USN: 1KS18CS097

SIGN: Sourabh

Name: Sourabh Santosh Kamble.

USN : 1KS18CS097

Semester: VI

Section: CSE 'B'

Subject: Computer Graphics and Visualization

Subject Code: 18CS62

Signature: Sourabh

## INTERNAL ASSESMENT -1

### PART-A

1.a. **APPLICATIONS OF COMPUTER GRAPHICS:**

Some of the known applications of computer graphics are,

i) Graphs and charts.
ii) Computer Aided Design (CAD).
iii) Virtual Reality (VR)
iv) Data Visualization.
v) Education and Training.
vi) Computer Art.
vii) Image Processing.
viii) Graphical User Interface (GUI).
ix) Entertainment.

**GRAPHS AND CHART:**

a) An early application of computer graphics was to plot simple data using a character printer.

b) Graphs and charts helps to summarize technical, mathematical data for Research, report generation etc.

EDUCATION AND TRAINING:

a) Computer graphics can be used to provide an graphical perpective to educating and providing training to various individuals.

b) Simulator tools can be used to provide an experience to individuals in fields like air crafts, aeroplanes, flights, etc.

| RASTER SCAN DISPLAY | RANDOM SCAN DISPLAY |
|---|---|
| The electron beam is swept across the entire screen from top to bottom, one row at a time. | The electron beam is directly pointed at the position on screen where picture is to be drawn. |
| Picture is stored as pixels in an refresh array - buffer. | Picture is stored in line drawing instruction in a display file. |
| Resolution is poor as it produces zig-zag lines leading to discrete points. | Resolution is good as CRT points out to a particular position on screen. |
| Screen points or pixels are used to draw image. | Mathematical functions are used to draw image. |

1.b. To illustrate Bresenan's algorithm approach

i) We consider conversion process for lines with positive slope less that 1.0.

ii) Pixel positions along a line path determined by sampling at unit intervals of x. starting from end point (x0,y0) of a given line.

iii) Consider the equation of a straight line

$$y = mx + C \cdot \text{ where}$$

$$m = \frac{dy}{dx}$$

ALGORITHM:

1. Input two end points and store left input as $(x_0, y_0)$.

2. Plot first point by setting colour frame buffer position $(x_0, y_0)$.

3. (if $m < 01$) Calculate $\Delta x, \Delta y, 2\Delta y, 2\Delta x, 2\Delta y - 2\Delta x$ and obtain ~~starting~~ starting value for decision parameter as. $P_k = 2\Delta y - \Delta x$.

4. At each $x_k$ along the line from $k = 0$ perform following, (if ~~$P_k < 1$~~)

a) if $P_k < 0$ then next point is $(x_{k+1}, y_k)$

b) if $P_k >= 0$ then next point is $(x_{k+1}, y_{k+1})$

c) $P_{k+1} = P_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k)$

5. Repeat step (4) ($\Delta x - 1$) times.

if ($m >= 1$)

3. Calculate $P_k = 2\Delta x - \Delta y$.

4. At each $x_k$ from $k = 0$ perform following.

a) if $P_k < 0$ then next point is $(x_k, y_{k+1})$

b) if $P_k >= 0$ then next point is $(x_{k+1}, y_{k+1})$

$\Rightarrow P_{k+1} = P_k + 2\Delta x$.

5. Repeat step (4) $\Delta x - 1$) time.

given points are (20,10) and (30,18).

