Computer Graphics : Transformation (Translate, Scale and Rotate)

Name : Obaidur Rahamn

Reg : 12101006

Code

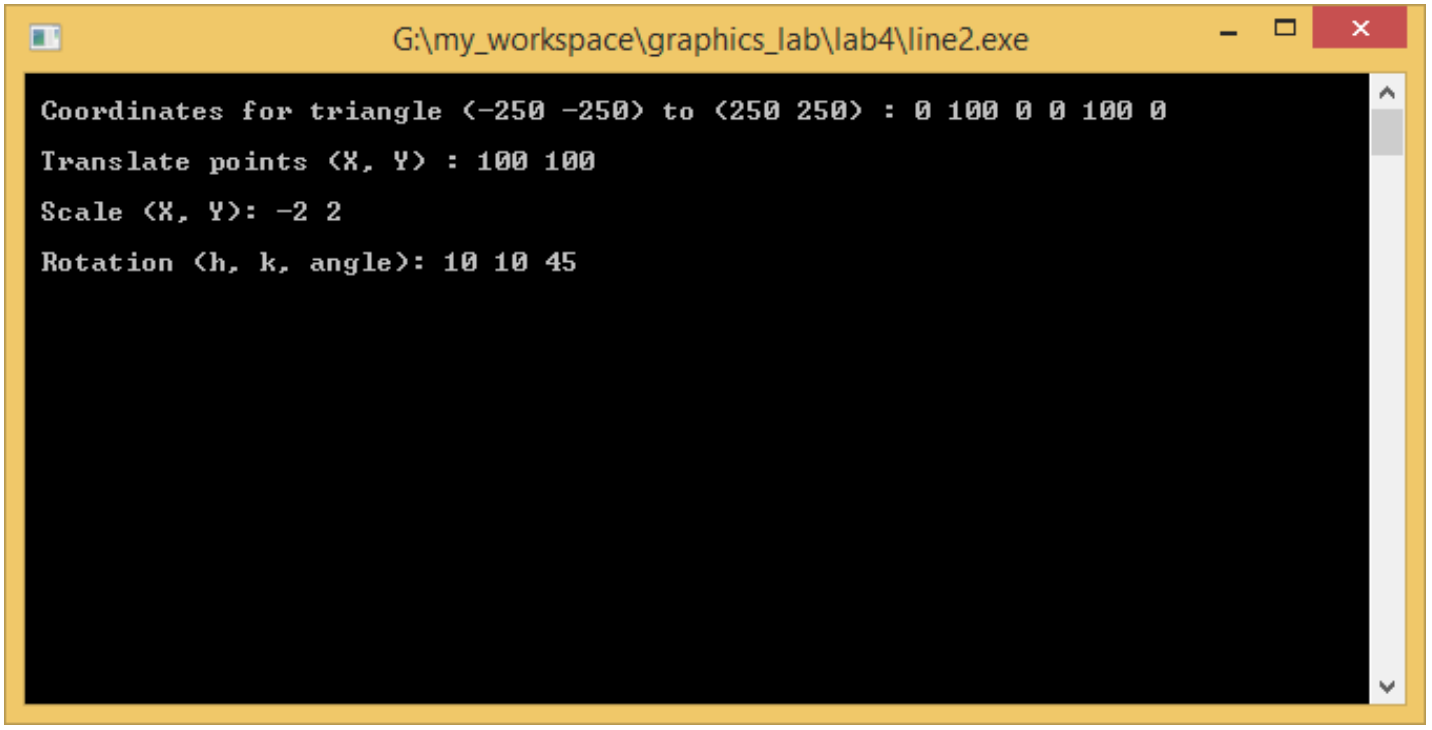
```
#include<graphics.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>

void joinPoints(int x0, int y0,  int x1,int y1, int color){
    int x, y, dx, dy, e;
    dx = x1 - x0; dy = y1 - y0;
    int tmp;
    if (abs(dy) <= abs(dx))
    {
        if (x1 < x0)
        {
            tmp = x0; x0 = x1; x1 = tmp;
            tmp = y0; y0 = y1; y1 = tmp;
        }

        dx = x1 - x0; dy = y1 - y0;
        if (y0<=y1)
        {
            e = -dx;
            x = x0; y = y0;
            for (int i = 0; i <= dx; i++)
            {
                putpixel(x, y, color);
                x++; e = e + 2 * dy;
                if (e >= 0)
                {
                    y++;
                    e = e - 2 * dx;
                }
            }
        }
        else
        {
            e = dx;
            x = x0; y = y0;
            for (int i = 0; i <= dx; i++)
            {
                putpixel(x, y, color);
                x++; e = e + 2 * dy;
                if (e <= 0)
                {
                    --y;
                    e = e + 2 * dx;
                }
```

```
47              }
48          }
49      }
50      else
51      {
52          if (y1 < y0)
53          {
54              tmp = x0; x0 = x1; x1 = tmp;
55              tmp = y0; y0 = y1; y1 = tmp;
56          }
57
58          dx = x1 - x0; dy = y1 - y0;
59          if (x0 <= x1)
60          {
61              e = -dy;
62              x = x0; y = y0;
63              for (int i = 0; i <= dy; i++)
64              {
65                  putpixel(x, y, color);
66                  y++; e = e + 2 * dx;
67                  if (e >= 0)
68                  {
69                      x++;
70                      e = e - 2 * dy;
71                  }
72              }
73          }
74          else
75          {
76              e = dy;
77              x = x0; y = y0;
78              for (int i = 0; i <= dy; i++)
79              {
80                  putpixel(x, y, color);
81                  y++; e = e + 2 * dx;
82                  if (e <= 0)
83                  {
84                      --x;
85                      e = e + 2 * dy;
86                  }
87              }
88          }
89      }
90  }
91
92  void plotLine(int x0, int y0,  int x1,int y1, int color){
93      int cx = getmaxx()/2;
94      int cy = getmaxy()/2;
95
96      x0 = cx+x0;
97      x1 = cx+x1;
98
99      y0 = cy-y0;
100     y1 = cy-y1;
101
102     joinPoints(x0, y0, x1, y1, color);
103 }
104
105 void drawAxis(){
106     int x = getmaxx();
107     int y = getmaxy();
108     joinPoints(0, y/2, x, y/2, WHITE);
```

