JADAVPUR UNIVERSITY

GRAPHICS ASSIGNMENT SUBMISSION : REPORT 01

SUBMITTED BY :

NAME : PRASENJIT BISWAS

CLASS : BCSE 3rd YEAR

ROLL NO:001310501015

1)DDA Line Drawing Algorithm:

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#include<dos.h>

int sign(int p){

if(p<0){

return -1;

}

else if(p==0){

return 0;

}

else{

return 1;

}

}

int main(void){

int x1, x2, y1, y2; //taking starting and ending point of the line

printf("Enter x1:");

scanf("%d", &x1);

printf("Enter x2:");

scanf("%d", &x2);

printf("Enter y1:");

scanf("%d", &y1);

printf("Enter y2:");

scanf("%d", &y2);

initwindow(700,740);

int xmax = getmaxx();

int ymax = getmaxy();

int i; //Creating Graph

for(i=0;i<=xmax;i+=10){

line(i,0,i,ymax);

}

for(i=0;i<=ymax;i+=10){

line(o,i,xmax,i);

}

setlinestyle(0,0,2);

line(xmax/2,0,xmax/2,ymax);

line(0,ymax/2,xmax,ymax/2);

int d1 = (x2-x1)\*10; //implementing the algo.

int d2 = (y2-y1)\*10;

x1 = xmax/2;

y1 = ymax/2;

//line(x1,y1,x2,y2);

double len = (abs(x2-x1)>abs(y2-y1))?abs(x2-x1):abs(y2-y1);

double dx, dy;

dx = ((x2-x1)/len)\*10;

dy = ((y2-y1)/len)\*10;

double x = x1+(5\*sign(x2-x1));

double y = y1+(5\*sign(y2-y1));

for(i=10;i<=len;i+=10){

floodfill(x,y,WHITE);

x = x+dx;

y = y+dy;

}

setcolor(CYAN); //drawing final line

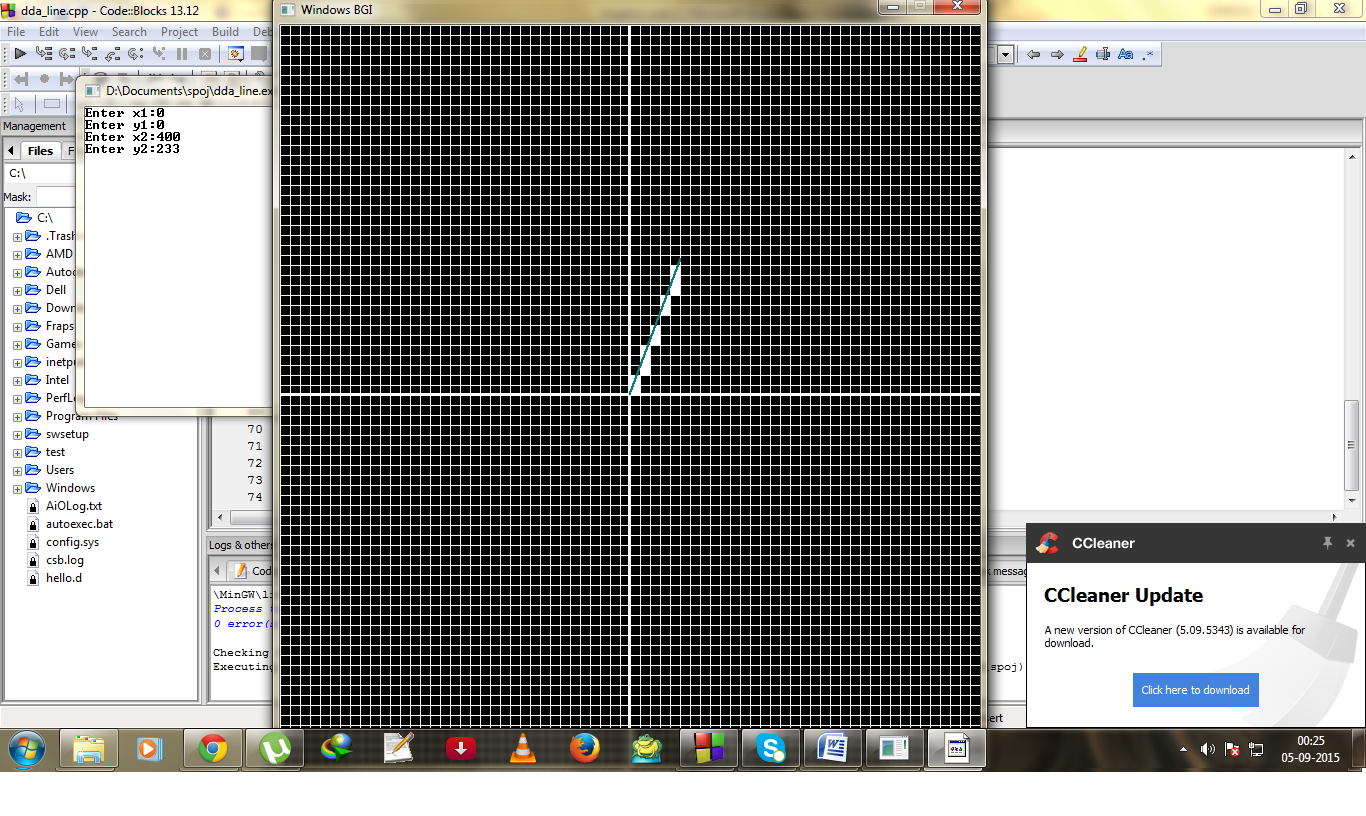
line(x1,y1,x2,y2);

getch();

closegraph();

return 0;