Let $(x_0, y_0) = (20, 10)$

$\Rightarrow \Delta x = 30 - 20 = 10$

$\Delta y = 18 - 10 = 8$

$m = \frac{\Delta y}{\Delta x} = \frac{8}{10} = 0.8 < 1$

$m < 1$

$\Delta x = 10 \Rightarrow 2\Delta x = 2 \times 10 = 20$

$\Delta y = 8 \Rightarrow 2\Delta y = 2 \times 8 = 16$

At $K = 0$, $x_0 = 20$, $y_0 = 10$

$P_k = P_0 = 2\Delta y - \Delta x = 16 - 10 = 6$

$P_0 = 6 > 0 \Rightarrow P_0 > 0$

$\Rightarrow (x_{k+1}, y_{k+1}) = (x_{k+1}, y_{k+1}) = (21, 11)$

~~At $k = 1$, $x_1 = 21$, $y_1 = 11$~~

~~$P_{k+1} =$~~ $P_1 =$

USN - 1KS 18CS097

SIGN - Aswath

$P_{K+1} = P_K + 2\Delta y - 2\Delta x(y_{K+1} - y_K)$

$P_1 = 6 + 16 - 20(11 - 10)$

$P_1 = 6 + 16 - 20$

$P_1 = 2 > 0$

At $K = 1$, $x_1 = 21$, $y_1 = 11$, $P_1 = 2 > 0$

$\Rightarrow (x_{K+1}, y_{K+1}) = ((x_{K+1}), (y_{K+1})) = (22, 12)$

$P_{K+1} = P_K + 2\Delta y - 2\Delta x(y_{K+1} - y_K)$

$P_2 = 2 + 16 - 20(12 - 11)$

$P_2 = -2 < 0$

At $K = 2$, $x_2 = 22$, $y_2 = 12$, $P_2 = -2$

$\Rightarrow (x_{K+1}, y_{K+1}) = (x_{K+1}, y_K) = (23, 12)$

$P_{K+1} = P_K + 2\Delta y - 2\Delta x(y_{K+1} - y_K)$

$P_3 = -2 + 16 - 20(12 - 12)$

$P_3 = 14 > 0$

At $K = 3$, $x_3 = 23$, $y_3 = 12$, $P_3 = 14$

$\Rightarrow (x_{K+1}, y_{K+1}) = (x_{K+1}, y_{K+1}) = (24, 13)$

$P_{K+1} = P_K + 2\Delta y - 2\Delta x(y_{K+1} - y_K)$

$P_4 = 14 + 16 - 20(13 - 12)$

$P_4 = 10 > 0$

At $K = 4$, $x_4 = 24$, $y_4 = 13$, $P_4 = 10$

$\Rightarrow (x_{K+1}, y_{K+1}) = (x_{K+1}, y_{K+1}) = (25, 14)$

$P_{K+1} = P_K + 2\Delta y - 2\Delta x(y_{K+1} - y_K)$

$P_5 = 10 + 16 - 20(14 - 13) = 6 > 0$

USN - 1KS18CS097
SIGN - Aqueab



At $K=5$, $x_5 = 25$, $y_5 = 14$, $P_5 = 6$

$\Rightarrow (x_{K+1}, y_{K+1}) = ((x_K + 1), (y_K + 1)) = (26, 15)$