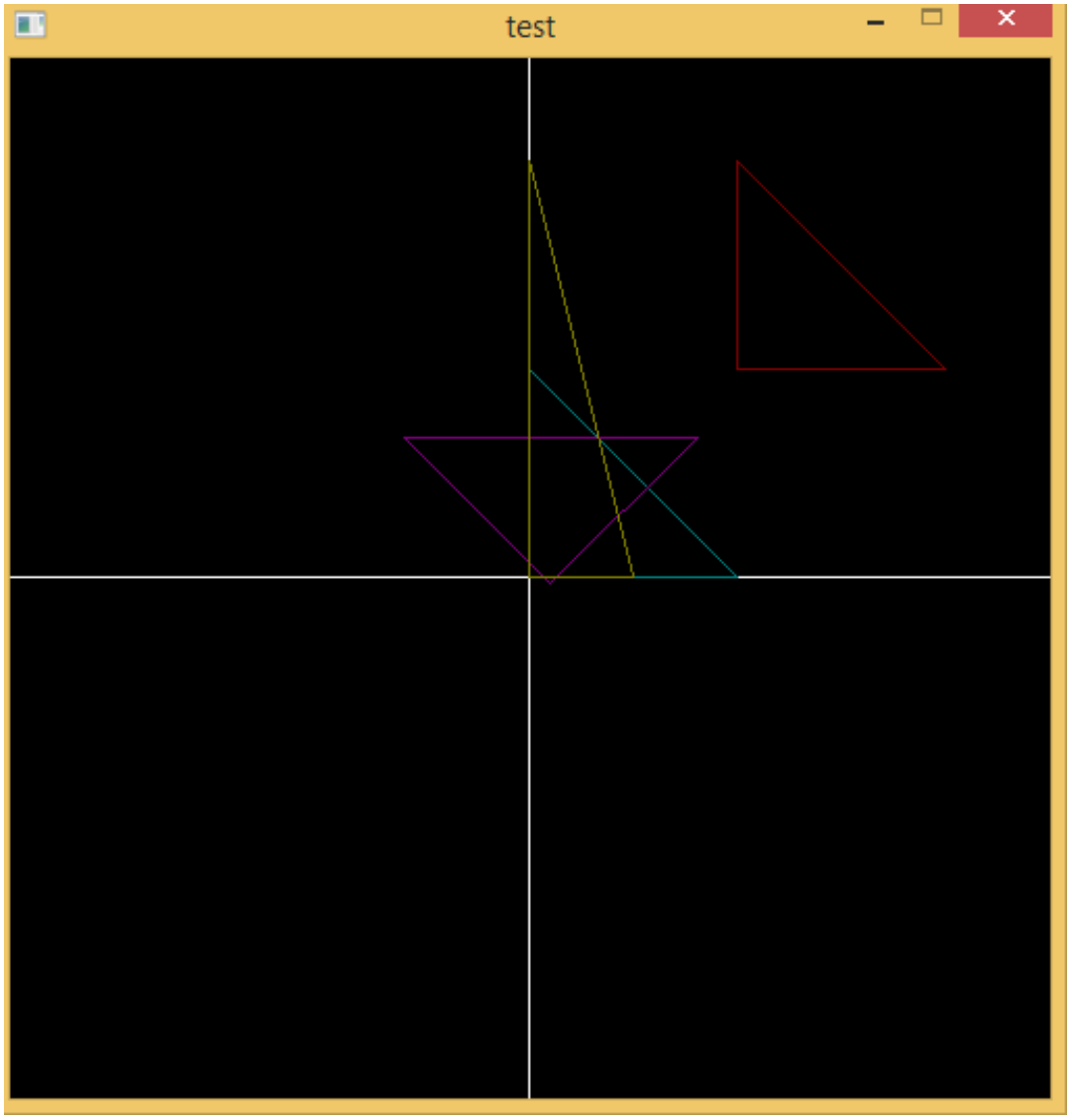
```
109     joinPoints(x/2, 0, x/2, y, WHITE);
110 }
111
112 void plotTriangle(int x0, int y0, int x1, int y1, int x2, int y2, int color){
113     plotLine(x0, y0, x1, y1, color);
114     plotLine(x0, y0, x2, y2, color);
115     plotLine(x1, y1, x2, y2, color);
116 }
117
118 int getXPrime(int x, int y, int h, int k, int t){
119     double tmp1 = cos((t*3.1416)/180);
120     double tmp2 = sin((t*3.1416)/180);
121     double xP = (tmp1*(x-h))+(tmp2*(k-y))+h+.5;
122     return (int)xP;
123 }
124
125 int getYPrime(int x, int y, int h, int k, int t){
126     double tmp1 = cos((t*3.1416)/180);
127     double tmp2 = sin((t*3.1416)/180);
128     double yP = (tmp2*(x-h))+(tmp1*(y-k))+k+.5;
129     return (int)yP;
130 }
131
132 int getXscaled(int x, int sx){
133     float tmp;
134     if(sx>=0)return x*sx;
135     tmp = x/abs(sx);
136     return (int)(tmp+.5);
137 }
138
139 int getYscaled(int y, int sy){
140     float tmp;
141     if(sy>=0)return y*sy;
142     tmp = y/abs(sy);
143     return (int)(tmp+.5);
144 }
145
146 int main()
147 {
148     int winW = 500;
149     int winH = 500;
150
151     int x1, y1,x2, y2,x3, y3,tx, ty,sx, sy,h, k, ang;
152
