**Lab Report:1**

**Title:**

*Course title: Computer Graphics*

*Course code: CSE-304*

*3rd year 1st Semester*

**Date of Submission**: 28.05.2023

****

**Submitted to-**

Dr. Md. Shorif Uddin

***Professor***

Dr. Morium Akter

***Assistant Professor***

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

| **Sl** | Class Roll | Exam Roll | Name |
| --- | --- | --- | --- |
| 01 | 2147 | 191323 | Shabrina Akter Shahana |

**1.Source Code of Scan Conversion of Point:**

#include <iostream>

int main()

{

int x = 3;

int y = 2;

int width = 10;

int height = 5;

char pixelBuffer[5][10];

for (int i = 0; i < height; ++i) {

for (int j = 0; j < width; ++j) {

pixelBuffer[i][j] = ' ';

}

}

if (x >= 0 && x < width && y >= 0 && y < height) {

pixelBuffer[y][x] = '\*';

}

for (int i = 0; i < height; ++i) {

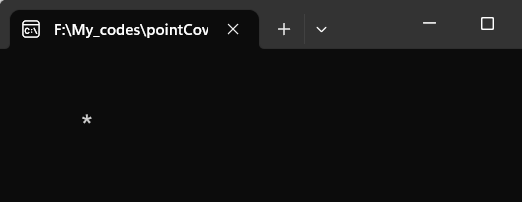
for (int j = 0; j < width; ++j) {

std::cout << pixelBuffer[i][j] << ' ';}

std::cout << std::endl;}

return 0;}

Output:



**2.Source Code for converting a line using DDA algorithm:**

#include <stdio.h>

#include <stdlib.h>

#include <graphics.h>

#include<math.h>

int main(){

int gdriver=DETECT,gmode;

int x1,y1,x2,y2,i,step,xn,yn,dx,dy;

initwindow(500,500,"Line");

printf("Enter the starting coordinate");

scanf("%d%d",&x1,&y1);

printf("Enter the end coordinate");

scanf("%d%d",&x2,&y2);

dx=x2-x1;

dy=y2-y1;

if(abs(dx)>abs(dy)){

step=abs(dx);}

else{

step=abs(dy);}

xn=dx/step;

yn=dy/step;

for(i=1;i<=step;i++){

putpixel(x1,y1,RED);

delay(100);

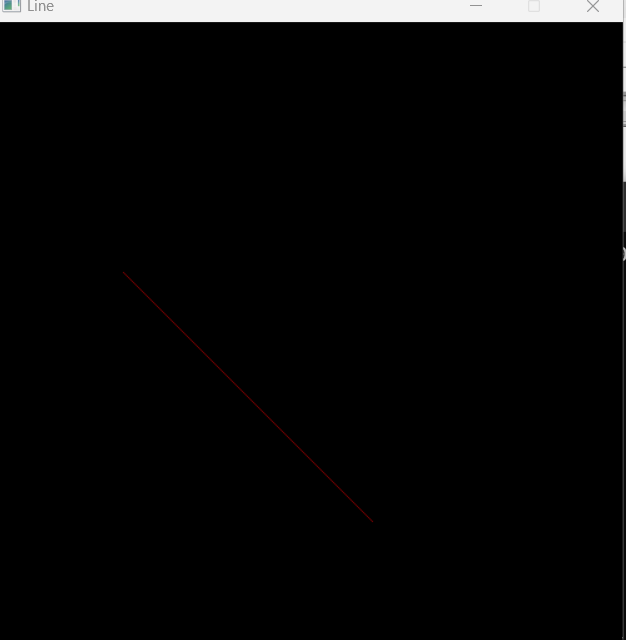
x1=x1+xn;

y1=y1+yn;}

getch();

closegraph();

**Output:**



**3.Source Code for Conversion of line using Bresenham Algoritm:**

#include <stdio.h>

#include <stdlib.h>

#include <graphics.h>

void drawLine(int x0, int y0, int x1, int y1) {

int dx = abs(x1 - x0);

int dy = abs(y1 - y0);

int sx, sy;

if (x0 < x1) {

sx = 1;

} else {

sx = -1;

}

if (y0 < y1) {

sy = 1;

} else {

sy = -1;

}

int err = dx - dy;

while (x0 != x1 || y0 != y1) {

putpixel(x0, y0, WHITE);

int e2 = 2 \* err;

if (e2 > -dy) {

err = err - dy;

x0 = x0 + sx;

