**COMPUTER GRAPHICS LAB FILE**

**Subject Code: COE-308**

|  |
| --- |
| **Submitted by:**  **Manish kumar gupta**  **260/co/08**  **Coe-1** |

Contents

[LINE DDA ALGORITHM 3](#_Toc107138117)

[LINE MID - POINT ALGORITHM 4](#_Toc107138118)

[LINE BRESENHAM ALGORITHM 5](#_Toc107138119)

[CIRCLE MID - POINT ALGORITHM 6](#_Toc107138120)

[ELLIPSE MID - POINT ALGORITHM 7](#_Toc107138121)

[HYPERBOLA MID - POINT ALGORITHM 8](#_Toc107138122)

[CIRCLE BRESENHAM ALGORITHM 9](#_Toc107138123)

[ELLIPSE BRESENHAM ALGORITHM 10](#_Toc107138124)

[HYPERBOLA BRESENHAM ALGORITHM 11](#_Toc107138125)

[BOUNDARY FILL ALGORITHM 12](#_Toc107138126)

[FLOOD FILL ALGORITHM 13](#_Toc107138127)

[CIRCLE USING FIRST ORDER DIFFERENTIAL APPROACH 14](#_Toc107138128)

[CIRCLE USING SECOND ORDER DIFFERENTIAL APPROACH 15](#_Toc107138129)

[SCAN LINE ALGORITHM 16](#_Toc107138130)

# LINE DDA ALGORITHM

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,15);

return 0;

}

int algo(float x1,float y1,float x2,float y2)//DDA

{ float x,y,dx,dy,step,xinc,yinc;

dx=x2-x1; dy=y2-y1;

step=(abs((int)dx)>abs((int)dy)) ? dx : dy;

step=abs((int)step);

if(!dx&&!dy)

{ set(x1,y1); return 0;

}

xinc=dx/step; yinc=dy/step;

x=x1; y=y1;

while(x!=x2&&y!=y2)

{ set(x,y);

x+=xinc; y+=yinc;

}

return 0;

}

int main()

{ int gdrv=DETECT,gmd;

float x1,x2,y1,y2;

printf("Enter coordinates(x1,y1): ");

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

printf("Enter coordinates(x2,y2): ");

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

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

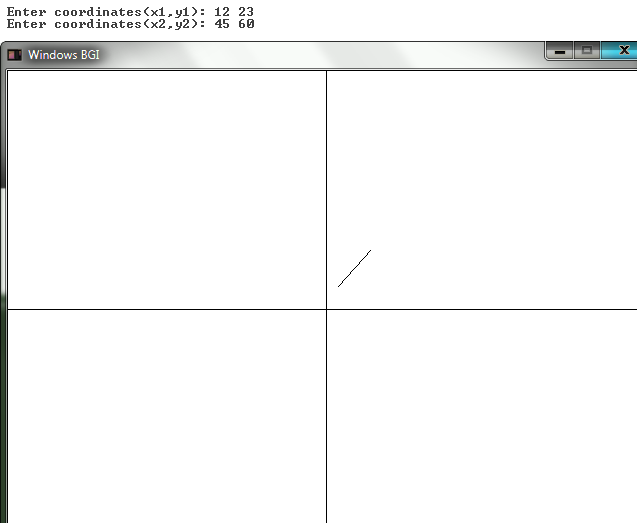
algo(x1,y1,x2,y2);

getch();

closegraph();

return 0;

}



# LINE MID - POINT ALGORITHM

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,15);

return 0;

}

int algo(float x1,float y1,float x2,float y2)//MID-POINT

{ float inc,m,dx,dy,x,y,d,step,xinc,yinc;

dx=x2-x1; dy=y2-y1;

m=(abs((int)dx)>abs((int)dy)) ? dy/dx : dx/dy;

step=(abs((int)dx)>abs((int)dy)) ? dx : dy;

step=abs((int)step);

if(!dx&&!dy)

{ set(x1,y1); return 0;

}

xinc=dx/step; yinc=dy/step;

x=x1; y=y1;

if(abs((int)dx)>abs((int)dy))

{while(x!=x2&&y!=y2)

{ set(x,y);

x+=xinc;

d=y1+m\*(x-x1);

if((y+0.5)>d) y=d;

else y+=yinc;