$P_{K+1} = P_K + 2\Delta y - 2\Delta x (y_{K+1} - y_K)$

$P_6 = 6 + 16 - 20(15 - 14)$

$P_6 = 2 > 0$

At $K=6$, $x_6 = 26$, $y_6 = 15$, $P_6 = 2$

$\Rightarrow (x_{K+1}, y_{K+1}) = (x_K + 1, y_K + 1) = (27, 16)$

$P_{K+1} = P_K + 2\Delta y - 2\Delta x (y_{K+1} - y_K)$

$P_7 = 2 + 16 - 20(16 - 15)$

$P_7 = -2 < 0$

At $K=7$, $x_7 = 27$, ~~$y_7$~~ $y_7 = 16$, $P_7 = -2$

$\Rightarrow (x_{K+1}, y_{K+1}) = (x_K + 1, y_K) = (28, 16)$

$P_{K+1} = P_K + 2\Delta y - 2\Delta x (y_{K+1} - y_K)$

$P_8 = -2 + 16 - 20(16 - 16)$

$P_8 = 14 > 0$

At $K=8$, $x_8 = 28$, $y_8 = 16$, $P_8 = 14$

$\Rightarrow (x_{K+1}, y_{K+1}) = (x_K + 1, y_K + 1) = (29, 17)$

$P_{K+1} = P_K + 2\Delta y - 2\Delta x (y_{K+1} - y_K)$

$P_9 = 14 + 16 - 20(17 - 16)$

$P_9 = 10 > 0$

At $K=9$, $x_9 = 29$, $y_9 = 17$, $P_9 = 10$

$\Rightarrow (x_{K+1}, y_{K+1}) = (x_K + 1, y_K + 1) = (30, 18)$

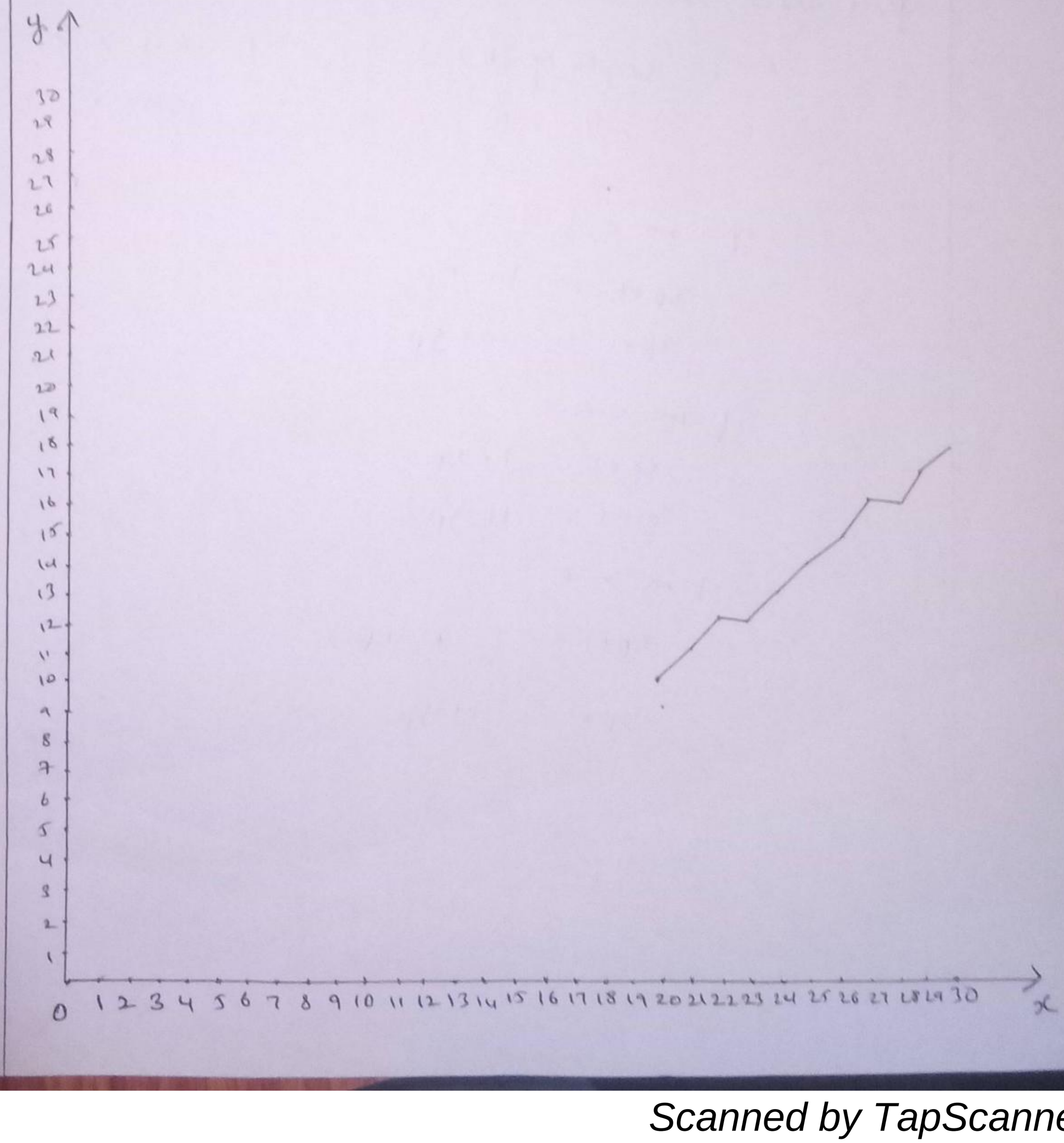
As second end point is obtained iteration stops.

USN - 1KS18CS097

SIGN - Pourabh



| K | $x_k$ | $y_k$ | $P_k$ | $(x_{k+1}, y_{k+1})$ |
|---|---|---|---|---|
| 0 | 20 | 10 | 6 | (21, 11) |
| 1 | 21 | 11 | 2 | (22, 12) |
| 2 | 22 | 12 | -2 | (23, 12) |
| 3 | 23 | 12 | 12 | (24, 13) |
| 4 | 24 | 13 | 10 | (25, 14) |
| 5 | 25 | 14 | 6 | (26, 15) |
| 6 | 26 | 15 | 2 | (27, 16) |
| 7 | 27 | 16 | -2 | (28, 16) |
| 8 | 28 | 16 | 14 | (29, 17) |
| 9 | 29 | 17 | 10 | (30, 18) |



USN - 1KS18CS097

SIGN - Sowmabh



1.c. DDA ALGORITHM

1. Calculate $\Delta x = x_n - x_0$, where $x_n$ - ending, $x_0$ is starting point.

2. Calculate $\Delta y = y_n - y_0$, where $y_n$ - ending, $y_0$ - starting point.

$m = \frac{\Delta y}{\Delta x}$ has to be calculated.

