EXPERIMENT – 2

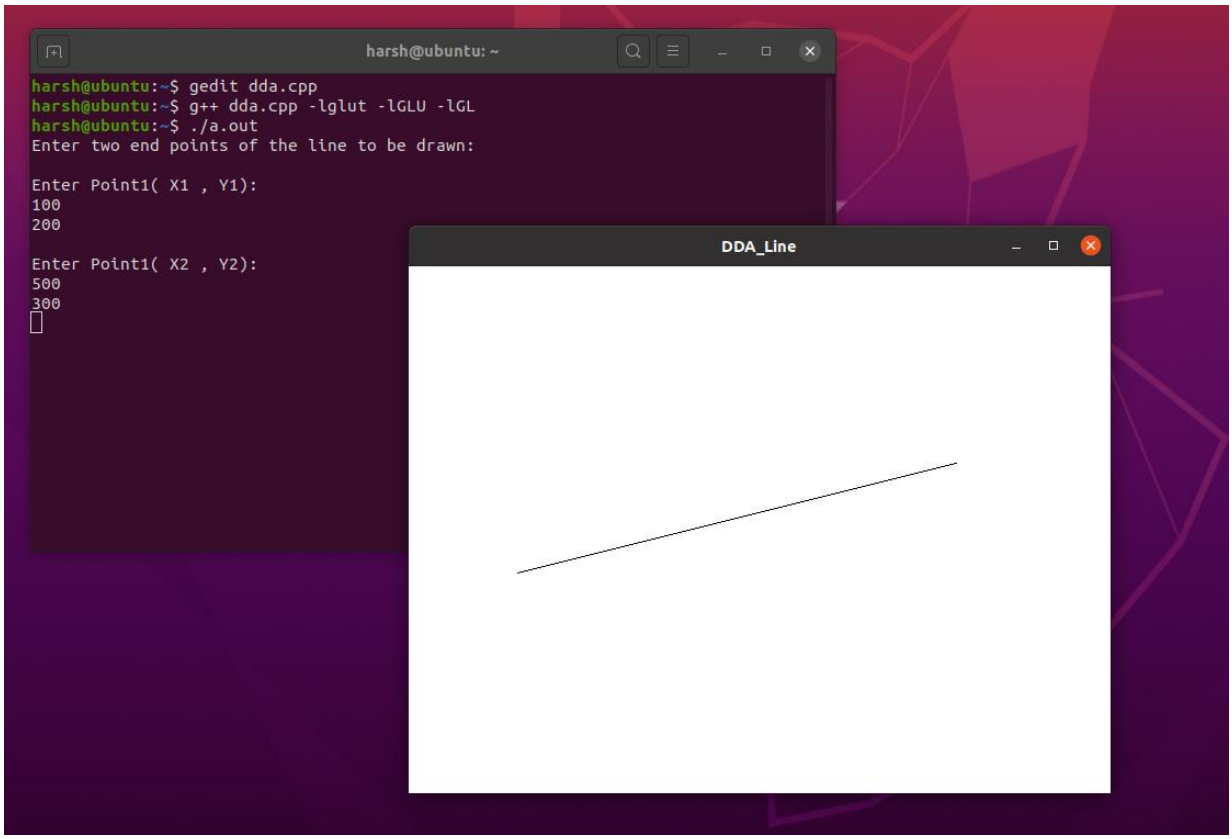
Drawing a line using DDA Algorithm

CODE :

```
#include <stdio.h>
#include<stdlib.h>
#include <math.h>
#include <GL/glut.h>

double X1, Y1, X2, Y2;

float round_value(float v)
{
  return floor(v + 0.5);
}
void LineDDA(void)
{
  double dx=(X2-X1);
  double dy=(Y2-Y1);
  double steps;
  float xInc,yInc,x=X1,y=Y1;

  steps=(abs(dx)>abs(dy))?(abs(dx)):(abs(dy));
  xInc=dx/(float)steps;
  yInc=dy/(float)steps;


  glClear(GL_COLOR_BUFFER_BIT);

  glBegin(GL_POINTS);

  glVertex2d(x,y);
  int k;

  for(k=0;k<steps;k++)
  {
    x+=xInc;
    y+=yInc;

    glVertex2d(round_value(x), round_value(y));
  }
  glEnd();

  glFlush();
}
void Init()
```

```
{

 glClearColor(1.0,1.0,1.0,0);

 glColor3f(0.0,0.0,0.0);

 gluOrtho2D(0 , 640 , 0 , 480);
}
int main(int argc, char **argv)
{
 printf("Enter two end points of the line to be drawn:\n");

 printf("\nEnter Point1( X1 , Y1):\n");
 scanf("%lf%lf",&X1,&Y1);
 printf("\nEnter Point1( X2 , Y2):\n");
 scanf("%lf%lf",&X2,&Y2);
 glutInit(&argc,argv);

 glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);

 glutInitWindowPosition(0,0);
 glutInitWindowSize(640,480);

 glutCreateWindow("DDA_Line");

 Init();

 glutDisplayFunc(LineDDA);

 glutMainLoop();
}
```

OUTPUT :


harsh@ubuntu: ~
harsh@ubuntu:~$ gedit dda.cpp
harsh@ubuntu:~$ g++ dda.cpp -lglut -lGLU -lGL
harsh@ubuntu:~$ ./a.out
Enter two end points of the line to be drawn:
Enter Point1(X1 , Y1):
100
200
Enter Point1(X2 , Y2):
500
300
DDA_Line