}

}

else

{while(x!=x2&&y!=y2)

{ set(x,y);

y+=yinc;

d=x1+m\*(y-y1);

if((x+0.5)>d) x=d;

else x+=xinc;

}

}

return 0;

}

int main()

{ int gdrv=DETECT,gmd;

float x1,x2,y1,y2;

printf("Enter coordinates(x1,y1): ");

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

printf("Enter coordinates(x2,y2): ");

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

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

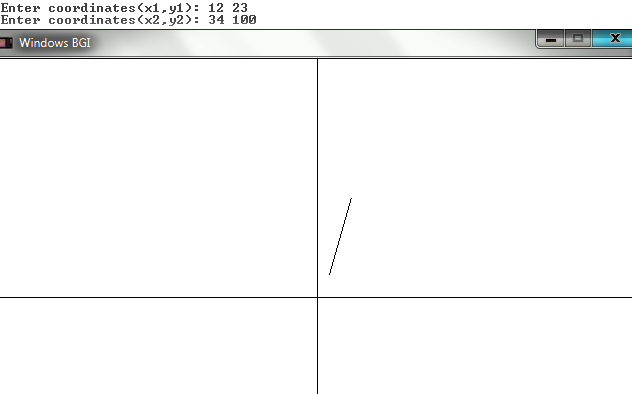
algo(x1,y1,x2,y2);

getch();

closegraph();

return 0;

}



# LINE BRESENHAM ALGORITHM

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,15);

return 0;

}

int algo(float x1,float y1,float x2,float y2)//BRESENHAM

{ float p,dx,dy,x,y,step,xinc,yinc;

dx=x2-x1; dy=y2-y1;

p=(abs((int)dx)>abs((int)dy)) ? dx-2\*dy : dy-2\*dx;

step=(abs((int)dx)>abs((int)dy)) ? dx : dy;

step=abs((int)step);

if(!dx&&!dy)

{ set(x1,y1); return 0;

}

xinc=dx/step; yinc=dy/step;

x=x1; y=y1;

if(abs((int)dx)>abs((int)dy))

{while(x!=x2&&y!=y2)

{ set(x,y);

if(p>0)

{ y=y; p=p-2\*dy; }

else

{ y+=yinc; p=p-2\*dy+2\*dx; }

x+=xinc;

}

}

else

{while(x!=x2&&y!=y2)

{ set(x,y);

if(p>0)

{ x=x; p=p-2\*dx; }

else

{ x+=xinc; p=p-2\*dx+2\*dy; }

y+=yinc;

}

}

return 0;

}

int main()

{ int gdrv=DETECT,gmd;

float x1,x2,y1,y2;

printf("\n Enter coordinates(x1,y1): ");

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

printf(" Enter coordinates(x2,y2): ");

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

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

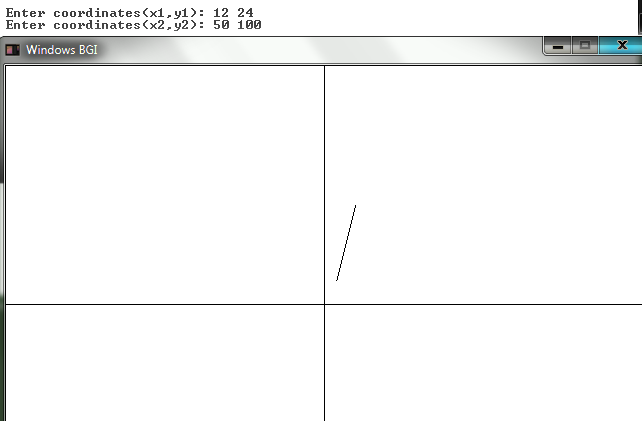
algo3(x1,y1,x2,y2);

getch();

closegraph();

return 0;

}



# CIRCLE MID - POINT ALGORITHM

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,15);

return 0;

}

int algo(float x1,float y1,float r1)//midpoint circle

{ float p,x,y;

p=3/4-2\*r1;

x=r1; y=0;

while(x>=y)

{ set(x1+x,y1+y);set(x1-x,y1+y);

set(x1+x,y1-y);set(x1-x,y1-y);

set(x1+y,y1+x);set(x1-y,y1+x);

set(x1+y,y1-x);set(x1-y,y1-x);

if(p<0)

{ x=x; p=p+2\*y+3; }

else

{ x--; p=p-2\*(x-y)+3; }

y++;

}

}

int main()

{ int gdrv=DETECT,gmd;

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

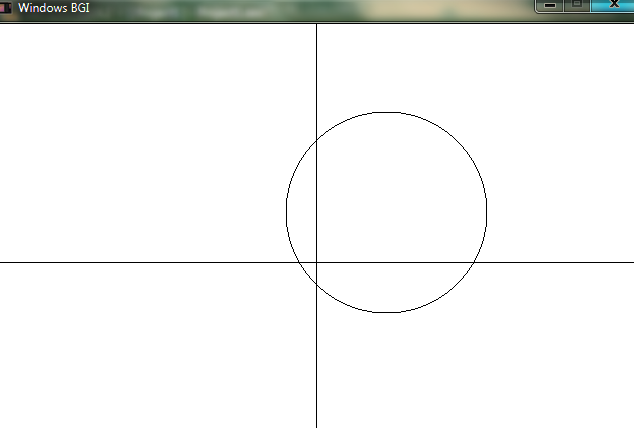
algo(70,50,100);

getch();

closegraph();

return 0;

}



# ELLIPSE MID - POINT ALGORITHM

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,15);

return 0;

}

int algo(float x1,float y1,float a,float b)//MID POINT ellipse

{ float p,x,y,sa,sb;

sa=a\*a; sb=b\*b;

p=sa+sb/4-sb\*a;

x=a; y=0;

while(sb\*x>=sa\*y)

{ set(x1+x,y1+y);set(x1-x,y1+y);set(x1+x,y1-y);set(x1-x,y1-y);

if(p>0)

{ x--; p=p-2\*sb\*x+sa\*(3+2\*y); }

else

{ x=x; p=p+sa\*(3+2\*y); }

y++;

}

p=sa/4+sb-sa\*sb+sb\*x\*(x-2)+sa\*y\*(y+1);

while(x>=0)

{ set(x1+x,y1+y);set(x1-x,y1+y);set(x1+x,y1-y);set(x1-x,y1-y);

if(p>0)

{ y=y; p=p-sb\*(2\*x+1); }

else

{ y++; p=p+2\*sa\*(1+y)-sb\*(2\*x+1); }

x--;

}

}

int main()

{ int gdrv=DETECT,gmd;

float x,y,r1,r2;

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

printf("\n Enter (x,y,a,b): ");

scanf("%f%f%f%f",&x,&y,&r1,&r2);

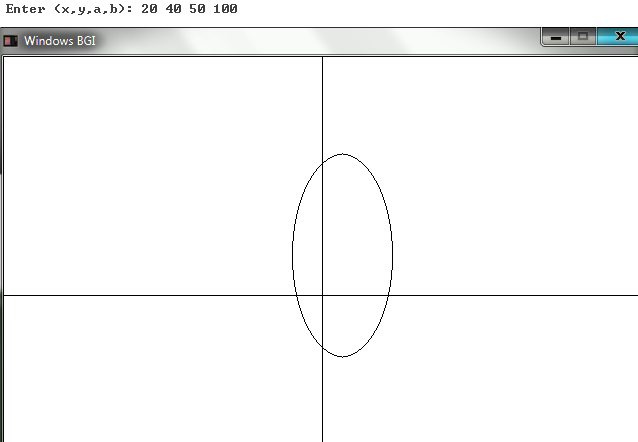
algo(x,y,r1,r2);

getch();

closegraph();

return 0;

}



# HYPERBOLA MID - POINT ALGORITHM

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,15);

return 0;

}

int algo(float x1,float y1,float a,float b)//MID POINT hyperbola

{ float p,x,y,sa,sb;

sa=a\*a; sb=b\*b;

p=-sa+sb/4-sb\*a;

x=a; y=0;

while(x<=5\*a)

{ set(x1+x,y1+y);set(x1-x,y1+y);set(x1+x,y1-y);set(x1-x,y1-y);

if(p<0)

{ x++; p=p+2\*sb\*(x+1)-sa\*(3+2\*y); }

else

{ x=x; p=p-sa\*(3+2\*y); }

y++;

}

}

int main()

{ int gdrv=DETECT,gmd;

float x,y,r1,r2;

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

printf("\n Enter (x,y,a,b): ");

scanf("%f%f%f%f",&x,&y,&r1,&r2);

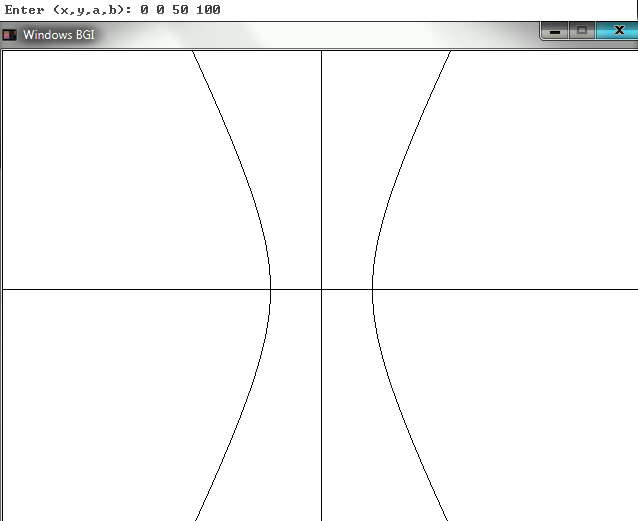
algo(x,y,r1,r2);

getch();

closegraph();

return 0;

}



# CIRCLE BRESENHAM ALGORITHM

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,15);

return 0;

}

int algo(float x1,float y1,float r1)//BRESENHAM circle

{ float p,x,y;

p=3-2\*r1;

y=r1; x=0;

while(y>=x)

{ set(x1+x,y1+y);set(x1-x,y1+y);set(x1+x,y1-y);set(x1-x,y1-y);

set(x1+y,y1+x);set(x1-y,y1+x);set(x1+y,y1-x);set(x1-y,y1-x);

if(p>0)

{ y--; p=p+2\*(3+2\*x)+4\*(1-y); }

else

{ y=y; p=p+2\*(3+2\*x); }

x++;

}

}

int main()

{ int gdrv=DETECT,gmd;

float x,y,r1,r2;

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

printf("\n Enter (x,y,a): ");

scanf("%f%f%f",&x,&y,&r1);

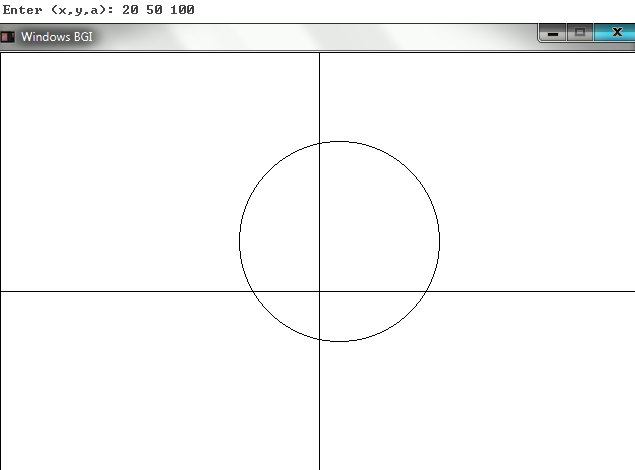
algo(x,y,r1);

getch();

closegraph();

return 0;

}



# ELLIPSE BRESENHAM ALGORITHM

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,15);

return 0;

}

int algo(float x1,float y1,float a,float b)//BRESENHAM ellipse

{ float p,x,y,sa,sb;

sa=a\*a; sb=b\*b;

p=2\*sb-sa-2\*sa\*b;

x=0; y=b;

while(sb\*x<=sa\*y)

{ set(x1+x,y1+y);set(x1-x,y1+y);set(x1+x,y1-y);set(x1-x,y1-y);

if(p>0)

{ y--; p=p+4\*sa\*(1-y)+sb\*2\*(3+2\*x); }

else

{ y=y; p=p+2\*sb\*(3+2\*x); }

x++;

}

p=sb\*(2\*x\*(x+1)+1)+2\*sa\*(1+y\*(y-2))-2\*sa\*sb;

while(y>=0)

{ set(x1+x,y1+y);set(x1-x,y1+y);set(x1+x,y1-y);set(x1-x,y1-y);

if(p>0)

{ x=x; p=p-2\*sa\*(2\*y+1); }

else

{ x++; p=p-2\*sa\*(2\*y+1)+sb\*4\*(x+1); }

y--;

}

}

int main()

{ int gdrv=DETECT,gmd;

float x,y,r1,r2;

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

printf("\n Enter (x,y,a,b): ");

scanf("%f%f%f%f",&x,&y,&r1,&r2);

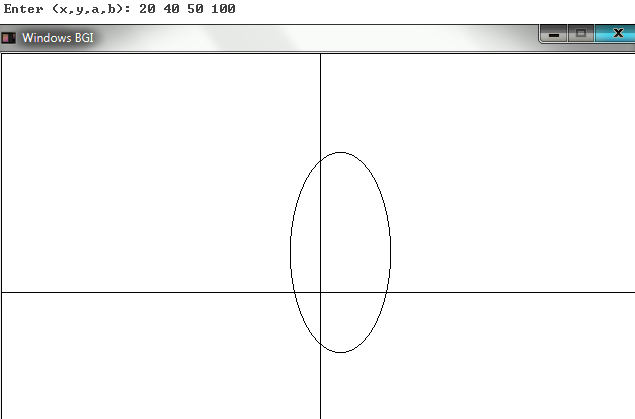
algo(x,y,r1,r2);

getch();

closegraph();

return 0;

}



# HYPERBOLA BRESENHAM ALGORITHM

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,15);

return 0;

}

int algo(float x1,float y1,float a,float b)//BRESENHAM hyperbola

{ float p,x,y,sa,sb;

sa=a\*a; sb=b\*b;

p=-2\*sa+sb+2\*sb\*a;

x=a; y=0;

while(x<=5\*a)

{ set(x1+x,y1+y);set(x1-x,y1+y);set(x1+x,y1-y);set(x1-x,y1-y);

if(p<0)

{ x++; p=p+4\*sb\*(x+1)-2\*sa\*(3+2\*y); }

else

{ x=x; p=p-2\*sa\*(3+2\*y); }

y++;

}

}

int main()

{ int gdrv=DETECT,gmd;

float x,y,r1,r2;

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

printf("\n Enter (x,y,a,b): ");

scanf("%f%f%f%f",&x,&y,&r1,&r2);

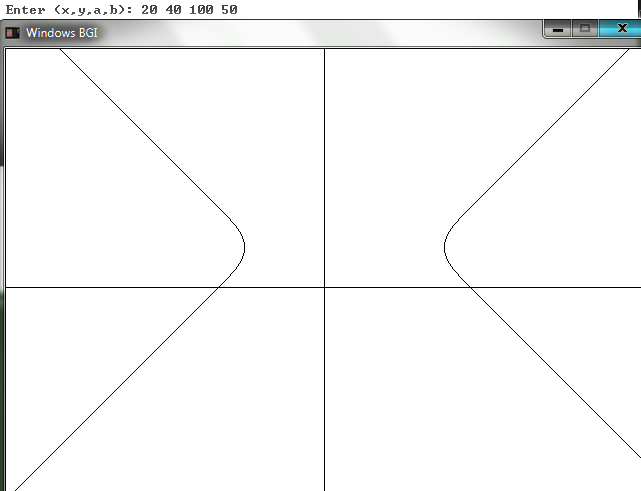
algo(x,y,r1,r2);

getch();

closegraph();

return 0;

}



# BOUNDARY FILL ALGORITHM

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y,int c)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,c);

return 0;

}

int bfill(float x,float y,int bc,int fc)

{ int c=(int)getpixel(319+(int)x,239-(int)y);

if(c!=bc&&c!=fc)

{ set(x,y,fc);

bfill(x+1,y,bc,fc);

bfill(x-1,y,bc,fc);

bfill(x,y+1,bc,fc);

bfill(x,y-1,bc,fc);

}

return 0;

}

int main()

{ int gdrv=DETECT,gmd;

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

rectangle(319-20,239-30,319-120,239-110);

line(319+0,239-0,319-150,239-150);

bfill(-110,100,15,10);

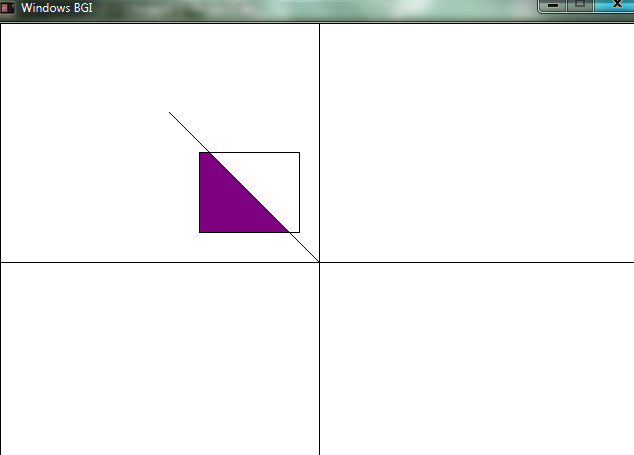
ffill(100,100,0,15);

getch();

closegraph();

return 0;

}



# FLOOD FILL ALGORITHM

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y,int c)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,c);

return 0;

}

int ffill(float x,float y,int pc,int fc)

{ int c=(int)getpixel(319+(int)x,239-(int)y);

if(c==pc)

{ set(x,y,fc);

ffill(x+1,y,pc,fc);

ffill(x-1,y,pc,fc);

ffill(x,y+1,pc,fc);

ffill(x,y-1,pc,fc);

}

return 0;

}

int main()

{ int gdrv=DETECT,gmd;

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

setcolor(4);

line(319+20,239-30,319+120,239-30);

setcolor(6);

line(319+120,239-30,319+120,239-110);

setcolor(11);

line(319+20,239-30,319+20,239-110);

setcolor(9);

line(319+20,239-110,319+120,239-110);

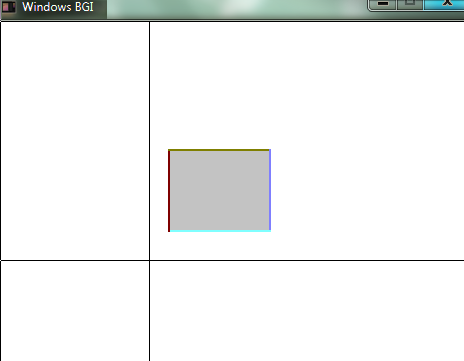
ffill(100,100,0,15);

getch();

closegraph();

return 0;

}



# CIRCLE USING FIRST ORDER DIFFERENTIAL APPROACH

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,15);

return 0;

}

int algo10(float x1,float y1,float r1)//first difference circle

{ float p,x,y;

p=3/4-2\*r1;

x=r1; y=0;

while(x>=y)

{ set(x1+x,y1+y);set(x1-x,y1+y);

set(x1+x,y1-y);set(x1-x,y1-y);

set(x1+y,y1+x);set(x1-y,y1+x);

set(x1+y,y1-x);set(x1-y,y1-x);

if(p<0)

{ x=x; p=p+2\*y+3; }

else

{ x--; p=p-2\*(x-y)+3; }

y++;

}

}

int main()

{ int gdrv=DETECT,gmd;

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

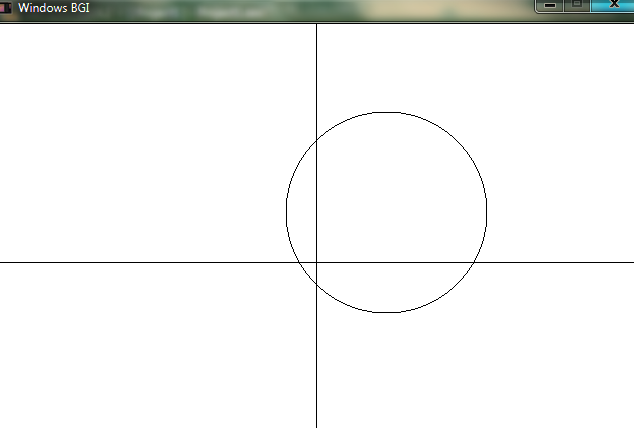
algo10(70,50,100);

getch();

closegraph();

return 0;

}



# CIRCLE USING SECOND ORDER DIFFERENTIAL APPROACH

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

int set(float x,float y)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,15);

return 0;

}

int algo(float x1,float y1,float r1)//second difference circle

{ float d,e,se,x,y;

d=3/4-2\*r1;

e=3; se=3-2\*r1;

x=r1; y=0;

while(x>=y)

{ set(x1+x,y1+y);set(x1-x,y1+y);

set(x1+x,y1-y);set(x1-x,y1-y);

set(x1+y,y1+x);set(x1-y,y1+x);

set(x1+y,y1-x);set(x1-y,y1-x);

if(d<0)

{ x=x; d+=e;e+=2;se+=2; }

else

{ x--; d+=se;e+=2;se+=4; }

y++;

}

}

int main()

{ int gdrv=DETECT,gmd;

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

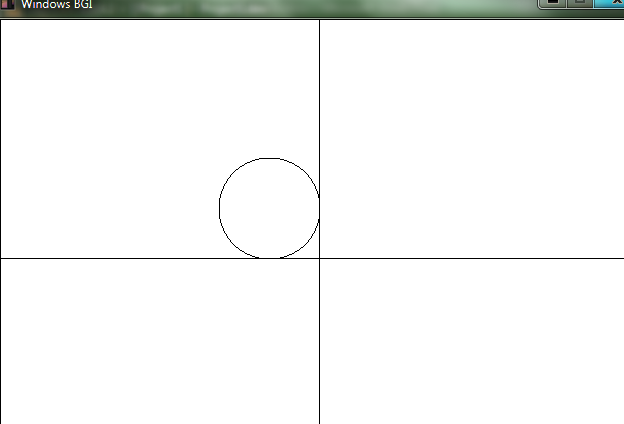
algo(-50,50,50);

getch();

closegraph();

return 0;

}



# SCAN LINE ALGORITHM

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#define R(a) (a+0.5)

#define N 6

int set(float x,float y,int c)

{ x=x+319; y=239-y;

x=R(x); y=R(y);

putpixel((int)x,(int)y,c);

return 0;

}

int algo()

{ float dx,dy,x,y,max,min;

float pt[N+1][2]={ {50, 20},{20, 40},{35, 60},{50, 40},{65,60},{80,40},{50, 20} };

float edge[N][2],m[N]={0},ar[N]={0};//x,y ymin,ymax

int i,j,count;

for(i=0;i<N;i++)//edge ymin,ymax

{ if((int)pt[i][1]>(int)pt[i+1][1])

{ edge[i][0]=pt[i+1][1];

edge[i][1]=pt[i][1];

}

else

{ edge[i][0]=pt[i][1];

edge[i][1]=pt[i+1][1];

}

}

for(i=0;i<N;i++)//edge m

{ dx=(pt[i][0]-pt[i+1][0]);

dy=(pt[i][1]-pt[i+1][1]);

if(!dy&&!dx) m[i]=0;

else m[i]=dx/dy;

}

min=240;//start y

max=-240;//end y

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

{ if((int)min>(int)edge[i][0])

min=edge[i][0];

if((int)max<(int)edge[i][1])

max=edge[i][1];

}

for(y=min+1;y<max;y++)

{

for(i=0,count=0;i<N;i++)//intersect x

{ if((int)y>=(int)edge[i][0]&&(int)y<=(int)edge[i][1])

{ ar[count]=pt[i][0]+m[i]\*(y-pt[i][1]);

if(ar[count]==ar[count-1])

{ if((edge[i][0]==edge[i-1][1])&&(edge[i][1]>edge[i-1][1]))

count--;

if((edge[i-1][0]==edge[i][1])&&(edge[i][0]<edge[i-1][1]))

count--;

}

count++;

}

}

for(i=0;i<count;i+=2)

{ for(x=ar[i];x<ar[i+1];x++)

set(x,y,10);

}

}

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

line(319+(int)pt[i][0],239-(int)pt[i][1],319+(int)pt[i+1][0],239-(int)pt[i+1][1]);

return 0;

}

int main()

{ int gdrv=DETECT,gmd,n;

initgraph(&gdrv,&gmd,"");

graphdefaults();

rectangle(0,0,639,479);

line(319,0,319,479);

line(0,239,639,239);

algo();

getch();

closegraph();

return 0;

}