}



1)Moving Car Algorithm:

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<dos.h>

#include<graphics.h>

int main()

{

int i=0,m=0;

initwindow(700,700);

while(!kbhit())

{

cleardevice();

i++;

if(i>getmaxx())

i-=670;

line(80+i,300,90+i,270);

line(90+i,270,120+i,270);

line(120+i,270,160+i,240);

line(160+i,240,230+i,240);

line(230+i,240,275+i,270);

line(275+i,270,310+i,270);

line(310+i,270,335+i,290);

line(335+i,290,335+i,300);

line(255+i,300,335+i,300);

line(180+i,300,210+i,300);

line(80+i,300,135+i,300);

arc(232+i,300,0,180,23);

arc(157+i,300,0,180,23);

circle(232+i,300,18);

circle(157+i,300,18);

pieslice(232+i,300,0+m,90+m,18);

pieslice(232+i,300,180+m,270+m,18);

pieslice(157+i,300,0+m,90+m,18);

pieslice(157+i,300,180+m,270+m,18);

if(m<-360)

m=90;

m-=3;

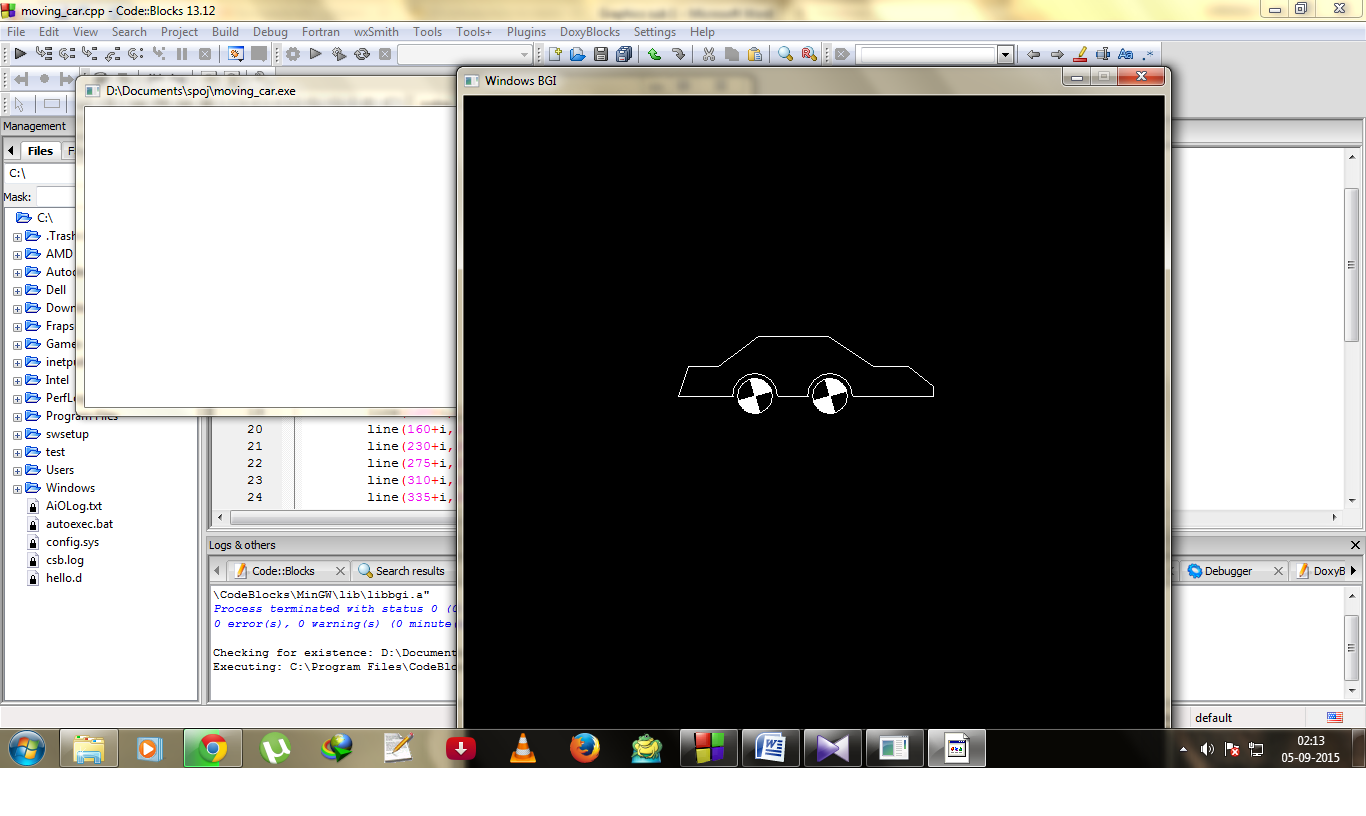
delay(5);

