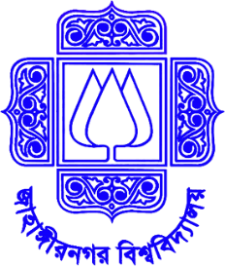
**Lab Report. 01**

*Course title: Computer Graphics Lab*

*Course code: CSE-304*

*3rd Year 1st Semester 2022*

**Date of Submission**: 28/05/2023



###### 

###### **Submitted to-**

###### **Dr. Mohammad Shorif Uddin** Professor &

***Dr. Morium Akter*** Associate Professor

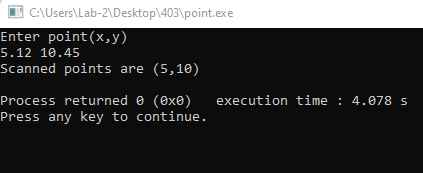
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl** | Class Roll | Exam Roll | Name |
| 01 | 403 |  | Raian Rashid |

Scan Conversion of a Point:

Code:

|  |  |
| --- | --- |
| #include<bits/stdc++.h>  using namespace std;  int main()  {  float x,y;  cout << "Enter point(x,y)" << endl; | cin >> x>>y;  int px,py;  px = floor(x);  py = floor(y);  cout << "Scanned points are ("<<px<<","<<py<<")"<<endl;  } |

Output:

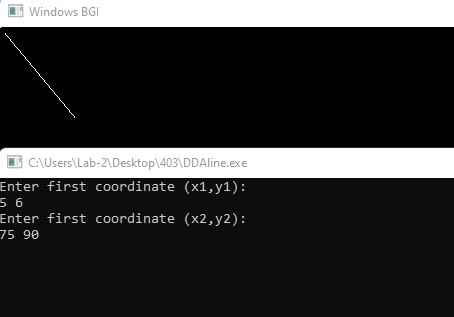


Scan Conversion of a Line using DDA Algorithm:

Code:

|  |  |
| --- | --- |
| #include<bits/stdc++.h>  #include<graphics.h>  using namespace std;  void drawlinedda(int x1, int y1, int x2, int y2)  {  int dx = x2 - x1;  int dy = y2 - y1;  int steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);  float xincrement = static\_cast<float>(dx) / steps;  float yincrement = static\_cast<float>(dy) / steps;  float x = x1;  float y = y1;  for(int i=0; i<=steps; i++)  {  putpixel(static\_cast<int>(x),static\_cast<int>(y),WHITE);  x += xincrement; | y += yincrement;  }  }  int main()  {  int gd = DETECT, gm;  initgraph(&gd,&gm, "");  int x1,y1,x2,y2;  cout << "Enter first coordinate (x1,y1):" << endl;  cin >> x1 >> y1;  cout << "Enter first coordinate (x2,y2):" << endl;  cin >> x2>>y2;  drawlinedda(x1,y1,x2,y2);  delay(50000000);  closegraph();  return 0;  } |

Output:

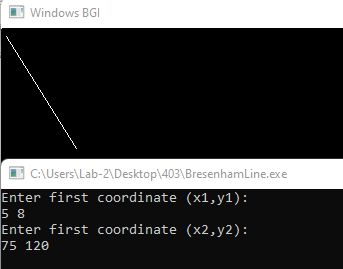


Scan Conversion of a Line using Bresenham Algorithm:

Code:

|  |  |
| --- | --- |
| #include<bits/stdc++.h>  #include<graphics.h>  using namespace std;  void drawlinebresenham(int x1, int y1, int x2, int y2)  {  int dx = abs(x2 - x1);  int dy = abs(y2- y1);  int sx = (x1 < x2)? 1 : -1;  int sy = (y1 < y2)? 1 : -1;  int err = dx - dy;  while(true)  {  putpixel(x1,y1,WHITE);  if(x1==x2 && y1==y2)  break;  int dt = 2\*err;  if(dt>-dy)  {  err -= dy;  x1 += sx;  }  if(dt < dx) | {  err += dx;  y1 += sy;  }  }  }  int main()  {  int gd = DETECT, gm;  initgraph(&gd,&gm, "");  int x1,y1,x2,y2;  cout << "Enter first coordinate (x1,y1):" << endl;  cin >> x1 >> y1;  cout << "Enter first coordinate (x2,y2):" << endl;  cin >> x2>>y2;  drawlinebresenham(x1,y1,x2,y2);  delay(50000000);  closegraph();  return 0;  } |

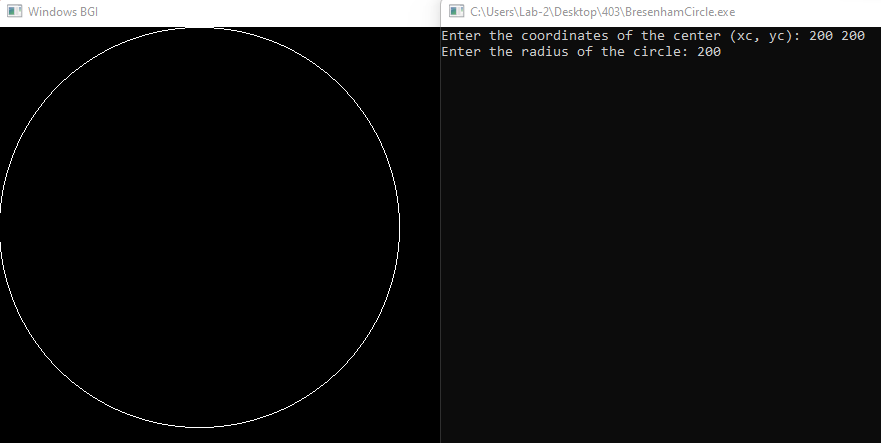
Output:



Scan Conversion of Circle using Bresenham Algorithm:

Code:

|  |  |
| --- | --- |
| #include <iostream>  #include <cmath>  #include <graphics.h>  using namespace std;  void drawCircleBresenham(int xc, int yc, int radius) {  int x = 0;  int y = radius;  int d = 3 - 2 \* radius;  while (x <= y) {  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;  cout << "Enter the coordinates of the center (xc, yc): ";  cin >> xc >> yc;  cout << "Enter the radius of the circle: ";  cin >> radius;  drawCircleBresenham(xc, yc, radius);  delay(5000);  closegraph();  return 0;  } |

Output: