program 8

Write a program to implement the cohen-Hodgeman polygon clipping algorithm. make provision to specify the input polygon and window for clipping.

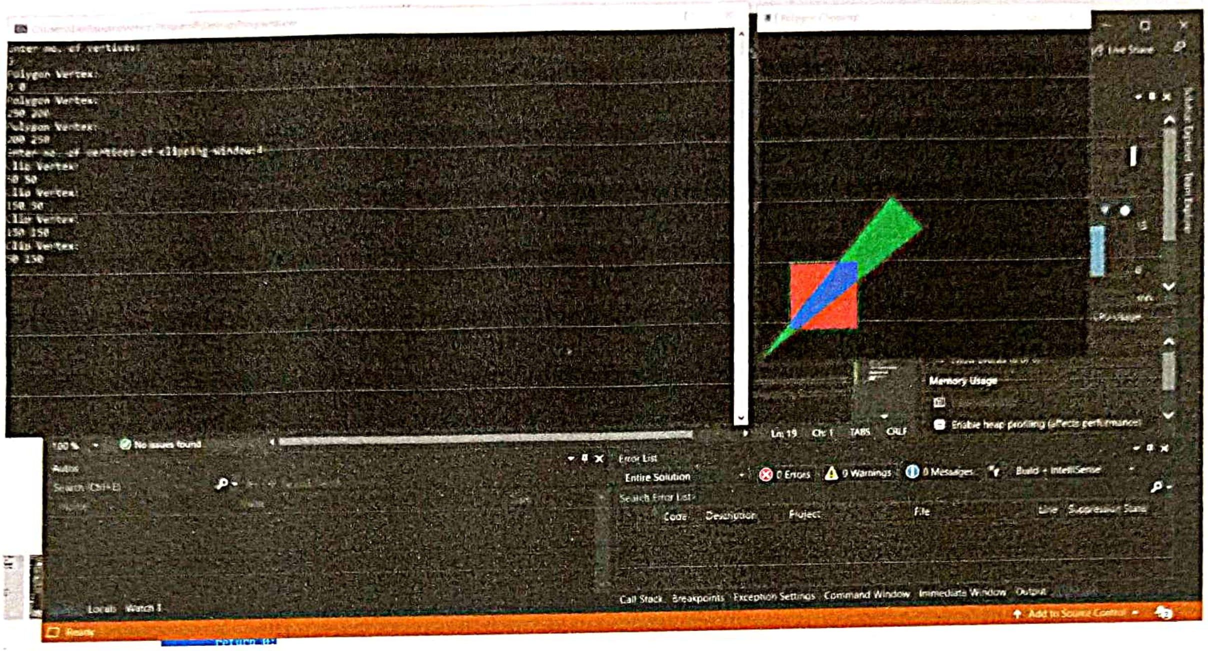
```
 #include <iostream>
 #include <GL/glut.h>

int poly_size, poly_points[20][2], org_poly_size, org_poly_points[20][2]
    clipper_size, clipper_points[20][2];

  const int MAX_POINTS = 20;

void drawpoly(int p[][2], int n){

        glBegin (GL_POLYGON);
        for (int i=0;i<n;i++)
              glVertex2f (p[i][0], p[i][1]);
         glEnd();
   }

int x_intersect (int x1, int y1, int x2, int y2, int x3, int y3,
                 int x4, int y4)
  {
          int num = (x1*y2 - y1*x2) * (x3-x4) - (x1-x2)
                      * (x3*x4 - y3*y4);

          int den = (x1-x2)* (y3-y4) - (y1-y2)*(x3-x4);
            return num/den;
  }
```

```
int   y-intersect(int x1, int y1, int x2, int y2, int x3, int y3,
            int x4 , int y4)
  {
      int num=(x1*y2-y1*x2)*(y3-y4)-(y1-y2)*
                   (x3*y4-y3*x4);
      int den= (x1-x2)*(x3-x4)-(y1-y2)*(x3-x4);
      return num/den;
  }

void clip(int poly-point[][2], int& poly-size, int x1, int y1, int x2,
                   int y2)
  {
      int new_points[MAX_POINTS][2], new_poly_size=0;
      for (int i=0;i<poly_size;i++)
         {
             int k=(i+1)%poly.size;
             int ix= poly-points[i][0], iy=poly-points[i][1];
             int kx= poly-points[k][0], ky =poly-points[k][1];
             int i-pos=(x2-x1)*(iy-y1)-(y2-y1)*(ix-x1);
             int k-pos=(x2-x1)*(ky-y1)-(y2-y1)*(kx-x1);
if (i_pos >=0 ++ k_pos >=0)
  {
     new_points[new_poly_size][0] = kx;
     new_points[new_poly_size][1] = ky;
     new_poly_size++;
  }
else if (i_pos < 0 ++ k_pos >=0)
  {
     new_points[new_poly_size][0] = x_intersect(x1, y1, x2, y2,
                                      ix, iy, kx, ky);

     new_points[new_poly_size][1] = y_intersect(x1, y1, x2, y2,
              ix, iy, kx, ky);
     new_poly_size++;
     new_points[new_poly_size][0] = kx;
     new_points[new_poly_size][1] = ky;
     new_poly_size++;
