1. Implement Brenham's line drawing algorithm for all types of slope.

```
#include<GL/glut.h>
#include<stdio.h>
int a1, b1, a2, b2;

void draw_pixel(int x, int y)
{
      glColor3f(1.0,0.0,0.0);
      glBegin(GL_POINTS);
      glVertex2i(x, y);
      glEnd();
}
void brenhams_line_draw(int a1, int b1, int a2, int b2)
{
      int dx=a2-a1,dy=b2-b1;
      int p=2*dy-dx;
      int twoDy=2*dy;
      int twoDyMinusDx=2*(dy-dx); // paranthesis are required
      int x=a1,y=b1;
      if(dx<0)
      {
            x=a2;
            y=b2;
            a2=a1;
      }
draw_pixel(x, y);
while(x<a2)
{
      x++;
      if(p<0)
      p+=twoDy;
            else
            {
                  y++;
                  p+=twoDyMinusDx;
            }
      draw_pixel(x, y);
}
}
void myInit()
{
      glClearColor(0.0,0.0,0.0,1.0);
      glMatrixMode(GL_PROJECTION);
      glLoadIdentity();
      gluOrtho2D(0.0, 500.0, 0.0, 500.0);
      glMatrixMode(GL_MODELVIEW);
}
void display()
{
      glClear(GL_COLOR_BUFFER_BIT);
```

```
      brenhams_line_draw(a1, b1, a2, b2);
      glFlush();
}
int main(int argc, char **argv)
{
      printf( "Enter Start Points (a1,b1)\n");
      scanf("%d %d", &a1, &b1);
      printf( "Enter End Points (a2,b2)\n");
      scanf("%d %d", &a2, &b2);
      glutInit(&argc, argv);
      glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
      glutInitWindowSize(500, 500);
      glutInitWindowPosition(0, 0);
      glutCreateWindow("Bresenham's Line Drawing");
      myInit();
      glutDisplayFunc(display);
      glutMainLoop();
}
```

OUTPUT :