2. Calculate how many points have to be generated between $x_0$ and $x_n$, $y_0$ and $y_n$

i.e., length of (k) = $\Delta y$ if $\Delta y > \Delta x$

length of (k) = $\Delta x$ if $\Delta x > \Delta y$

3 - if $m < 1$

$x_{p+1} = 1 + x_p$

$y_{p+1} = m + y_p$

if $m = 0$

$x_{p+1} = 1 + x_p$

$y_{p+1} = 1 + y_p$

if $m > 1$

$x_{p+1} = \frac{1}{m} + x_p$

$y_{p+1} = 1 + y_p$

## PART - B

~~9~~ 4.

4.a. **OPENGL OUTPUT PRIMITIVE FUNCTIONS :**

Some of the known OpenGL output primitive functions are :

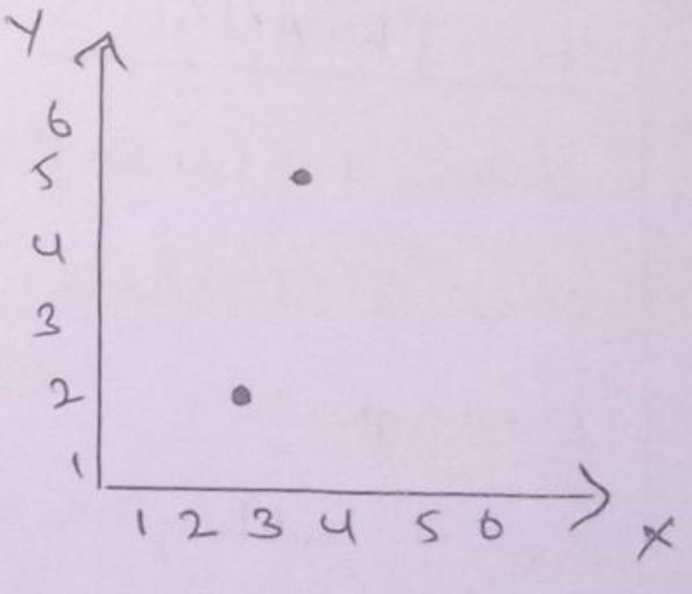
i) GL_POINTS
ii) GL_LINES
iii) GL_LINE_LOOP
iv) GL_TRIANGLES
v) GL_TRIANGLE_STRIP
vi) GL_TRIANGLE_FAN
vii) GL_QUADS
viii) GL_QUAD_STRIP
ix) GL_POLYGON

**GL_POINTS**

~~GL_POINTS~~ : This openGL function is used to represent points based on given vertex.

Eg:

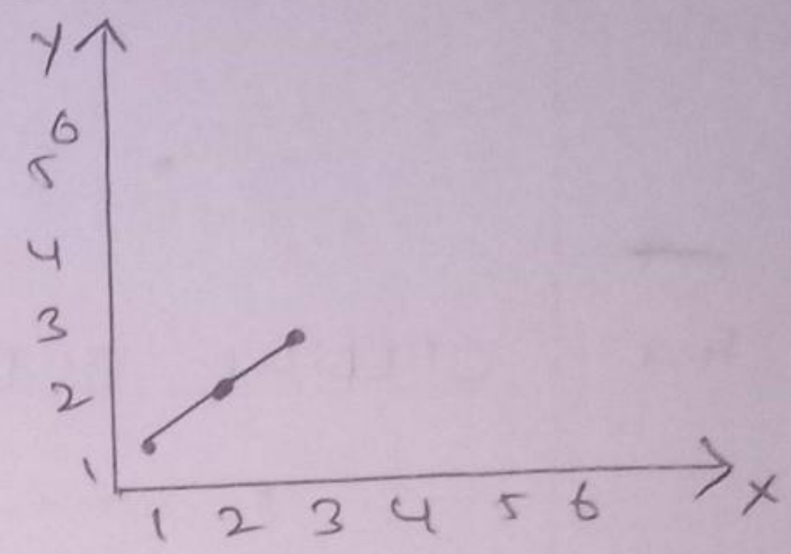
```
glBegin (GL_POINTS);
~~glVertex1i (3);~~
~~glVertex1i(6~~
glVertex2i (3,2);
glVertex2i (4,5);
glEnd();
```




