Unit-I: Introduction to Computer Graphics
Short Answer Questions:

Q. What is aspect ratio? (Nov 22)
Ans. Aspect Ratio: Aspect ratio is the ratio of a display screen's width to its height, typically expressed as W:H (e.g., 16:9). It determines how visuals are proportioned on a screen.

Q. What do you mean by Raster scan display? (Nov 22)
Ans. Raster scan displays create images by scanning the screen line-by-line from top to bottom. It is used in CRTs and modern display monitors.

Q. What is the concept of refreshing in the CRTs? (Nov 22)
Ans. Refreshing refers to redrawing the screen repeatedly to keep the image visible. CRTs require frequent refreshing to avoid flicker.

Q. What does the acronym pixel stands for? What are the characteristics of a pixel? (Nov 22)
Ans. Pixel stands for "picture element" and is the smallest addressable unit on the screen. Its main characteristics are resolution, brightness, and color depth.

Q. What are common input devices used for interacting with computer graphics applications? (Nov 23)
Ans. Common input devices include mouse, keyboard, touchscreen, graphic tablets, and light pens. These allow precise control in graphic creation.

Q. What are the key differences between color CRT and monochrome CRT displays? (Nov 23)
Ans. Color CRTs display images in multiple colors using RGB phosphors, while monochrome CRTs display in single color (e.g., green or white).

Q. What types of input devices are commonly used in modern computing systems? (Nov 24)
Ans. Devices like stylus pens, game controllers, 3D scanners, and motion sensors are used in modern systems for better interaction and precision.

Q. What are some common applications of virtual reality environments? (Nov 24)
Ans. VR environments are used in gaming, simulation training, education, architecture, and healthcare for immersive experiences.

Q. How do flat panel displays achieve higher resolutions and refresh rates? (Nov 24)
Ans. They use technologies like LCD and OLED with finer pixel density and faster response time to achieve higher resolutions and refresh rates.

Q. Why have CRT displays become less common in recent years? (Nov 24)
Ans. CRTs are bulky, energy-intensive, and heavy, which led to their replacement by slim, efficient flat panel displays like LCDs and LEDs.

Long Answer Questions:

Q.
a) Distinguish between random and raster scan systems. (Nov 22)
b) What are the applications of Computer Graphics? Explain.
Ans. (a) Distinguish between Random and Raster Scan Systems: Random scan and raster scan are two types of display systems used in computer graphics.
Random scan system, also called a vector scan or calligraphic display, draws images by directing the electron beam only to the parts of the screen where drawing is required. It uses a sequence of straight-line segments and is ideal for rendering line drawings and wireframe models. This system

offers high resolution and flicker-free display for line art but is not suitable for complex shaded or colored images.

Raster scan system, on the other hand, draws the entire screen line by line from top to bottom. It stores image data in a frame buffer and refreshes the screen at regular intervals. This is the standard system used in most monitors and supports complex graphics with colors, shading, and filled areas. It is more versatile and suited for multimedia applications.

In summary, random scan is best for vector graphics and technical drawings, whereas raster scan is used for realistic images, videos, and gaming due to its pixel-based rendering capabilities.

(b) What are the Applications of Computer Graphics?: Computer graphics play a vital role across multiple domains due to their ability to visually represent data and simulate environments.

In entertainment, it is used extensively in movies, animations, video games, and virtual reality to create lifelike characters and immersive scenes. Education and training utilize graphics for interactive simulations such as flight simulators or medical procedure models, enhancing learning through visualization.

In engineering and design, CAD (Computer-Aided Design) uses graphics to create and analyze designs in automotive, aerospace, and architecture. Scientific visualization helps in interpreting large datasets from physics, biology, or meteorology by turning abstract numbers into understandable visuals, like weather maps or 3D molecular models.

In business and information systems, graphs, dashboards, and infographics help in decision-making by representing complex data clearly. Medical imaging uses computer graphics to render CT scans, MRIs, and 3D visualizations of human anatomy.

User interfaces also benefit from computer graphics, making software more interactive and user-friendly. Overall, computer graphics enhance communication, creativity, and comprehension, making them indispensable in today's digital world.

Q. What are the differences between RGTB and CMY models? Explain in detail. (Nov 22)

Ans. The RGB (Red, Green, Blue) and CMY (Cyan, Magenta, Yellow) color models are two fundamental systems used in digital imaging and printing, respectively, but they serve different purposes and operate on opposite principles.

