```
#include<iostream>
#include<stdlib.h>
#include<stdio.h>
#include<GL/glut.h>
#include<math.h>
#define ROUND(x)((int)(x+0.5))
using namespace std;
int x1,x2,z1,z2;

void draw_pixel(int x, int y)
{
glColor3f(1.0,0.0,0.0);
glBegin(GL_POINTS);
glVertex2i(x,y);
glEnd();
}


void drawline(int X1, int Y1, int X2, int Y2)
{
float x,y,dx,dy,length; int i;
dx=abs(X2-X1);
dy=abs(Y2-Y1);
if(dx>=dy)
 length=dx;
else length=dy;
dx=(X2-X1)/length;
dy=(Y2-Y1)/length;
x=X1;
y=Y1;
i=1;
while(i<=length)
{
draw_pixel(ROUND(x),ROUND(y));
x=x+dx;
y=y+dy;
i=i+1;
}
glFlush();
}

void drawpatt(int ax, int ay, int bx, int by, int cx, int cy, int dx, int dy, int
n)
{
int m1x, m1y, m2x, m2y, m3x, m3y, m4x, m4y;
drawline(ax,ay,bx,by);
drawline(bx,by,cx,cy);
drawline(cx,cy,dx,dy);
drawline(dx,dy,ax,ay);
```

```
//midpoint calculations;

m1x=(ax+bx)/2;
m1y=(ay+by)/2;
m2x=(bx+cx)/2;
m2y=(by+cy)/2;
m3x=(cx+dx)/2;
m3y=(cy+dy)/2;
m4x=(dx+ax)/2;
m4y=(dy+ay)/2;
n--; if(n!=0)
{
drawpatt(m1x, m1y, m2x, m2y, m3x, m3y, m4x, m4y,n);
}
}

void display(void)
{
float x,y,dx,dy,length;
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(1.0,0.0,0.0);
glBegin(GL_LINES);
glVertex2i(-200,0);
glVertex2i(200,0);
glVertex2i(0,-200);
glVertex2i(0,200);
glEnd();
drawpatt(x1,z1,x1,z2,x2,z2,x2,x1,5);
}

void Init(void)
{
glClearColor(1,1,1,0);
gluOrtho2D(-200.0,200.0,-200.0,200.0);
}

int main(int argc, char **argv)
{
/* cout<<"\n Enter the value of left bottom x1::"; cin<<x1;
cout<<"\n Enter the value of left bottom y1:"; cin<<y1;
cout<<"\n Enter the value of right top x2:"; cin<<x2;
cout<<"\n Enter the value of right top y2:"; cin<<y2; */

printf("Enter the value of left bottom x1:");
scanf("%d",&x1);
printf("Enter the value of left bottom y1:");
scanf("%d",&z1);
printf("Enter the value of right top x2:");
scanf("%d",&x2);
printf("Enter the value of right top y2:");
```

```
scanf("%d",&z2);

glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(500,500);
glutInitWindowPosition(100,100);
glutCreateWindow("Pattern Drawing");
Init();
glClear(GL_COLOR_BUFFER_BIT);
glutDisplayFunc(display);
glFlush();
glutMainLoop();
return 0;
}
```