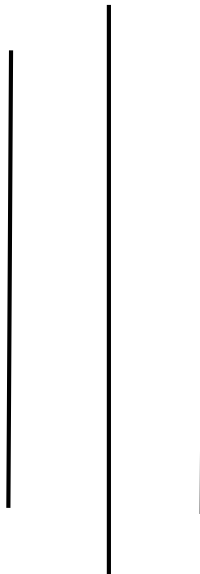


# SAGARMATHA ENGINEERING COLLEGE

*(TU Affiliated)*

**Sanepa, Lalitpur**



LAB NO: 2

A LAB REPORT ON

BRESENHAM'S LINE ALGORITHM

**Submitted By:**

Name:.....

Faculty/Year:.....

Roll No:.....

Date:.....

**Submitted To:**

Department of electronics and Computer Engineering

Signature:.....

Date:.....



## COMPUTER GRAPHICS LAB-02

### TITLE

### BRESENHAM'S LINE ALGORITHM

### OBJECTIVES

- ✓ To be familiar with fundamental knowledge of Bresenham's line drawing algorithm and its implementation

### HARDWARE/SOFTWARE REQUIRED

- ✓ C Compiler

### RELATED THEROY

#### BLA

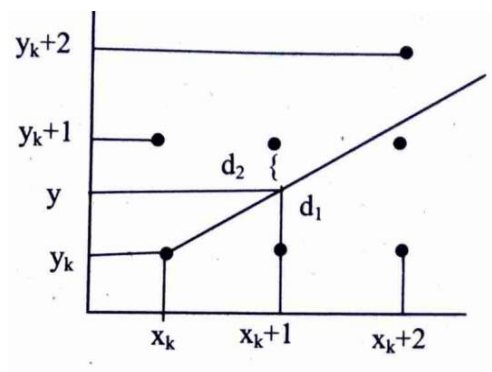
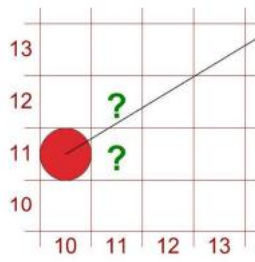
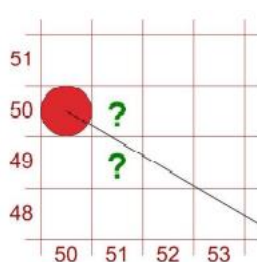
The BLA is a more efficient method used to plot pixel position along a straight-line path.

Its idea is to move across the x-axis in unit intervals and at each step choose between two y coordinates e.g. which pixel to draw?

(11, 11) or (11, 12)?

(51, 50) or (51, 49)?

The solution is to choose that which is closer to the original line.





## COMPUTER GRAPHICS LAB-02

### ALGORITHM

Consider one point of the line as  $(X_1, Y_1)$  and the second point of the line as  $(X_2, Y_2)$ .

```
1. Input two points  $(x_1, y_1)$  and  $(x_2, y_2)$ 
2. Compute  $\Delta x = |x_2 - x_1|$  &  $\Delta y = |y_2 - y_1|$ 
3. If  $(x_2 > x_1)$   $lx = 1$  else  $lx = -1$ 
4. If  $(y_2 > y_1)$   $ly = 1$  else  $ly = -1$ 
5. Plot first point  $(x_1, y_1)$ 
6. If  $(\Delta x > \Delta y)$  {                                     /* i.e. when  $|m| < 1$  */
    → calculate  $p_0 = 2\Delta y - \Delta x$ 
    → Starting at  $k = 0$  to  $\Delta x$  times, repeat
        If  $p_k < 0$                                        /* next point  $(x_k + 1, y_k)$  */
             $x_{k+1} = x_k + lx$  ,  $y_{k+1} = y_k$ 
             $P_{k+1} = p_k + 2\Delta y$ 
        else                                             /* next point  $(x_k + 1, y_k + 1)$  */
             $x_{k+1} = x_k + lx$  ,  $y_{k+1} = y_k + ly$ 
             $P_{k+1} = p_k + 2\Delta y - 2\Delta x$ 
    }ENDIF
7. Else                                             /* i.e. when  $|m| > 1$  */
    {
        → calculate  $p_0 = 2\Delta x - \Delta y$ 
        → Starting at  $k = 0$  to  $\Delta y$  times, repeat
            If  $p_k < 0$                                    /* next point  $(x_k, y_k + 1)$  */
                 $x_{k+1} = x_k$ 
                 $y_{k+1} = y_k + ly$ 
                 $P_{k+1} = p_k + 2\Delta x$ 
            else                                         /* next point  $(x_k + 1, y_k + 1)$  */
                 $x_{k+1} = x_k + lx$ 
                 $y_{k+1} = y_k + ly$ 
                 $P_{k+1} = p_k + 2\Delta x - 2\Delta y$ 
    }
```



## COMPUTER GRAPHICS LAB-02

### IMPLEMENTATION OF BLA LINE ALGORITHM



## COMPUTER GRAPHICS LAB-02

### OUTPUT

### RESULT:

Program is compiled, BLA line algorithm implementation was done.