/\* 3 Write C++ program to draw a concave polygon and fill it with desired color using scan fill algorithm. Apply the concept of inheritance. \*/

#include<iostream>

#include<graphics.h>

using namespace std;

int main()

{

int gd=DETECT, gm;

initgraph(&gd,&gm,NULL);

int cnt,temp,i,j,n, ymax, ymin, yscan;

float inter\_x[10], m[10],dx,dy;

int x[10],y[10];

cout<<"Enter no. of vertices of polygon: ";

cin>>n;

for(i=0;i<n;i++)

{

cout<<" Enter " << i+1 << " vertices of polygon (as x and y) is: ";

cin>>x[i]>>y[i];

}

x[n]=x[0];

y[n]=y[0];

for(i=0;i<n;i++) //draw a polygon

{

line(x[i],y[i],x[i+1],y[i+1]);

}

ymax=0;

ymin=480;

for(i=0;i<n;i++) //find ymax and ymin

{

if(y[i]>ymax)

ymax=y[i];

if(y[i]<ymin)

ymin=y[i];

}

for(i=0;i<n;i++) //cal slope of each side

{

dx=x[i+1]-x[i];

dy=y[i+1]-y[i];

if(dx==0)

m[i]=0;

else if(dy==0)

m[i]=0;

else

m[i]=(float)dy/dx;

}

for(yscan=ymax;yscan>ymin;yscan--)

{

cnt=0;

for(i=0;i<n;i++)

{

if(y[i]>yscan && y[i+1]<=yscan || y[i]<=yscan && y[i+1]>yscan)

{

if(m[i]==0)

inter\_x[cnt]=x[i]; //inactive edge

else

inter\_x[cnt]= x[i] + (yscan-y[i])/m[i]; //active edge

cnt++;

}

}

for(j=0;j<cnt-1;j++) //find intersection points

{

if(inter\_x[j]<=inter\_x[j+1])

{

temp=inter\_x[j];

inter\_x[j]=inter\_x[j+1];

inter\_x[j+1]=temp;

}

}

for(j=0;j<cnt-1;j=j+2) //fill the polygon using line

{

line(inter\_x[j],yscan,inter\_x[j+1],yscan);

delay(100);

}

} //end of for loop

getch();

return 0;

}

/\*

avanish@avanish-Lenovo-G580:~/Desktop/CG\_PRACT\_CRB$ g++ 1scanline.cpp -lgraph

(base) avanish@avanish-Lenovo-G580:~/Desktop/CG\_PRACT\_CRB$ ./a.out

Enter no. of vertices of polygon: 5

Enter 1 vertices of polygon (as x and y ) is: 400 400

Enter 2 vertices of polygon (as x and y ) is: 400 100

Enter 3 vertices of polygon (as x and y ) is: 200 300

Enter 4 vertices of polygon (as x and y ) is: 100 200

Enter 5 vertices of polygon (as x and y ) is: 100 400

\*/