RGB is an additive color model used primarily in digital displays like computer monitors, televisions, and cameras. It combines red, green, and blue light in varying intensities to produce a wide spectrum of colors. When all three colors are combined at full intensity, the result is white; when none are added, the result is black. RGB is ideal for devices that emit light.

CMY, on the other hand, is a subtractive color model used in color printing. It uses cyan, magenta, and yellow inks or pigments to absorb (subtract) light. When all three colors are combined, they ideally absorb all light, producing black. In practice, black ink (K) is added, forming the CMYK model, to improve depth and save ink. CMY is used for materials that reflect light rather than emit it.

Key differences:

- RGB is additive (light-based); CMY is subtractive (ink-based).
- RGB is used for screens; CMY is used for printing.
- Combining RGB colors gives white; combining CMY colors gives black.

These differences define their respective applications in visual media and print production.

Q. Discuss the technology behind flat panel displays, including LCD and OLED. Describe the advantages of flat panel displays over CRT monitors in terms of size, power efficiency and image quality. (Nov 23)

Ans. Flat panel displays have revolutionized modern visual technology, primarily through Liquid Crystal Display (LCD) and Organic Light Emitting Diode (OLED) technologies.

LCDs use liquid crystals that do not emit light directly. Instead, a backlight (typically LED) passes through polarized panels and liquid crystal layers that manipulate light to form images. The crystals align in response to electric signals, allowing varying light passage and color filtering to display content.

OLEDs, in contrast, consist of organic compounds that emit light when electricity is applied. Each pixel in an OLED panel acts as its own light source, enabling deep blacks and vibrant colors without the need for a backlight. This self-emissive property also allows for thinner and more flexible displays.
Advantages over CRTs:

1. Size & Weight: Flat panels are significantly thinner and lighter than bulky Cathode Ray Tube (CRT) monitors, allowing for sleeker devices and wall-mounted screens.
2. Power Efficiency: LCDs consume less power than CRTs as they don't need electron beams or bulky vacuum tubes. OLEDs, especially on dark images, are even more energy-efficient.
3. Image Quality: Flat panels offer higher resolutions, better brightness, and superior contrast ratios. OLEDs, particularly, excel in color accuracy, wide viewing angles, and ultra-fast response times.

These benefits have made flat panel displays the industry standard across TVs, monitors, smartphones, and laptops.

Q. Write about working of common input devices. Explain how these devices are used in graphic design, gaming, and 3D modeling applications. (Nov 23)
Ans. Common input devices include the keyboard, mouse, graphic tablet, joystick, touchscreen, and scanner. These devices convert user actions into digital signals, allowing interaction with computers.

- Keyboard: It sends keypress signals to the CPU for text and command input.
- Mouse: Detects motion and button clicks to control cursor movement. It uses optical or laser sensors for accurate tracking.
- Graphic Tablet: A pen-based device that detects pressure and angle to input hand-drawn graphics, ideal for digital artists.
- Joystick/Gamepad: Offers analog and digital input for controlling movements, mostly in games and simulations.
- Touchscreen: Combines display and input using capacitive or resistive sensing to detect finger or stylus touches.
- Scanner: Converts physical images or documents into digital format for editing or storage.

In graphic design, the graphic tablet allows precision drawing, the mouse enables navigation and tool selection, and the scanner digitizes artwork. In gaming, joysticks and gamepads offer immersive control, while keyboards and mice are used for strategy and FPS games. For 3D modeling, mice with high DPI, graphic tablets, and 3D mice allow smooth object manipulation and detailed control, enhancing productivity and precision in CAD and animation applications.

Q. What are the advantages and disadvantages of LCD flat panel displays compared to traditional CRT displays? What is the difference between working of an LED and OLED flat panel display? (Nov 24)
Ans. LCD (Liquid Crystal Display) panels offer several advantages over traditional CRT (Cathode Ray Tube) displays. LCDs are slimmer, lighter, and consume significantly less power. They produce sharper images with no geometric distortion and are ideal for flat-screen monitors and TVs. LCDs also emit less radiation and generate less heat, making them safer and more efficient.
However, LCDs have limited viewing angles, slower response times, and may suffer from color accuracy issues compared to CRTs. CRTs, though bulky and power-hungry, offer better color reproduction, contrast ratios, and faster refresh rates, which were once preferred for professional graphics work and gaming.
The difference between LED and OLED flat panel displays lies in their backlighting and pixel control. LED (Light Emitting Diode) displays are essentially LCDs that use LEDs as a backlight behind the screen. The backlight is always on, and liquid crystals control light passage.
OLED (Organic Light Emitting Diode) displays, in contrast, have self-emissive pixels that generate their own light. This enables true blacks, higher contrast, better color accuracy, and faster refresh rates. OLEDs are more flexible and thinner but are also more expensive and prone to screen burn-in compared to LED panels.