}

getch();

closegraph();

}



1. Bresenham Line Algorithm

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#include<dos.h>

int main(void){

int x1, y1, x2, y2; //taking initial values

printf("Enter x1 and y1:");

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

printf("Enter x2 and y2:");

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

double dx = x2-x1;

double dy = y2-y1;

double m = dy/dx;

double e = m-0.5;

initwindow(700,700); //initializing graphics window

int xmax = getmaxx();

int ymax = getmaxy();

int i;

for(i=0; i<=xmax; i+=10){ //drawing graph

line(i,0,i,ymax);

}

for(i=0; i<=ymax; i+=10){

line(0,i,xmax,i);

}

setlinestyle(0,0,2);

line(xmax/2,0,xmax/2,ymax);

line(0,ymax/2,xmax,ymax/2);

int dx2 = dx;

dx = (x2-x1)\*10;

dy = (y2-y1)\*10;

x1 = xmax/2; //implementing Bresen. Algo.

y1 = ymax/2;

x2 = x1+dx;

y2 = y1-dy;

double x = x1+5;

double y = y1-5;

for(i=1;i<=dx2;i++){

floodfill(x,y,WHITE);

while(e>0){

y=y-10;

e=e-1;

}

x=x+10;

e=e+m;

}

setcolor(CYAN); //drawing final line

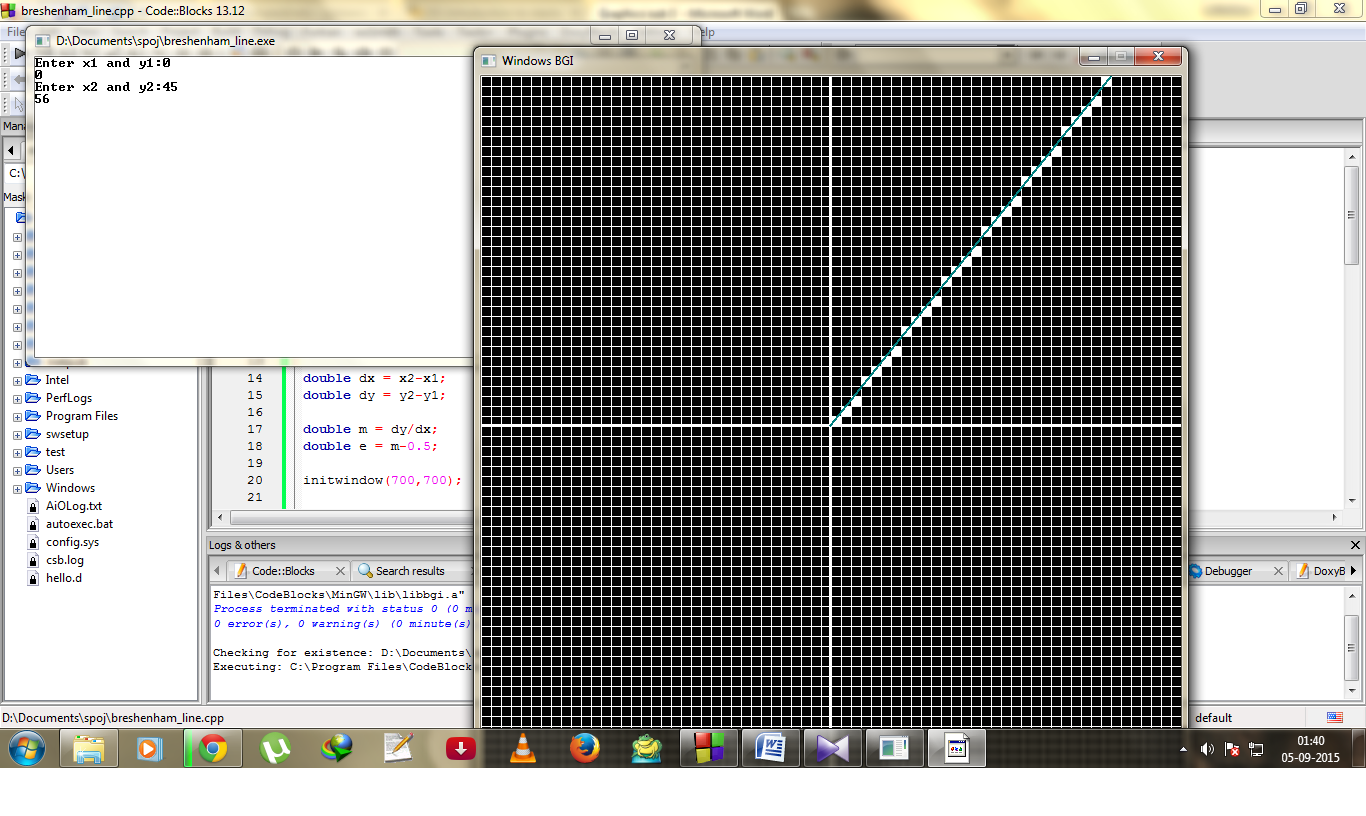
line(x1,y1,x2,y2);