  }
  else if (i_pos>=0 ++ k_pos<0)
    {
        new_points[new_poly_size][0] = x_intersect(x1, y1, x2,
                 y2, ix, iy, kx, ky);
        new_points[new_poly_size][0] = y_intersect(x1, y1, x2, y2,
              ix, iy, kx, ky);
       new_poly_size++
    }
    else
      {
      }
  }
```

```
poly-size = new_poly_size;
for(int i=0; i<poly-size;i++)
    {
        poly-points[i][0] =new-points[i][0];
        poly-points[i][1]= new-points[i][1];
    }
}
void init()
   {
       glClearColor (0.0f, 0.0f, 0.0f, 0.0f);
       glMatrixmode (GL-PROJECTION);
       glLoadIdentity();
       glOrtho( 0.0, 500.0, 0.0, 500.0, 0.0, 500.0);
       glClear (GL-COLOR-BUFFER-BIT);
   }
void display()
    {
         init();
         glColor3f (1.0f, 0.0f, 0.0f);
         drawpoly (clipper-points, clipper_size);

         glColor3f (0.0f, 1.0f, 0.0f);
         drawpoly(org-poly-points, org-poly-size);
```

```
for ( int i=0; i<clipper_size ; i++)
 {
    int k = (i+1) % clipper_size;

      clip(poly_points, poly size, clipper_points[i][0],
           clipper_points[i][1], clipper_points[k][0],
           clipper_points[k][1]);

    }
    glColor3f (0.0f, 0.0f, 1.0f);
    drawPoly ( poly_points, poly_size);
     glFlush();
   }

 int main( int argc, char* argv[])
  {
    printf("Enter no. of vertices\n");
    scanf ( "%d", &poly_size);
    org_poly_size = poly_size.

    printf("Enter the no. of vertices of clipping window");
    scanf_s ( "%d", & clipper_size);
    for ( int i=0; i<clipper_size; i++)
     {
        printf(" clip vertex\n");
        scanf_s("%d %d", & clipper_points[i][0],
                         & clipper_points[i][1]);

     }
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition (100,100);
    glutCreateWindow("polygon clipping!");
    glutDisplayFunc (display);
    glutMainLoop();
    return 0;
}
```

OUTPUT:



```
Enter no. of vertices:
3
Polygon Vertex:
0 0
Polygon Vertex:
250 200
Polygon Vertex:
200 250
Enter no. of vertices of clipping window4
Clip Vertex:
50 50
Clip Vertex:
150 50
Clip Vertex:
150 150
Clip Vertex:
50 150
```