Unit-II: Scan-conversions
Short Answer Questions:

Q. What do you mean by scan conversion? (Nov 22)
Ans. Scan conversion is the process of converting geometric data into pixel-based data for display on raster screens. It maps mathematical shapes to discrete screen points.

Q. Why is scan conversion essential in computer graphics and image rendering? (Nov 23)
Ans. Scan conversion is essential to accurately render vector shapes on raster devices, ensuring visual consistency and precision.

Q. What are some common applications where Bresenham's line drawing algorithm is used? (Nov 23)
Ans. Bresenham's algorithm is used in graphic editors, CAD software, and embedded systems for fast and accurate line drawing.

Q. What is the primary purpose of flood fill and boundary fill techniques in computer graphics? (Nov 23)
Ans. These techniques are used to color enclosed areas in graphics, such as filling shapes or backgrounds efficiently.

Q. How do algorithms for line drawing differ between raster and vector graphics? (Nov 24)
Ans. Raster line drawing uses discrete pixels and algorithms, while vector graphics rely on mathematical equations for smooth, scalable lines.

Q. How does the choice of algorithm affect the quality and efficiency of circle drawing? (Nov 24)
Ans. Efficient algorithms like Midpoint or Bresenham's reduce computation and ensure symmetry, affecting both speed and visual quality.

Q. How do algorithms for ellipse drawing differ from those for circle drawing? (Nov 24)
Ans. Ellipse algorithms handle two radii and require more complex calculations than circle algorithms, which have a single radius and simpler symmetry.

Long Answer Questions:

Q. Write Bresenham's circle drawing algorithm. (Nov 22)
Ans. Bresenham's circle drawing algorithm is an efficient method for drawing a circle using only integer arithmetic. It avoids floating-point calculations, making it suitable for raster-based systems. The algorithm exploits the symmetry of a circle, drawing all eight octants simultaneously from the first octant calculations.
The circle is defined by its center (x<sub>c</sub>, y<sub>c</sub>) and radius **r**. Starting from the topmost point (0, r), the algorithm calculates the next pixel position using a decision parameter **p** to choose between two potential pixel locations.
Steps:

1. Initialize:
    - $x = 0, y = r$
    - $p = 3 - 2r$
2. Repeat until $x <= y$:
    - Plot eight symmetric points:

(x + xc, y + yc), (-x + xc, y + yc),
(x + xc, -y + yc), (-x + xc, -y + yc),
(y + xc, x + yc), (-y + xc, x + yc),
(y + xc, -x + yc), (-y + xc, -x + yc)

- If $p < 0$:
    - $p = p + 4x + 6$

   - Else:
      - p = p + 4(x - y) + 10
      - y = y - 1
   - Increment x by 1

This method is widely used in computer graphics due to its speed and accuracy on pixel grids.

Q. Circle and ellipse drawing algorithms are fundamental in graphics rendering. Compare and contrast the Bresenham and midpoint circle drawing algorithms, highlighting their principles, efficiency and applications. (Nov 23)

Ans. Bresenham's and Midpoint circle drawing algorithms are both efficient rasterization techniques for drawing circles using only integer calculations, making them ideal for real-time computer graphics.

Bresenham's Circle Algorithm uses a decision parameter to determine the next pixel, updating it with simple integer additions and subtractions. It draws all eight octants of the circle simultaneously using symmetry, starting from the topmost point (0, r). The algorithm is extremely fast and optimized for hardware implementations due to its avoidance of floating-point arithmetic.

Midpoint Circle Algorithm, on the other hand, is based on the midpoint criteria: it checks whether the midpoint between two possible pixels lies inside or outside the circle boundary. Depending on this, it chooses the appropriate pixel and updates the decision parameter accordingly. Like Bresenham's, it exploits circle symmetry but is slightly more general and easier to adapt to other shapes (like ellipses).

Key Differences:

- Precision: Both are equally accurate for raster devices.
- Efficiency: Bresenham's is marginally faster and simpler to implement due to fewer calculations.
- Applications: Both are used in CAD tools, game engines, and embedded systems where drawing efficiency and precision are essential.

Ultimately, both algorithms are foundational, and choice depends on specific implementation requirements.

Q. What are some common algorithms for filling of closed region in raster graphics? Discuss their working and how do these algorithms differ in terms of efficiency and accuracy? (Nov 24)

Ans. In raster graphics, region filling is essential to color enclosed areas. Three widely used algorithms for this purpose are:

1. Flood Fill Algorithm:
   - Working: Starts from a seed point inside the region and spreads outward by checking neighboring pixels (4-connected or 8-connected). If a pixel is part of the region and not already filled, it's filled and its neighbors are added to the queue/stack.
   - Types: Recursive and Iterative.
   - Use case: Paint bucket tool in graphics editors.
2. Boundary Fill Algorithm:
   - Working: Begins at a seed point and fills outward until it hits a boundary color. It is suitable for filling regions with a clearly defined border.
   - Variants: 4-connected and 8-connected.
   - Drawback: Can lead to stack overflow for large regions (in recursive versions).
3. Scanline Fill Algorithm:
   - Working: Fills pixel rows (scanlines) between the intersection points of the region boundary and horizontal lines. This is a more efficient, line-by-line method.
   - Best for: Complex polygons and large areas.

Efficiency and Accuracy:

- Flood/Boundary Fill are simple but slower and memory-intensive for large regions.
- Scanline Fill is more efficient and accurate for polygonal regions with fewer recursive calls and faster filling.

Each algorithm suits different scenarios based on shape complexity, size, and performance needs.

Q. What are some common circle drawing algorithms and how do they work? Explain the concept of Mid-point. (Nov 24)

Ans. In computer graphics, circle drawing algorithms are used to render circles on pixel-based displays using efficient and accurate methods. The most common algorithms include:

1. Midpoint Circle Drawing Algorithm:
    - This algorithm is based on the idea of using the midpoint between two potential pixels to determine the next pixel to draw.
    - Starting from the top of the circle (0, r), it uses eight-way symmetry to calculate points in all octants of the circle.
    - It uses a decision parameter p initialized to 1 - r, which helps decide whether the next point will be directly east or southeast of the current point.
    - The algorithm updates p based on the midpoint's position relative to the circle's perimeter.
    - It is highly efficient because it uses only integer addition and subtraction, avoiding floating-point operations.
2. Bresenham's Circle Drawing Algorithm:
    - A variation of the midpoint method that also relies on incremental integer calculations.
    - It optimizes performance by minimizing calculations and is commonly used in embedded systems and real-time applications.

Concept of Midpoint:: The midpoint is the pixel between two candidates (e.g., (x+1, y) and (x+1, y)). If the midpoint lies inside the circle, the pixel closer to the center is chosen. This technique improves performance and accuracy in rasterization of circles.

Unit-III: 2-Dimensional Graphics
Short Answer Questions:

Q. What is viewing transformation? (Nov 22)
Ans. Viewing transformation maps a scene from world coordinates to screen coordinates. It controls how objects are viewed and displayed on the screen.

Q. What are homogeneous coordinates? (Nov 22)
Ans. Homogeneous coordinates use an extra dimension (w) to simplify matrix transformations in graphics. They help represent translations as matrix multiplications.

Q. What is the difference between window and viewport in the context of clipping? (Nov 23)
Ans. A window defines the part of the scene to display, while a viewport specifies where on the screen it appears. It's a crucial part of the clipping and scaling process.

Q. What are the common types of 2D transformations used in computer graphics? (Nov 24)
Ans. Common 2D transformations include translation, rotation, scaling, and shearing. These operations manipulate the position and shape of objects in 2D space.

Long Answer Questions:

Q. Magnify the triangle P(0,0), Q(2,2) and R(0,4) to four times its size while keeping R(10,4) fixed. (Nov 22)
Ans. To**magnify a triangle** by a scaling factor while keeping a particular point fixed, we use the concept of **scaling with a fixed point**. In this case, we are asked to **magnify triangle P(0,0), Q(2,2), R(0,4)** by **four times** its size while keeping **R(10,4)** fixed (note: there's a discrepancy—original R is (0,4), but fixed point is (10,4), so we assume the point (10,4) is the fixed reference and not a vertex).
General Formula for Scaling with a Fixed Point (xf, yf):
For any point (x, y) scaled by factor s with fixed point (xf, yf):

x' = xf + s * (x - xf)
y' = yf + s * (y - yf)
Given:

- Scaling factor s = 4
- Fixed point = (10, 4)
- Points to scale: P(0,0), Q(2,2), R(0,4)

Apply the formula:
P(0,0):

x' = 10 + 4*(0 - 10) = 10 - 40 = -30
y' = 4 + 4*(0 - 4) = 4 - 16 = 2
Q(2,2):

x' = 10 + 4*(2 - 10) = 10 - 32 = -22
y' = 4 + 4*(2 - 4) = 4 - 8 = -4
R(0,4):

x' = 10 + 4*(0 - 10) = -30
y' = 4 + 4*(4 - 4) = 4
Final Transformed Triangle:

- P' = (-30, 2)
- Q' = (-22, -4)
- R' = (-30, 4)

Thus, the triangle is magnified 4 times in size relative to the fixed point (10,4).

Q. Describe the primary 2D transformation operations, including translation, rotation, scaling, shearing and reflection.Explain how these transformations are applied to graphical objects and discuss their applications in tasks like graphics modeling and animation.(Nov 23),(Nov 23)

Ans. To **magnify a triangle** by a scaling factor while keeping a particular point fixed, we use the concept of **scaling with a fixed point**. In this case, we are asked to **magnify triangle P(0,0), Q(2,2), R(0,4)** by **four times** its size while keeping **R(10,4)** fixed (note: there's a discrepancy—original R is (0,4), but fixed point is (10,4), so we assume the point (10,4) is the fixed reference and not a vertex).

General Formula for Scaling with a Fixed Point (xf, yf):

For any point (x, y) scaled by factor s with fixed point (xf, yf):

x' = xf + s * (x - xf)
y' = yf + s * (y - yf)

Given:

- Scaling factor s = 4
- Fixed point = (10, 4)
- Points to scale: P(0,0), Q(2,2), R(0,4)

Apply the formula:

P(0,0):

x' = 10 + 4*(0 - 10) = 10 - 40 = -30
y' = 4 + 4*(0 - 4) = 4 - 16 = 2

Q(2,2):

x' = 10 + 4*(2 - 10) = 10 - 32 = -22
y' = 4 + 4*(2 - 4) = 4 - 8 = -4

R(0,4):

x' = 10 + 4*(0 - 10) = -30
y' = 4 + 4*(4 - 4) = 4

Final Transformed Triangle:

- P' = (-30, 2)
- Q' = (-22, -4)
- R' = (-30, 4)

Thus, the triangle is magnified 4 times in size relative to the fixed point (10,4).

Q. What is the difference between homogeneous coordinates and Cartesian coordinates in 2D transformations?What is the result of applying the translation matrix [1, 0, 2; 0, 1, 3; 0, 0,1] to the point (4, 5) after rotating this point on 60° around origin? (Nov 24)

Ans. In **2D transformations**, **Cartesian coordinates** represent points as (x, y), while **homogeneous coordinates** extend this to (x, y, 1). The addition of a third coordinate allows multiple geometric transformations (like **translation, rotation, scaling**) to be represented as **matrix multiplications**, which is not possible using Cartesian coordinates alone.

Homogeneous coordinates enable the use of **3×3 matrices** to unify transformations, allowing easy composition (chaining) of multiple transformations in a single matrix operation.

Applying the Transformation:

Given:

- Original point: **(4, 5)**
- Rotation: **60° about origin**
- Translation matrix:

$$T = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

Step 1: Convert to homogeneous coordinates:

$$P = \begin{bmatrix} 4 \\ 5 \\ 1 \end{bmatrix}$$

Step 2: Rotate the point by 60°

Rotation matrix (counterclockwise):

$$R = \begin{bmatrix} \cos 60^\circ & -\sin 60^\circ & 0 \\ \sin 60^\circ & \cos 60^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.5 & -0.866 & 0 \\ 0.866 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplying R × P:

$$R \cdot P = \begin{bmatrix} 0.5 * 4 + (-0.866) * 5 \\ 0.866 * 4 + 0.5 * 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 - 4.33 \\ 3.464 + 2.5 \\ 1 \end{bmatrix} = \begin{bmatrix} -2.33 \\ 5.964 \\ 1 \end{bmatrix}$$

Step 3: Apply translation:

$$T \cdot (RotatedP) = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -2.33 \\ 5.964 \\ 1 \end{bmatrix} = \begin{bmatrix} -2.33 + 2 \\ 5.964 + 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.33 \\ 8.964 \\ 1 \end{bmatrix}$$

Final Answer:

After rotating the point (4, 5) by 60° around the origin and translating it by (2, 3), the final point becomes approximately (-0.33, 8.96) in Cartesian coordinates.

Q. Explain the Cohen-Sutherland line clipping algorithm. Discuss the steps involved in clipping line segmentsand how the algorithm handles various cases, such as lines completely inside or outside the viewing area. Illustrate with suitable example and diagram. (Nov 23)

Ans. The Cohen-Sutherland Line Clipping Algorithm is a classic computer graphics algorithm used to clip a line segment against a rectangular clipping window. It divides the 2D space into 9 regions using region codes (outcodes) assigned to the endpoints of the line. Each outcode is a 4-bit binary code representing the position relative to the clipping window: Top, Bottom, Right, Left.

Steps:

1. Assign Outcodes: Each endpoint of the line is given a 4-bit outcode.
2. Trivial Acceptance: If both outcodes are 0000, the line is entirely inside the clipping area.
3. Trivial Rejection: If the bitwise AND of the outcodes ≠ 0, the line is entirely outside the window in a common region.
4. Clipping Required: Otherwise, the line intersects the clipping window. Find the intersection point with the clipping boundary using the slope-intercept form and update the endpoint with the new clipped point.
5. Repeat the steps until the line is either accepted or rejected.

Example:

For a clipping window (xmin=10, xmax=30, ymin=10, ymax=20) and line segment from (5,5) to (35,25):

- The line spans both outside and inside.
- Clipping calculates intersection points with the rectangle edges.
- Result: The line is clipped to fit inside (10,10) to (30,20).

This algorithm is **efficient**, especially for cases where many lines are trivially accepted or rejected, and works well in **interactive graphics** systems.

Q. How do clipping algorithms differ between vector and raster graphics? Given the line segment with end points (3, 4) and (8, 10) and the clipping rectangle with corners (4, 5) and (10, 8), apply the Cohen-Sutherland algorithm to clip the line segment. (Nov 24)

Ans. **Clipping algorithms differ** between vector and raster graphics in how they handle graphical data:

- **Vector graphics** use geometric shapes like lines and curves. Clipping involves computing new coordinates for objects (e.g., using algorithms like Cohen-Sutherland or Liang-Barsky) to redraw only the visible portion.
- **Raster graphics** work with pixels. Clipping determines which pixels fall within the clipping region and masks or discards the others—commonly used in image processing or scanline rendering.

Cohen-Sutherland Clipping Example

Given line segment: P1(3,4), P2(8,10)
Clipping rectangle: (xmin=4, ymin=5), (xmax=10, ymax=8)

Step 1: Assign region codes:

- For P1(3,4):
  - Left of xmin → 1
  - Below ymin → 1
  - Code = **0101**
- For P2(8,10):
  - Above ymax → 1
  - Code = **1000**

Step 2: Bitwise AND = 0101 & 1000 = 0000 → not trivially rejected → proceed to clip.

Step 3: Clip P1(3,4)

- First clip against bottom (y=5):
  - Slope (m) = (10−4)/(8−3) = 6/5
  - At y=5: x = x1 + (1/m)(y−y1) = 3 + (5/6)(5−4) ≈ 3.83 → new P1 ≈ (3.83, 5)
- Then clip P2(8,10) against top (y=8):
  - At y=8: x = 8 − (1/m)(10−8) = 8 − (5/6)(2) ≈ 6.33 → new P2 ≈ (6.33, 8)

Result: Clipped line from (3.83, 5) to (6.33, 8)

This shows how **vector clipping** uses precise math to redraw the visible line.

Q. What is clipping? Write an algorithm for it. (Nov 22)

Ans. **Clipping** is the process of confining drawing or rendering to a specific region of interest or viewport. In computer graphics, it helps in eliminating objects or parts of objects that are outside the viewing area to improve performance and clarity. Clipping is commonly applied to lines, polygons, and text in both 2D and 3D graphics.

There are various clipping algorithms, with **Cohen-Sutherland** and **Liang-Barsky** being commonly used for line clipping.

Cohen-Sutherland Line Clipping Algorithm

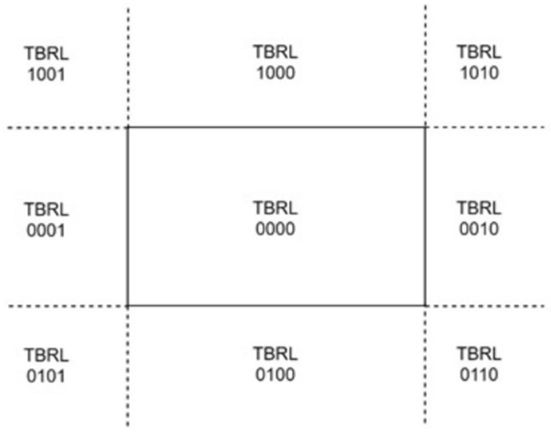
The Cohen-Sutherland Line Clipping Algorithm is used to determine which parts of a line segment lie inside a defined rectangular clipping window and which parts lie outside. The algorithm divides the 2D space into 9 regions. The middle region represents the area inside the clipping window, and the surrounding 8 regions represent the space outside.

