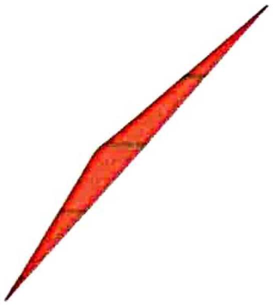
program 4

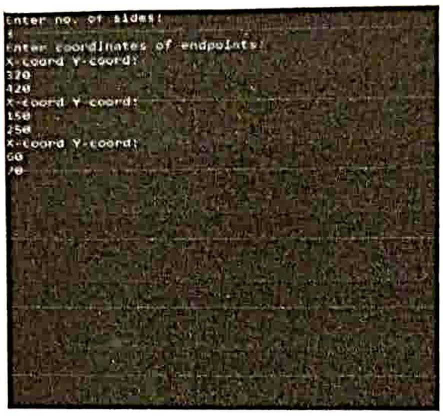
Write a program to fill any given polygon using scan-line area filing algorithm.

```
#include <stdlib.h>
#include <gl/glut.h>
#include <stdio.h>
#include <algorithm>
#include <iosream>
#include <windows.h>

using namespace std;
float x[100], y[100];
 int n,m;
int wx=500, wy=500;
 static float intx[10]={0};

 void draw_line(float x1, float y1, float x2, float y2)
    {
            sleep(100);
            glColor3f(1,0,0);
            glBegin(GL-LINES);
            glVertex2f(x1, y1);
            glVertex2f(x2,y2);
           glEnd();
           glFlush();
    }
```

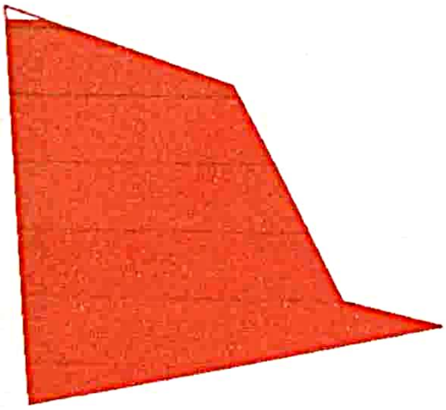
```
void   EdgeDetect (float x1, float y1, float x2, float y2, int scanline)
    {
            float  temp;
            if (y2 <y1)
                {
                    temp = x1 ; x1 = x2 ; x2 = temp;
                    temp = y1 ; y1 = y2 ; y2 = temp;
                }
                if (scanline > y1 && scanline <y2)
                        intx[m++] = x1 + (scanline -y1)*(x2-x1)/(y2-y1);

        }

        void      scanfill (float x[], float y[])
                  {
                    for (int s1=0; s1<=wy ; s++)

                          {
                              m=0;
                            for (int i=0; i<n; i++)
                              {
                                  EdgeDetect(x[i], y[i], x[(i+1)%n],
                                                 y[(i+1)%n], s1);
                              }
                            sort (intx, (intx+m));
                            if (m>=2)
                                 for(int i=0; i<m; i= i+2)
                                      draw-line(intx [i], s1, intx [i+1],
                                                                   s1);
                  }    }
```

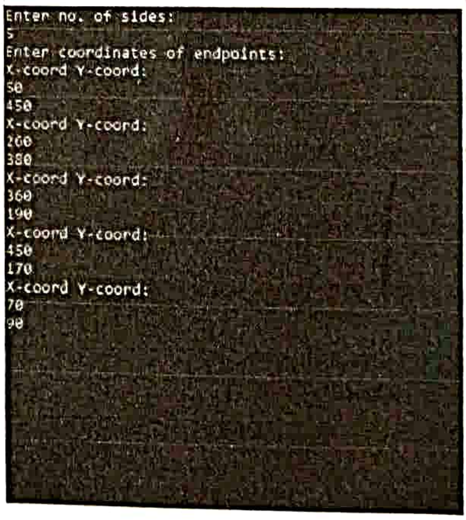
```
void   display_filled_polygon()
  {
       glClear (GL_COLOR_BUFFER_BIT);
       glLineWidth (2);
       glBegin (GL_LINE_LOOP);
       for (int i=0 ; i<n;i++)
                  glVertex2f (x[i], y[i]);
        glEnd();
        scanfill (x,y);
  }

void myInit()
    {
        glClearColor (1,1,1,1);
        glColor3f (0,0,1);
        glPointSize (1);
        gluOrtho2D (0,wx,0,wy);
  }
 void   main(int ac, char* av[])
   {
          glutInit(&ac,av);
          printf("Enter no of sides:\n");
          scanf_s ( "%d",&n);
        printf(" Enter  coordinates of endpoints:\n");
          for (int i=0;i<n;i++)
               {
                    printf ("x-coord y-coord:\n");
                    scanf_s("%f %f", &x[i],&y[i]);
               }
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutWindowSize(500,500);
    glutWindowPosition(0,0);
    glutCreateWindow("scanline");
    glutDisplayFunc(display_filled_polygon);
    myInit();
    glutMainLoop();
}
```



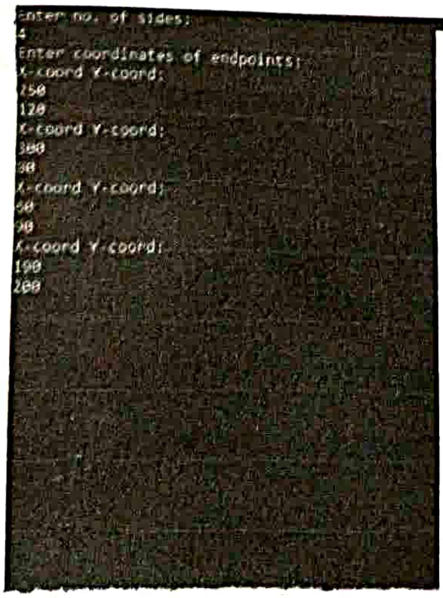


```
Enter no. of sides:
3
Enter coordinates of endpoints:
X-coord Y-coord:
370
420
X-coord Y-coord:
150
250
X-coord Y-coord:
60
70
```





```
Enter no. of sides:
5
Enter coordinates of endpoints:
X-coord Y-coord:
50
450
X-coord Y-coord:
200
380
X-coord Y-coord:
360
190
X-coord Y-coord:
450
170
X-coord Y-coord:
70
90
```



```
Enter no. of sides:
4
Enter coordinates of endpoints:
X-coord Y-coord:
250
120
X-coord Y-coord:
300
30
X-coord Y-coord:
50
90
X-coord Y-coord:
190
200
```