NAME -Ashutosh Kumar

Roll - 18

4. Write a program to do the following:

a. Ask the user to enter the co-ordinates of two polygons (The co-ordinates should be

chosen in such a way that the polygons overlap with each other)

b. Fill the overlapping area.

CODE -

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

const int MAX\_POINTS = 20;

// Function to find x intersection points.

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\*y4 - y3\*x4);

int den = (x1-x2) \* (y3-y4) - (y1-y2) \* (x3-x4);

return num/den;

}

// Function to find y-intersection points.

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) \* (y3-y4) - (y1-y2) \* (x3-x4);

return num/den;

}

// Clipping for

void clip(int poly\_points[][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][1] = y\_intersect(x1,y1, x2, y2, ix, iy, kx, ky);

new\_poly\_size++;

}

else

{

//No points are added

}

}

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 printPoly(int arr[]) {

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

setfillstyle(XHATCH\_FILL, RED);

fillpoly(4, arr);

getch();

closegraph();

//return 0;

}

void suthHodgClip(int poly\_points[][2], int poly\_size, int clipper\_points[][2], int clipper\_size)

{

int gd = DETECT, gm;

int arr[100];

int count=0;

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]);

}

for (int i=0,j=0; i < poly\_size; i++) {

arr[j++]=poly\_points[i][0];

arr[j++]=poly\_points[i][1];

count++;

}

printPoly(arr);

//setcolor(getmaxcolor());

//setfillstyle(SOLID\_FILL, RED);

//fillpoly(5, arr);

//getch();

//closegraph();

//initgraph(&gd, &gm, "");

//setfillstyle(XHATCH\_FILL, RED);

//fillpoly(count, arr);

//closegraph();

//for (int i=0; i < 2\*count; i++){

// printf("%d",arr[i]);

//}

//printf("\n");

//for (int i=0; i < poly\_size; i++)

//printf("%d,%d",poly\_points[i][0], poly\_points[i][1]);

}

int main()

{

int poly\_size = 3;

int poly\_points[20][2] = {{100,150}, {200,250}, {300,200}};

int clipper\_size = 4;

int clipper\_points[][2] = {{150,150}, {150,200}, {200,200}, {200,150} };

suthHodgClip(poly\_points, poly\_size, clipper\_points, clipper\_size);

return 0;

}