordinates (0, 0) and radius as 8, generate all the points to form a circle.

Solution-Centre Coordinates of Circle (X_0, Y_0) = (0, 0)

Radius of Circle = 8

Step-01: Assign the starting point coordinates (X_0, Y_0) as-

$X_0 = 0$,$Y_0 = R = 8$

Step-02: Calculate the value of initial decision parameter P_0 as-

$P_0 = 3 – 2 \times R$

$P_0 = 3 – 2 \times 8$

$P_0 = -13$

Step-03: As $P_{initial} < 0$, so case-01 is satisfied.

Thus,$X_{k+1} = X_k + 1 = 0 + 1 = 1$

$Y_{k+1} = Y_k = 8$

$P_{k+1} = P_k + 4 \times X_{k+1} + 6 = -13 + (4 \times 1) + 6 = -3$

Step-04:

This step is not applicable here as the given centre point coordinates is (0, 0).

Step-05:

Step-03 is executed similarly until $X_{k+1} >= Y_{k+1}$ as follows-

18.Explain boundary fill algorithm.

Boundary Filled Algorithm: This algorithm uses the recursive method. First of all, a starting pixel called as the seed is considered. The algorithm checks boundary pixel or adjacent pixels are colored or not. If the adjacent pixel is already filled or colored then leave it, otherwise fill it.

```
void boundaryFill4(int x, int y, int fill_color,int boundary_color) {
   if(getpixel(x, y) != boundary_color &&
     getpixel(x, y) != fill_color) {
      putpixel(x, y, fill_color);
      boundaryFill4(x + 1, y, fill_color, boundary_color);
      boundaryFill4(x, y + 1, fill_color, boundary_color);
      boundaryFill4(x - 1, y, fill_color, boundary_color);
      boundaryFill4(x, y - 1, fill_color, boundary_color);   }}
```

19.Explain cohen-sutherland line clipping algorithm.

1) Assign the region codes to both endpoints. 2) Perform OR operation on both of these endpoints.
3) if OR = 0000, then it is completely visible (inside the window). else Perform AND operation on both these endpoints.
i) if AND ? 0000, then the line is invisible and not inside the window. Also, it can't be considered for clipping.
ii) else AND = 0000, the line is partially inside the window and considered for clipping.
4) After confirming that the line is partially inside the window, then we find the intersection with the boundary of the window. By using the following formula:-

Slope:- m= (y2-y1)/(x2-x1)
x = x + (y_wmax – y)/m
y = y_wmax
x = x + (y_wmin – y)/m
y = y_wmin
y = y+ (x_wmin – x)*m
x = x_wmin
y = y + (x_wmax -x)*m
x = x_wmax

5) overwrite the endpoints with a new one and update it. 6) Repeat the 4th step till your line doesn't get completely clippe.

20.Explain in detail polygon Edge fill algorithm

Step 1 − Find out the Ymin and Ymax from the given polygon.
Scan Line AlgorithmStep 2 − ScanLine intersects with each edge of the polygon from Ymin to Ymax. Name each intersection point of the polygon. As per the figure shown above, they are named as p0, p1, p2, p3.
Step 3 − Sort the intersection point in the increasing order of X coordinate i.e. p0,p1
, p1,p2, and p2,p3
Step 4 − Fill all those pair of coordinates that are inside polygons and ignore the alternate pairs.

21.Describe the solid area filling algorithm.

```
Procedure fill (x, y, color, color1: integer)
int c;
c=getpixel (x, y);
if (c!=color) (c!=color1)
{
   setpixel (x, y, color)
   fill (x+1, y, color, color 1);
   fill (x-1, y, color, color 1);
   fill (x, y+1, color, color 1);
   fill (x, y-1, color, color 1);
}
```

22.Explain clipping algorithm.

In the algorithm, first of all, it is detected whether line lies inside the screen or it is outside the screen. All lines come under any one of the following categories:
1. Visible: If a line lies within the window, i.e., both endpoints of the line lies within the window. A line is visible and will be displayed as it is.
2. Not Visible: If a line lies outside the window it will be invisible and rejected. Such lines will not display. If any one of the following inequalities is satisfied, then the line is considered invisible. Let A (x1,y2) and B (x2,y2) are endpoints of line xmin,xmax are coordinates of the window.
ymin,ymax are also coordinates of the window.

$x1>xmax$
$x2>xmax$
$y1>ymax$
$y2>ymax$
$x1<xmin$
$x2<xmin$
$y1<ymin$
$y2<ymin$

23.Explain projection in detail and it's types(parallel projection).

Projection in computer graphics refers to the process of transforming 3D coordinates into 2D coordinates.
<u>Parallel Projection:</u> Parallel projection is a type of projection in which all the lines of sight from the objects to the projection plane are parallel. This means that the projection lines are parallel in both directions, and they do not converge towards a single point.
<u>Orthographic Projection:</u> In orthographic projection, the projection lines are perpendicular to the projection plane. There are two main subtypes: multi-view orthographic projection and axonometric projection.In multi-view orthographic projection, multiple 2D views (top, front, side) of a 3D object are created, each with its own projection plane.
<u>Oblique Projection:</u> Oblique projection involves projecting the three-dimensional object onto a two-dimensional plane at an angle. There are different types of oblique projections, such as cavalier and cabinet projections.

24.What is vanishing points in computer graphics.

In computer graphics, a vanishing point refers to a point in the 2D image plane where parallel lines in a 3D scene appear to converge. The concept of vanishing points is closely related to perspective and is crucial for creating realistic images and scenes. When you look at a scene in the real world, parallel lines like railroad tracks or the edges of a building seem to meet at a point in the distance. This point is the vanishing point, and it gives the viewer a sense of depth and distance.
<u>Convergence of Lines:</u> Parallel lines that are parallel in the 3D world will converge towards a vanishing point in the 2D image. <u>Number of Vanishing Points:</u> The number of vanishing points depends on the number of sets of parallel lines in a scene. Three-point perspective involves three vanishing points.
<u>Perspective Projection:</u> Vanishing points are a result of perspective projection, which simulates how objects appear smaller as they move farther away from the viewer.<u>Realism:</u> The use of vanishing points in computer graphics helps create realistic scenes by mimicking the way our eyes perceive depth in the real world.
<u>Artistic and Technical Applications:</u> Vanishing points are widely used in various fields, including computer graphics, architectural design, art, and video game development, to create visually compelling and accurate representations of three-dimensional spaces.