**GL_LINES** : This OpenGL function is used to represent a line segment connecting two lines.

Eg:

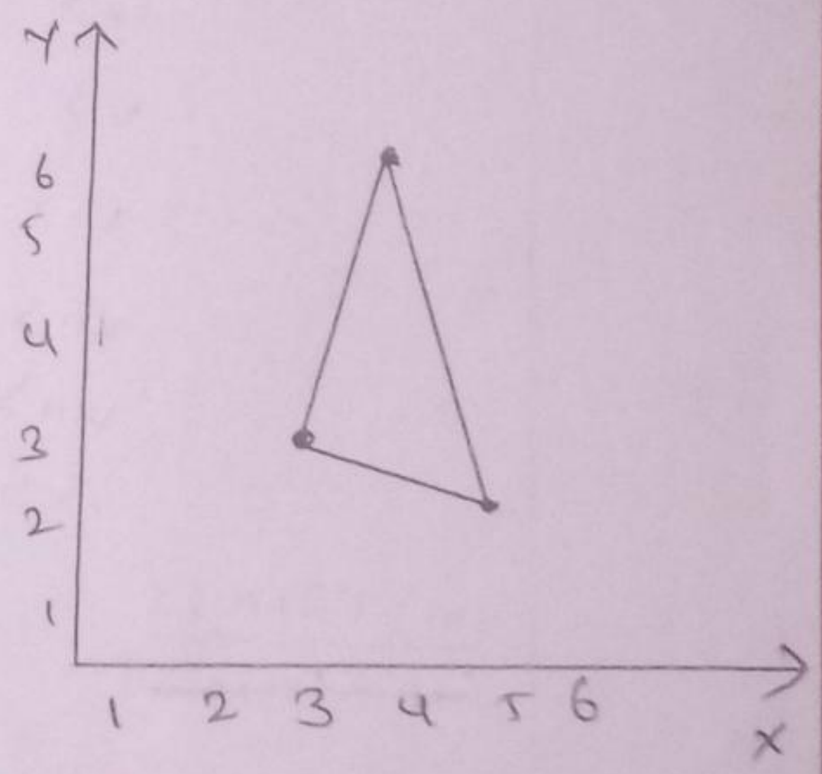
```
glBegin (GL_LINES);
 glVertex2i (1,1);
 glVertex2i (2,2);
 glVertex2i (3,3);
 glEnd();
```




GL_LINES_LOOP: This OpenGL function is similar to GL_LINES in addition the last point draws a line segment with first point.

Eg:

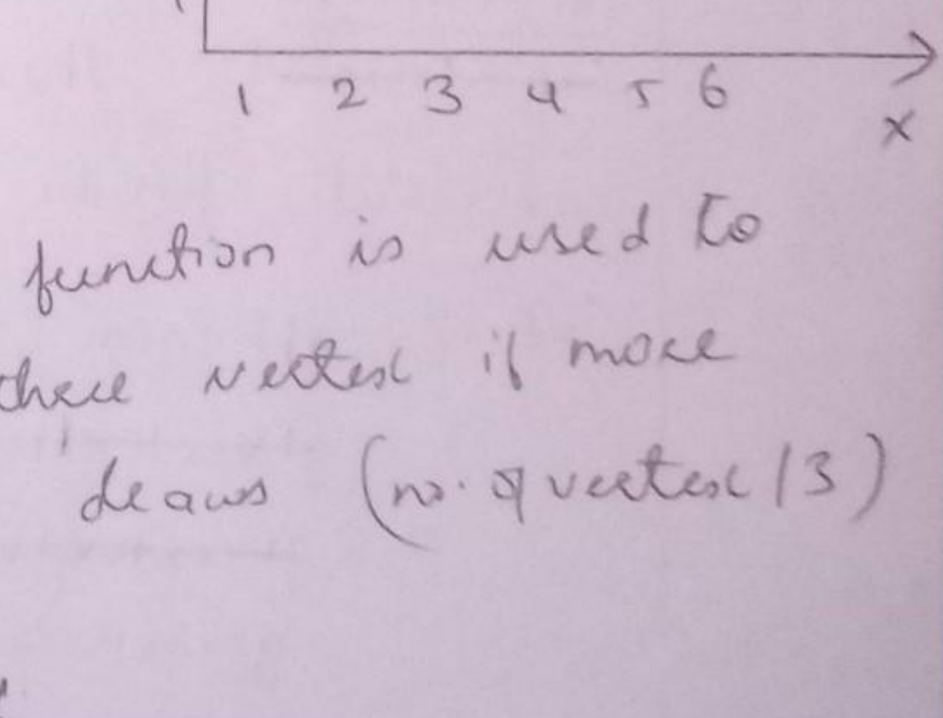
```
glBegin (GL_LINE_LOOP);
 glVertex2i (3,3);
 glVertex2i (4,6);
 glVertex2i (5,2);
 glEnd();
```