}

if (e2 < dx) {

err = err + dx;

y0 = y0 + sy;

}

}

}

int main() {

int gd = DETECT, gm;

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

int x0,y0,x1,y1;

printf("Enter the starting coordinate");

scanf("%d%d",&x0,&y0);

printf("Enter the end coordinate");

scanf("%d%d",&x1,&y1);

drawLine(x0, y0, x1, y1);

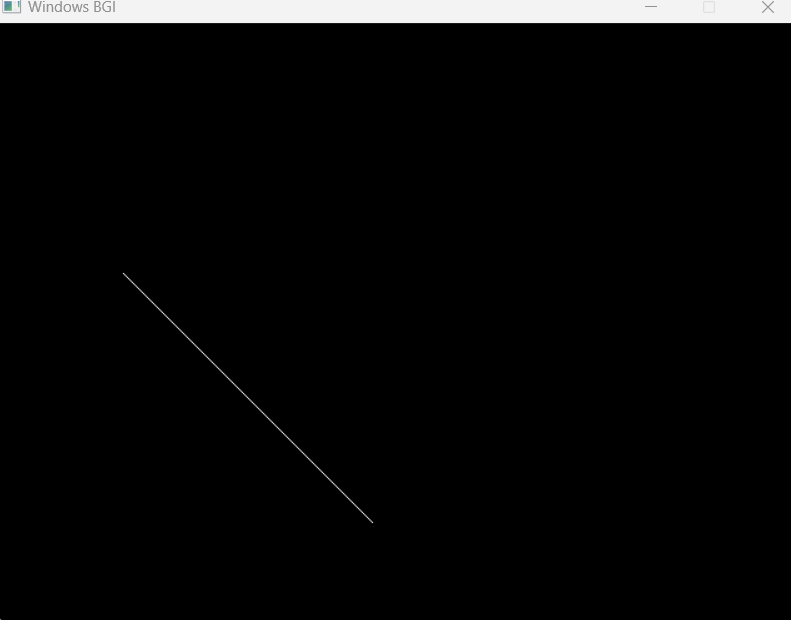
delay(508800);

closegraph();

return 0;

}

**Output:**



**4.Source Code for circle conversion using Bresenham Circle Algorithm:**

#include <graphics.h>

#include <stdio.h>

void drawCircle(int xc, int yc, int radius) {

int x = 0;

int y = radius;

int d = 3 - 2 \* radius;

while (x <= y) {

// Plot the eight points

putpixel(xc + x, yc + y, WHITE);

putpixel(xc - x, yc + y, WHITE);

putpixel(xc + x, yc - y, WHITE);

putpixel(xc - x, yc - y, WHITE);

putpixel(xc + y, yc + x, WHITE);

putpixel(xc - y, yc + x, WHITE);

putpixel(xc + y, yc - x, WHITE);

putpixel(xc - y, yc - x, WHITE);

if (d < 0) {

d += 4 \* x + 6;

} else {

d += 4 \* (x - y) + 10;

y--;

}

x++;

}

}

int main() {

int gd = DETECT, gm;

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

int xc, yc, radius;

printf("Enter the center coordinates (xc, yc): ");

scanf("%d %d", &xc, &yc);

printf("Enter the radius: ");

scanf("%d", &radius);

drawCircle(xc, yc, radius);

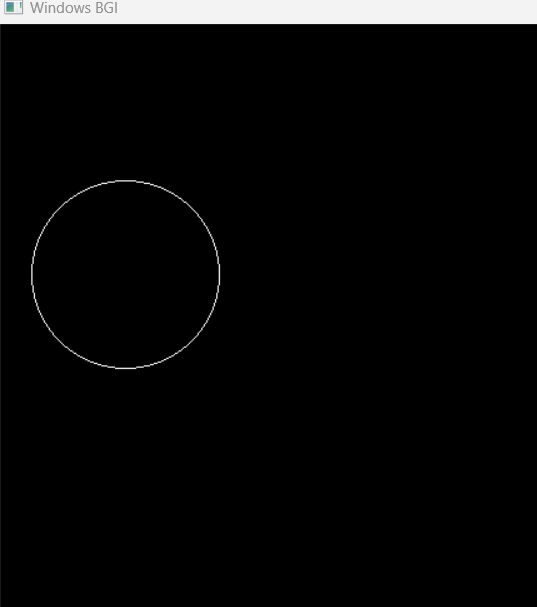
delay(5088800); // Delay to show the circle

closegraph();

return 0;

}

**Output:**

****