153     printf("\n Coordinates for triangle (%d %d) to (%d %d) : ",
154            -winW/2, -winH/2, winW/2, winH/2);
155     scanf("%d %d %d %d %d %d", &x1, &y1, &x2, &y2, &x3, &y3);
156
157     printf("\n Translate points (X, Y) : ");
158     scanf("%d %d", &tx, &ty);
159
160     printf("\n Scale (X, Y): ");
161     scanf("%d %d", &sx, &sy);
162
163     printf("\n Rotation (h, k, angle): ");
164     scanf("%d %d %d", &h, &k, &ang);
165
166     initwindow(winW,winH,"test");
167     drawAxis();
168
169     plotTriangle(x1, y1, x2, y2, x3, y3, 3);
170
```

```
171     // translate
172     plotTriangle(x1+tx, y1+ty, x2+tx, y2+ty, x3+tx, y3+ty, 4);
173
174     // rotate
175     plotTriangle(getXPrime(x1, y1, h, k, ang), getYPrime(x1, y1, h, k, ang),
176                  getXPrime(x2, y2, h, k, ang), getYPrime(x2, y2, h, k, ang),
177                  getXPrime(x3, y3, h, k, ang), getYPrime(x3, y3, h, k, ang), 5);
178
179     // scaling
180     plotTriangle(getXscaled(x1, sx), getYscaled(y1, sy),
181                  getXscaled(x2, sx), getYscaled(y2, sy),
182                  getXscaled(x3, sx), getYscaled(y3, sy), 6);
183
184     getch();
185     closegraph();
186     return 0;
187 }
```

Input



Output


test