GL_TRIANGLES: This OpenGL function is used to draw a Triangle - ~~the~~ with three vertex if more than 3 vertices are input it draws (no. of vertex/3) triangles.

Eg:

```
glBegin(GL_TRIANGLES);
glVertex2i(4,4);
glVertex2i(5,2);
glVertex2i(6,3);
glVertex2i(1,1);
glVertex2i(2,2);
glVertex3i(3,1);
```

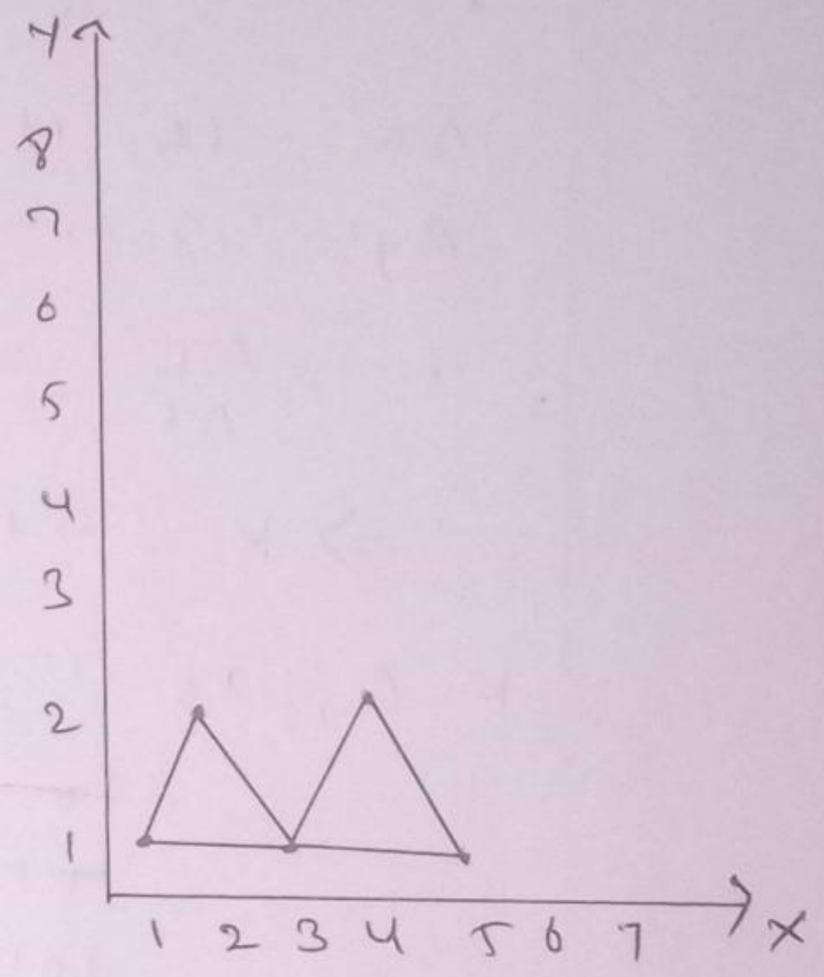


glEnd();

GL_TRIANGLE_STRIP : This function used to draw triangles in a strip manner where the one point is common between two triangles.