getch();

closegraph();

return 0;

}



1. Bresenham Circle Algorithm:

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#include<dos.h>

void mh(int\*xi,int\*yi,int\*x\_c,int\*y\_c,int\*del){

\*xi=\*xi+1;

\*del=\*del+(2\*(\*xi))+1;

\*x\_c=\*x\_c+10;

}

void md(int\*xi,\*yi,int\*x\_c,int\*y\_c,int\*del){

\*xi=\*xi+1;

\*yi=\*yi-1;

\*del=\*del+(2\*(\*yi))+2;

\*x\_c=\*x\_c+10;

\*y\_c=\*y\_c+10;

}

void mv(int\*xi,\*yi,int\*x\_c,int\*y\_c,int\*del){

\*yi=\*yi-1;

\*del=\*del-(2\*(\*yi))+1;

\*y\_c=\*y\_c+10;

}

int main(void){

int R;

printf("Enter radius of circle:");

scanf("%d",&R);

int gd=DETECT, gm;

initgraph(&gd,&gm,"C:\\TurboC3\\BGI");

int xmax=getmaxx();

int ymax=getmaxy();

int i;

for(i=0;i<=xmax;i+=10){

line(i,0,i,ymax);

}

for(i=0;i<=ymax;i+=10){

line(0,i,xmax,i);

}

setlinestyle(0,0,2);

line(xmax/2,0,xmax/2,ymax);

line(0,ymax/2,xmax,ymax/2);

int xi=0;

int x\_chosen = xmax/2+5;

int yi=0;

int y\_chosen = ymax/2-(R\*10)-5;

int del\_i=2\*(1-R);

int lim=0;

while(yi>=lim){

floodfill(x\_chosen,y\_chosen,WHITE);

int y\_chosen2 = y\_chosen+(2\*(ymax/2 - y\_chosen));

int x\_chosen2 = x\_chosen-(2\*(x\_chosen - xmax/2));

floodfill(x\_chosen,y\_chosen2,WHITE);

floodfill(x\_chosen2,y\_chosen,WHITE);

floodfill(x\_chosen2,y\_chosen2,WHITE);

if(del\_i<0){

int del = (2\*del\_i)+(2\*yi)-1;

if(del<=0){

mh(&xi,&yi,&x\_chosen,&y\_chosen,&del\_i);

}

else{

md(&xi,&yi,&x\_chosen,&y\_chosen,&del\_i);

}

}

else if(del\_i>0){

int del = (2\*del\_i)-(2\*xi)-1;

if(del<=0){

md(&xi,&yi,&x\_chosen,&y\_chosen,&del\_i);

}

else{

mv(&xi,&yi,&x\_chosen,&y\_chosen,&del\_i);

}

}

else if(del\_i==0){

md(&xi,&yi,&x\_chosen,&y\_chosen,&del\_i);

}

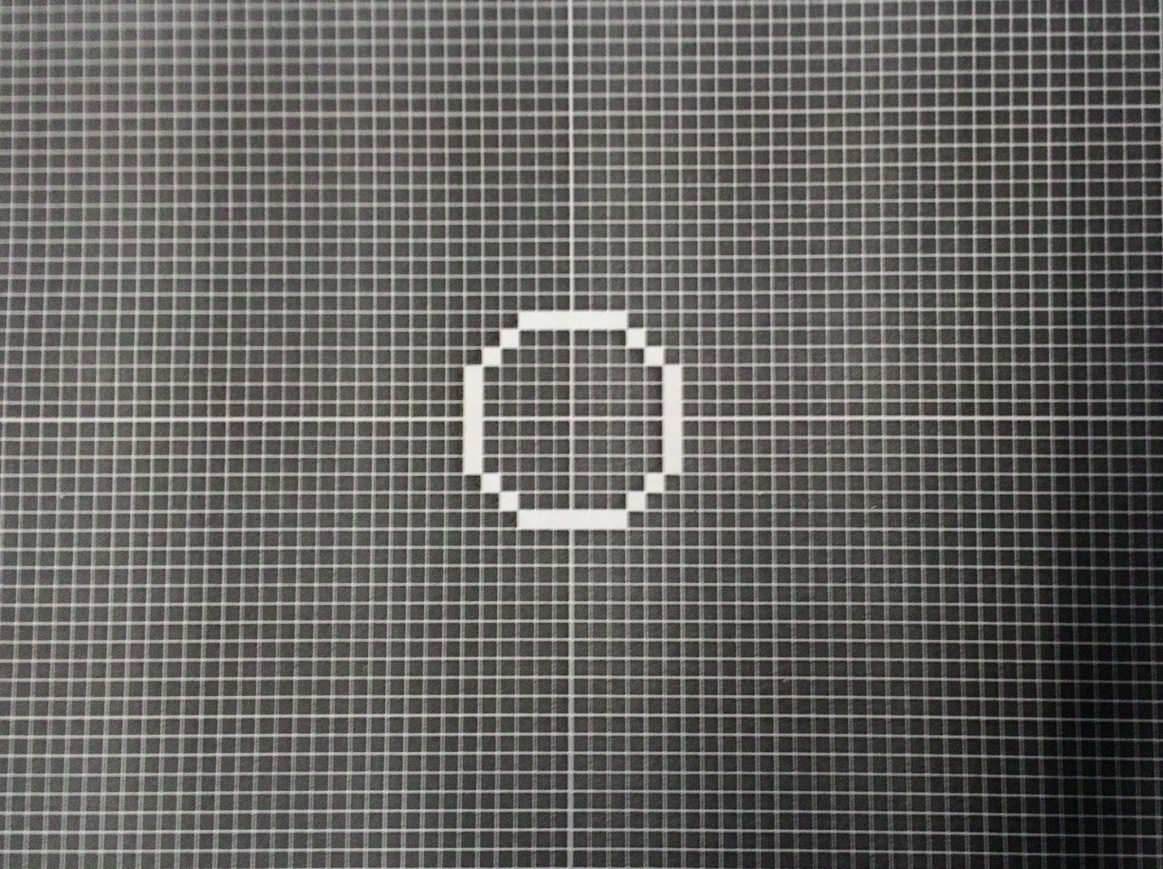
}

getch();

closegraph();

return 0;

}



1. Midpoint Line Algorithm

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#include<dos.h>

void plot(int x,int y){

int xmax=getmaxx();

int ymax=getmaxy();

int x0=xmax/2;

int y0=ymax/2;

int xpl=x0 + (x\*10);

int ypl=y0 - (y\*10);

xpl = xpl + 5;

ypl = ypl - 5;

floodfill(xpl,ypl,WHITE);

}

int main(void){

int x1, y1, x2, y2;

printf("Enter x1 and y1:");

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

printf("Enter x2 and y2:");

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

int dx = (x2-x1);

int dy = (y2-y1);

int d = (2\*dy)-dx;

int de = (2\*dy);

int dne = 2\*(dy-dx);

initwindow(700,700);

int xmax = getmaxx();

int ymax = getmaxy();

int i;

for(i=0;i<=xmax;i+=10){

line(i,0,i,ymax);

}

for(i=0;i<=ymax;i+=10){

line(0,i,xmax,i);

}

setlinestyle(0,0,2);

line(xmax/2,0,xmax/2,ymax);

line(0,ymax/2,xmax,ymax/2);

int x=x1;

int y=y1;

plot(x,y);

while(x<x2){

//plot(x,y);

if(d<=0){

//move east

d = d+de;

}

else{

y = y+1;

d = d+dne;

}

x=x+1;

plot(x,y);

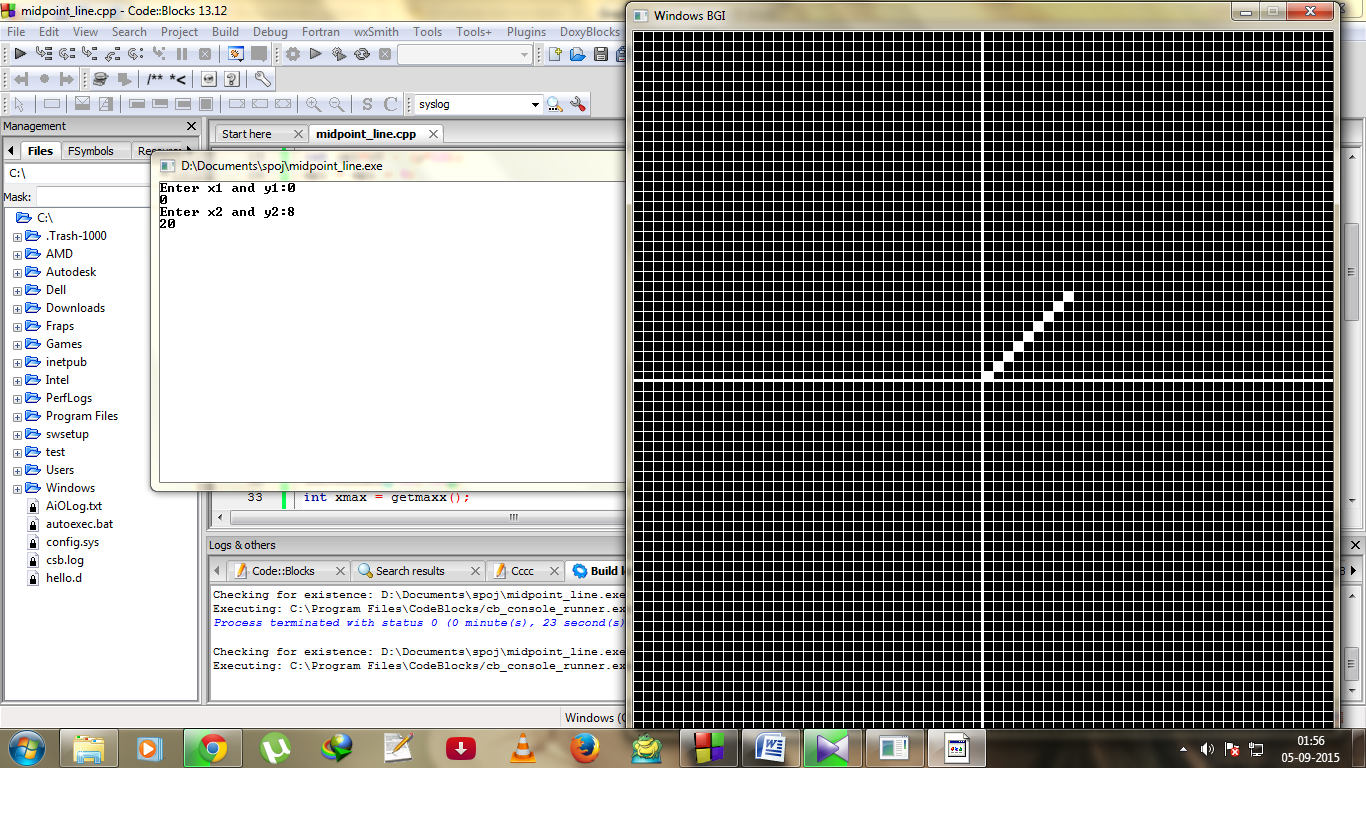
}

getch();

closegraph();

return 0;

}



1. Midpoint Circle Algorithm (Order 1):

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#include<dos.h>

void plot(int x,int y){

int xmax=getmaxx();

int ymax=getmaxy();

int x\_0=xmax/2;

int y\_0=ymax/2;

int x\_c=x\_0 + (x\*10);

int y\_c=y\_0 - (y\*10);

x\_c = x\_c + 5;

y\_c = y\_c - 5;

floodfill(x\_c,y\_c,WHITE);

}

int main(void){

int R;

printf("Enter the radius of the circle:");

scanf("%d", &R);

initwindow(700,700);

int xmax = getmaxx();

int ymax = getmaxy();

int i;

for(i=0;i<xmax;i+=10){

line(i,0,i,ymax);

}

for(i=0;i<=ymax;i+=10){

line(o,i,xmax,i);

}

setlinestyle(0,0,2);

//line(xmax/2,0,xmax/2,ymax);

//line(0,ymax/2,xmax,ymax/2);

int x=0;

int y=R;

int d=1-R;

plot(x,y);

plot(y,x);

plot(x,-y);

plot(y,-x);

plot(-x,y);

plot(-y,x);

plot(-x,-y);

plot(-y,-x);

while(x<y){

if(d<0){

d=d+(2\*x)+3;

}

else{

y=y-1;

d=d+2\*(x-y)+5;

}

x=x+1;

plot(x,y);

plot(y,x);

plot(x,-y);

plot(y,-x);

plot(-x,y);

plot(-y,x);

plot(-x,-y);

plot(-y,-x);

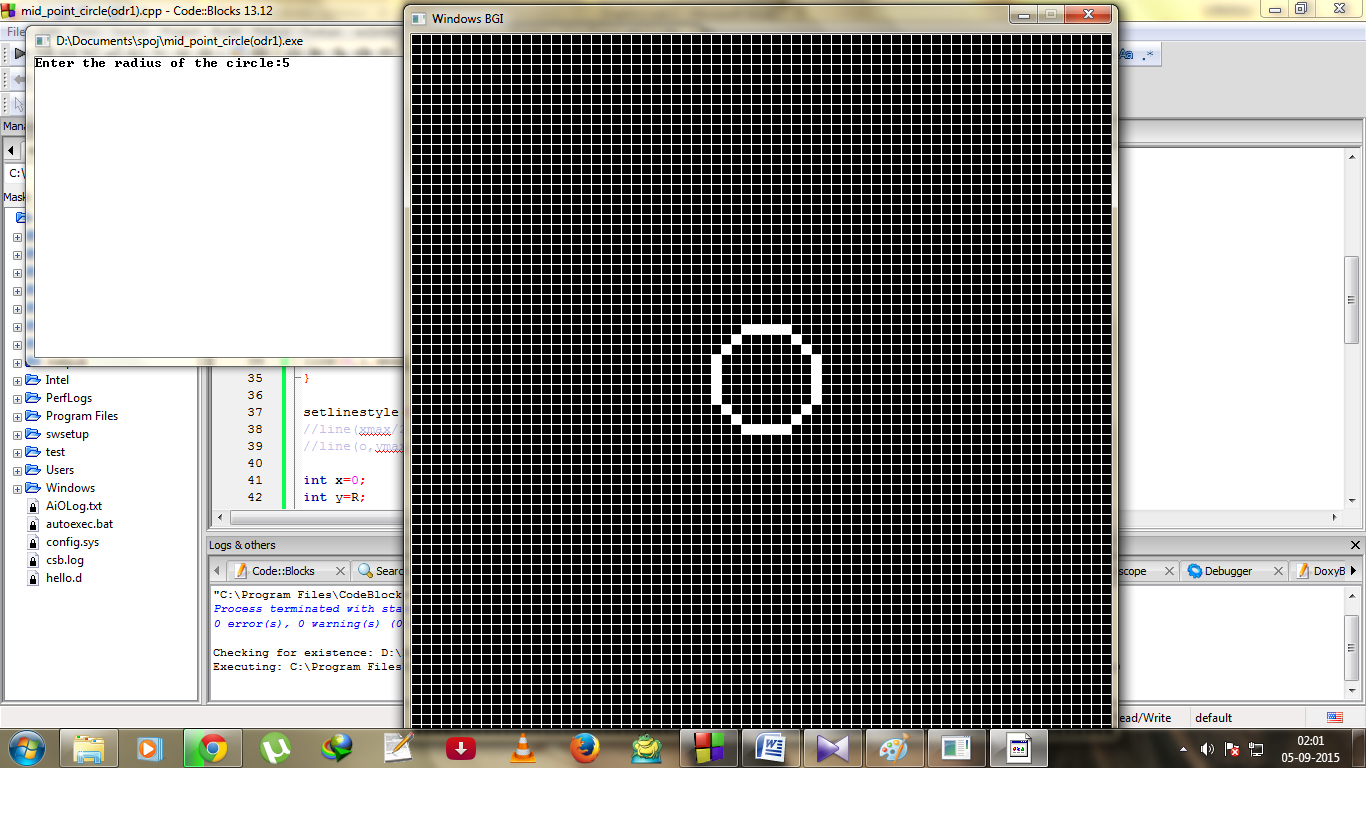
}

getch();

closegraph();

return 0;

}



1. Midpoint Circle Algorithm (Order 2):

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#include<dos.h>

void plot(int x,int y){

int xmax=getmaxx();

int ymax=getmaxy();

int x\_0 = xmax/2;

int y\_0 = ymax/2;

int x\_c = x\_0 + (x\*10);

int y\_c = y\_0 + (y\*10);

x\_c = x\_c + 5;

y\_c = y\_c - 5;

floodfill(x\_c,y\_c,WHITE);

}

int main(void){

int R;

printf("Enter the radius of the circle:");

scanf("%d",&R);

initwindow(700,700);

int xmax = getmaxx();

int ymax = getmaxy();

int i;

for(i=0; i<=xmax; i+=10){

line(i,0,i,ymax);

}

for(i=0; i<=ymax; i+=10){

line(0,i,xmax,i);

}

int x = 0;

int y = R;

int d = 1-R;

int de = 3;

int dse = -(2\*R)+5;

int ince\_de = 2;

int ince\_dse = 2;

int incse\_de = 2;

int incse\_dse = 4;

plot(x,y);

plot(y,x);

plot(-x,y);

plot(-y,x);

plot(x,-y);

plot(y,-x);

plot(-x,-y);

plot(-y,-x);

while(x<y){

if(d<0){

d = d + de;

de = de + ince\_de;

dse = dse + ince\_dse;

}

else{

d = d + dse;

y = y - 1;

de = de + incse\_de;

dse = dse + incse\_dse;

}

x = x + 1;

plot(x,y);

plot(y,x);

plot(-x,y);

plot(-y,x);

plot(x,-y);

plot(y,-x);

plot(-x,-y);

plot(-y,-x);

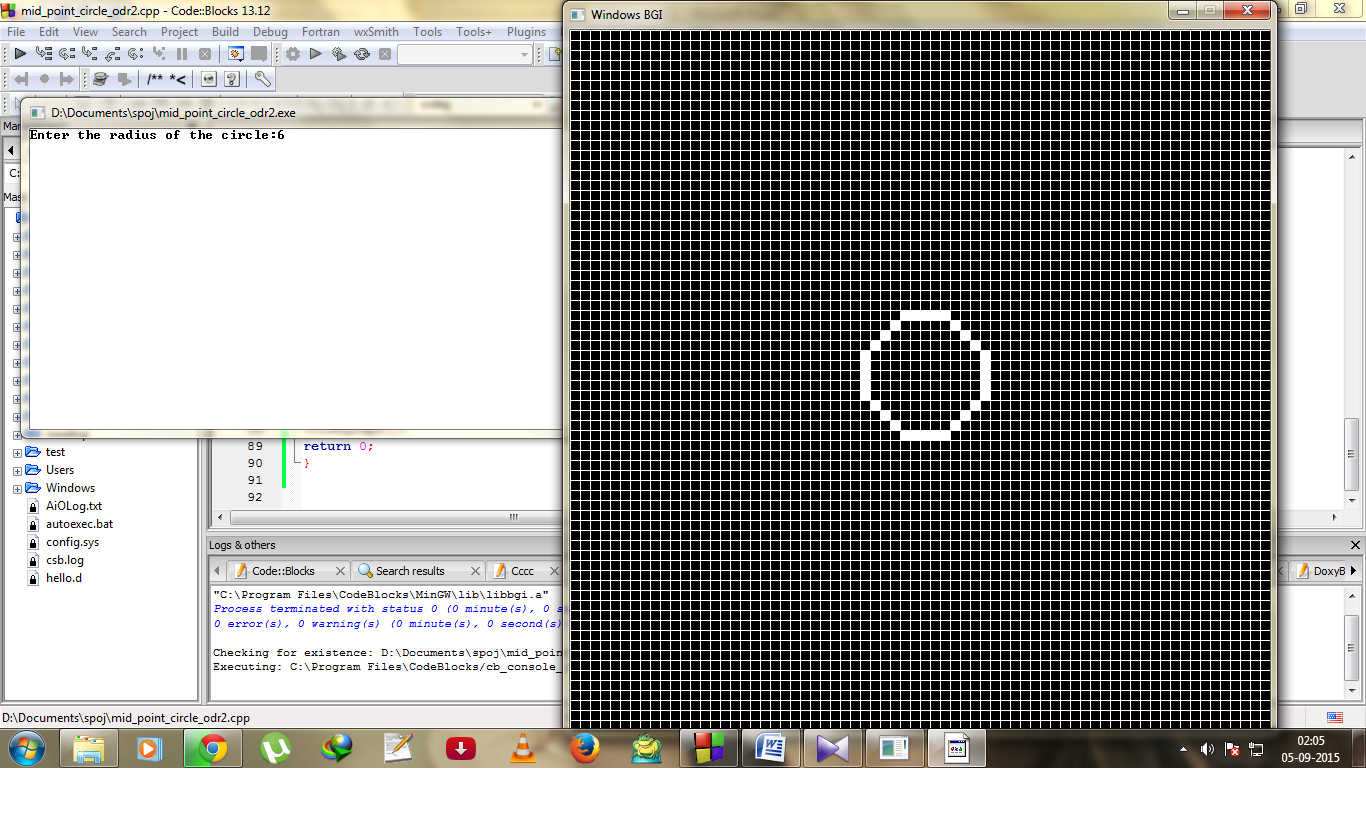
}

getch();

closegraph();

return 0;

}



1. Ellipse Algorithm:

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#include<dos.h>

void plot(int x,int y){

int xmax=getmaxx();

int ymax=getmaxy();

int x\_0 = xmax/2;

int y\_0 = ymax/2;

int x\_c = x\_0 + (x\*10);

int y\_c = y\_0 + (y\*10);

x\_c = x\_c + 5;

y\_c = y\_c - 5;

floodfill(x\_c,y\_c,WHITE);

}

int main(void){

int a, b;

printf("Enter a and b:");

scanf("%d%d",&a,&b);

initwindow(700,700);

int xmax = getmaxx();

int ymax = getmaxy();

int i;

for(i=0; i<=xmax; i+=10){

line(i,0,i,ymax);

}

for(i=0; i<=ymax; i+=10){

line(0,i,xmax,i);

}

double x=0, y=0;

double d1 = (b\*b)-(a\*a\*b)+(0.25\*a\*a);

plot(x,y);

plot(-x,y);

plot(x,-y);

plot(-x,-y);

while(((a\*a)\*(y-0.5))>(b\*b\*(x+1))){

if(d1<0){

d1 = d1+(b\*b\*(2\*x+3));

}

else{

d1 = d1 + (b\*b\*(2\*x+3))+(a\*a\*(2-2\*y));

y = y-1;

}

x = x+1;

plot(x,y);

plot(-x,y);

plot(x,-y);

plot(-x,-y);

}

double d2 = (b\*b\*((x+0.5))) + (a\*a\*(3-2\*y));

while(y>0){

if(d2<0){

d2 = d2+(a\*a\*(3-2\*y));

x=x+1;

}

else{

d2 = d2+(a\*a\*(3-2\*y));

}

y = y-1;

plot(x,y);

plot(-x,y);

plot(x,-y);

plot(-x,-y);

}

getch();

closegraph();

return 0;

}