By using a simple **4-bit region code** (TBRL), the algorithm can efficiently determine whether a line lies entirely inside, entirely outside, or partially inside the clipping window.

How the 4-Bit Region Code Works?

The 4-bit region code is used to identify the position of a point relative to the clipping window. Each bit in the code represents a direction (Top, Bottom, Right, Left). Here is how the code works:

- T (Top) − 1 if the point is above the window, 0 otherwise.
- B (Bottom) − 1 if the point is below the window, 0 otherwise.
- R (Right) − 1 if the point is to the right of the window, 0 otherwise.
- L (Left) − 1 if the point is to the left of the window, 0 otherwise.



Step 1: Assign Region Codes
First, we calculate the region codes for P1 and P2.
For P1(70, 20) −

- $xw_{min} = 50$, $xw_{max} = 80$
- $yw_{min} = 10$, $yw_{max} = 40$

Since 50 ≤ 70 ≤ 80 and 10 ≤ 20 ≤ 40, the point P1 lies inside the window, so its region code is 0000.
For P2(100, 10): x = 100, which is greater than $xw_{max} = 80$. The y-coordinate is on the boundary $yw_{min} = 10$.
Therefore, the region code for P2 is 0010 (indicating that it is to the right of the window).
Step 2: Check for Trivial Cases
Since the region code of P1 is 0000 and the region code of P2 is 0010, the line is partially inside the window. We now need to find the intersection point where the line crosses the window boundary.
Step 3: Calculate the Intersection Point
The slope of the line P1P2 is calculated as −

- Slope (m) = (y2 - y1) / (x2 - x1)
- Slope (m) = (10 - 20) / (100 - 70) = -10 / 30 = -1/3

Next, we find the intersection point on the right boundary (x = 80). The equation of the line is −

- y - y1 = m(x - x1)
- y - 20 = (-1/3)(80 - 70)
- y - 20 = (-1/3) * 10
- y = 20 - 10/3 = 20 - 3.33 = 16.67

Thus, the intersection point is M(80, 16.67).
Step 4: Draw the Clipped Line
The visible portion of the line is between P1(70, 20) and the intersection point M(80, 16.67). The portion of the line beyond M is outside the window and is discarded.
So in this example, we clipped the line P1(70, 20) and P2(100, 10) against a window with boundaries (50, 10) and (80, 40). We calculated the intersection point on the right boundary and determined the visible portion of the line that lies within the clipping window.

Unit-IV: 3-Dimensional Graphics
Short Answer Questions:

Q. What is the centre of projection in perspective projection? (Nov 22)
Ans. It is the point where all projection lines converge in perspective projection, simulating human eye perception. It gives depth and realism to 3D images.

Q. How the 3D images are represented on 2D plane in computer graphics? (Nov 22)
Ans. 3D objects are projected onto a 2D plane using transformations like perspective or orthographic projection. This allows 3D visualization on flat screens.

Q. What are 3D viewing devices and what applications benefit from their use? (Nov 23)
Ans. Devices like VR headsets and 3D glasses provide immersive experiences. They benefit applications such as gaming, medical imaging, and virtual simulations.

Q. What is the primary distinction between parallel and perspective projections in 3D rendering? (Nov 23)
Ans. Parallel projection maintains size and angles, used in engineering drawings, while perspective projection mimics real-world depth and size reduction with distance.

Q. What are the challenges and limitations associated with performing 3D geometric transformations and shading in real-time graphics? (Nov 23)
Ans. Real-time 3D rendering requires high computation for accurate transformations and realistic lighting. Limitations include hardware performance and memory constraints.

Q. What is perspective projection and how does it differ from parallel projection in computer graphics? (Nov 24)
Ans. Perspective projection converges lines at a point, creating depth; parallel projection keeps lines parallel, used for technical accuracy.

Q. What are some common applications of parallel projection in computer graphics? (Nov 24)
Ans. Commonly used in CAD, architectural design, and engineering drawings where accurate dimensions are crucial.