Eg:

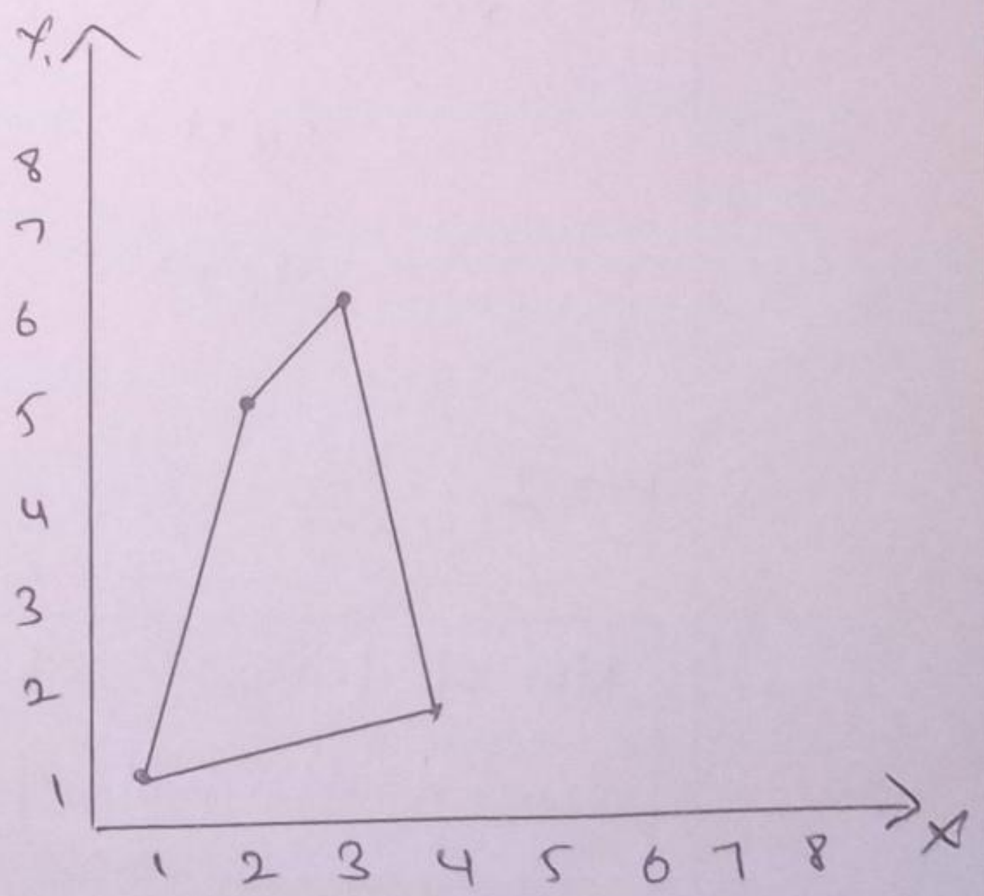
```
glBegin(GL_TRIANGLE_STRIP);
glVertex2i(1,1);
glVertex2i(2,2);
glVertex2i(3,1);
glVertex2i(4,2);
glVertex2i(5,1);
glEnd();
```




GL_QUADS: This openGL function is used. to draw quadrilaterals.

Eg:

```
glBegin(GL_QUADS);
glVertex2i(1,1);
glVertex2i(2,5);
glVertex2i(3,6);
glVertex2i(4,2);
glEnd();
```



~~PART-A~~

1.c. Eg:

Let the points be (11, 12) and (15, 17)

$x_0 = 11, \quad y_0 = 12, \quad x_n = 15, \quad y_n = 17$

$\Rightarrow \Delta x = 15 - 11 = 4$

$\Delta y = 17 - 12 = 5$

$m = \frac{\Delta y}{\Delta x} = \frac{5}{4} = 1.1 > 1$

$\Rightarrow K = \Delta y = 5$

$K = 0, \quad x_0 = 11, \quad y_0 = 12,$

$x_{p+1} = \frac{1}{m} + x_p = \frac{4}{5} + 11 = \frac{59}{5}$

$y_{p+1} = 1 + y_p = 12 + 1 = 13$

$K = 1, \quad x_1 = \frac{59}{5}, \quad y_0 = 13$

$x_{p+1} = \frac{63}{5}$

$y_{p+1} = 14$

~~K = 2~~

No. of pages - 12

Questions attempted:

1.a, 1.b, 1.c

4.a