Q. What will be the change in the 3D rotation matrix if the rotation is clock-wise? (Nov 22)
Ans. In clockwise rotation, the angle in the rotation matrix becomes negative, altering the direction of the sine terms in transformation.

Long Answer Questions:

Q. Explain the terms: Projection plane, view plane, coordinate and view volume with regards to 3D graphics.State and explain the anomalies of perspective projection.(Nov 22)
Ans. In3D computer graphics, several fundamental terms are used to describe the process of projecting 3D objects onto a 2D display surface.

- Projection Plane: This is the surface onto which the 3D scene is projected to create a 2D image. In perspective projection, it is typically placed in front of the viewer and perpendicular to the line of sight.
- View Plane: Often used interchangeably with the projection plane, the view plane is the area on which the final image is displayed after projection. It is aligned with the viewer's screen and defines the viewing window.
- Coordinate System: 3D graphics use multiple coordinate systems—world coordinates (object placement), view coordinates (camera view), and screen coordinates (final output). Transformations are applied to move objects between these spaces.

- **View Volume**: The portion of 3D space that is visible from the camera's perspective. For perspective projection, it is a frustum (truncated pyramid); for orthographic projection, it is a rectangular box.

Anomalies of Perspective Projection:

1. **Perspective Distortion**: Objects far from the viewer appear much smaller, which can sometimes lead to unnatural representations.
2. **Vanishing Points**: Parallel lines converge at vanishing points, which may confuse depth perception if not handled properly.
3. **Foreshortening**: Parts of objects closer to the camera appear disproportionately large compared to farther parts.

These anomalies are natural in real vision but need careful handling in graphics to avoid unrealistic rendering.

Q. Compare and contrast parallel and perspective projections, highlighting their properties, use cases and the effects they produce in 3D scenes.Discuss when each projection type is most suitable. (Nov 23)

Ans. **Parallel and Perspective projections** are fundamental techniques in 3D computer graphics for displaying three-dimensional objects on a two-dimensional screen.

Parallel Projection:

In parallel projection, all projection lines are parallel and perpendicular to the projection plane. It preserves the true shape and size of objects regardless of their depth or distance from the viewer. This method does **not simulate depth** or foreshortening, making it ideal for **technical and engineering drawings** where accurate measurements and proportions are essential. Types include **orthographic** and **oblique projections**.

Perspective Projection:

Perspective projection mimics how the human eye perceives the world. Projection lines converge at a **single point** called the **vanishing point**, and objects appear smaller as they move farther from the viewer. This projection introduces **depth and realism**, making it best suited for **visual simulations, video games, architectural visualization**, and any scenario where a natural-looking scene is required.

Comparison:

- **Depth Perception**: Present in perspective, absent in parallel.
- **Realism**: Perspective appears more lifelike.
- **Accuracy**: Parallel is better for technical accuracy.
- **Use Case**: Parallel for blueprints and CAD; perspective for rendering scenes with depth.

Conclusion:

Use **parallel** when precise dimensions matter. Use **perspective** when realism and visual depth are desired. Each serves distinct purposes in different domains.

Q. What is the perspective projection matrix and how does it transform 3D coordinates to 2D coordinates on the image plane? What are the different types of parallel projections and how are they used in computer graphics applications? (Nov 24)

Ans. A **perspective projection matrix** is used in 3D computer graphics to simulate the way objects appear smaller as they get farther from the viewer, creating a realistic depth effect. It transforms 3D coordinates (x,y,z)(x, y, z)(x,y,z) into 2D coordinates (x′,y′)(x', y')(x′,y′) on the image plane by applying a matrix that introduces a perspective divide. The general form of the transformation is:

$$x' = \frac{x}{z}, \quad y' = \frac{y}{z}$$

This matrix creates a **vanishing point** effect, where parallel lines converge at a point in the distance, enhancing realism in 3D scenes such as games and simulations.

Types of Parallel Projections:

1. **Orthographic Projection**:

    - Projects 3D points directly onto the view plane using parallel lines.
    - No perspective distortion—object sizes remain constant.
    - Commonly used in engineering drawings and CAD software.
2. Oblique Projection:
    - A form of parallel projection where the projectors are not perpendicular to the projection plane.
    - Includes cabinet and cavalier projections.
    - Useful for technical illustrations where depth needs to be shown without perspective.

Usage in Applications:

- Perspective projections for realism in visual media.
- Parallel projections for accuracy in design and drafting. Each has distinct advantages depending on the graphical goal.