4. DRAW A TRIANGLE

```c
#include <stdio.h>
#include <conio.h>
int main()
{
    int i, j;
    clrscr(); // Clear the screen (Turbo C++ specific)
    for (j = 0; j < 4; j++) // Loop for rows
    {
        for (i = 0; i < 4 - j - 1; i++) // Print leading spaces
        {
            printf(" ");
        }
        for (i = 0; i <= j; i++) // Print stars and spaces between them
        {
            printf("*");
            if (i < j)
            {
                printf("   "); // Space between stars
```

```c
        }
    }
    printf("\n");
}


    getch(); // Wait for a key press (Turbo C++ specific)
    return 0;
}
```

OUTPUT:

5. DRAW A CIRCLE USING INBUILD FUNCTION

```c
#include<stdio.h>
#include<graphics.h>
int main()
{
    int gd =DETECT,gm,x,y,radius;
    initgraph(&gd,&gm," ");
    printf("enter the value of x:");
    scanf("%d",&x);
    printf("enter the value of y:");
    scanf("%d",&y);
```

```c
    printf("enter the value of radius:");
    scanf("%d",&radius);
    circle(x,y,radius);
    getch();
    closegraph();
    return 0;
}
```

OUTPUT: enter the value of x 150

enter the value of y 150

enter the value of radius 100

6.DRAW A CIRCLE USING LOOP FUNCTION

```c
#include<stdio.h>
#include<graphics.h>
int main()
{
    int gd =DETECT,gm,x,y,radius,i,j,k;
    initgraph(&gd,&gm," ");
```

```c
printf("enter the value of x:");
scanf("%d",&x);
printf("enter the value of y:");
scanf("%d",&y);
printf("enter the value of radius:");
scanf("%d",&radius);
for(i=x,j=y,k=radius;i<=150,j<=150,k<100;i++,j++,k++)
{
circle(i,j,k);
}
getch();
closegraph();
return 0;
}
```

OUTPUT: enter the value of x 20

ENTER THE value of y 20

enter radius 10

7.DRAW A LINE USING LINE FUNCTION

```c
#include<stdio.h>
#include<graphics.h>
int main()
{
    int gd =DETECT,gm,x1,x2,y1,y2;
    initgraph(&gd,&gm," ");
    printf("enter the value of x1:");
    scanf("%d",&x1);
    printf("enter the value of y1:");
    scanf("%d",&y1);
    printf("enter the value of x2:");
    scanf("%d",&x2);
    printf("enter the value of y2:");
    scanf("%d",&y2);
    line(x1,y1,x2,y2);
    getch();
    closegraph();
    return 0;
}
```

OUTPUT: enter the value of x1 0

enter the value of y1 0

enter the value of x1 100

enter the value of y2 300

8.DRAW A CURVE USING INBUILD FUNCTION

```c
#include<stdio.h>
#include<graphics.h>
int main()
{
    int gd =DETECT,gm,x,y,stA,enA,radius;
    initgraph(&gd,&gm," ");
    printf("enter the value of x:");
    scanf("%d",&x);
    printf("enter the value of y:");
    scanf("%d",&y);
    printf("enter the starting angle:");
    scanf("%d",&stA);
```

```c
    printf("enter the ending angle:");

    scanf("%d",&enA);

    printf("enter radius");

    scanf("%d",&radius);

    arc(x,y,stA,enA,radius);

    getch();

    closegraph();

    return 0;
}
```

OUTPUT: 200

         200

         100

         200

         100

9.DRAW A RECTANGLE USING INBUILD FUNCTION

```c
#include<stdio.h>
#include<graphics.h>
```

```c
int main()
{
    int gd =DETECT,gm,leftB,leftT,rightT,rightB;
    initgraph(&gd,&gm," ");
    printf("enter the left bottom:");
    scanf("%d",&leftB);
    printf("enter the left top:");
    scanf("%d",&leftT);
    printf("enter the right top:");
    scanf("%d",&rightT);
    printf("enter the right bottom:");
    scanf("%d",&rightB);
    rectangle(leftB,leftT,rightT,rightB);
    getch();
    closegraph();
    return 0;
}
```

OUTPUT: 50
        100

100

50

10. DRAW A BAR USING INBUILD FUNCTION

```c
#include<stdio.h>
#include<graphics.h>
int main()
{
    int gd =DETECT,gm,left,top,right,bottom;
    initgraph(&gd,&gm," ");
    printf("enter the value of left side in a bar:");
    scanf("%d",&left);
    printf("enter the value of top in a bar:");
    scanf("%d",&top);
    printf("enter the value of right side in a bar :");
    scanf("%d",&right);
    printf("enter the value of bottom in a bar:");
    scanf("%d",&bottom);
    bar(left,top,right,bottom);
```

```
    getch();

    closegraph();

    return 0;

}
```

OUTPUT: 250

  350

  350

  250

11. 3D BAR USING INBUILD FUNCTION

```
#include <stdio.h>

#include<graphics.h>

int main()

{

int gd=DETECT,gm,left,top,right,bottom,depth,topflag;

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

printf("Enter the value of leftside of a 3d bar:");

scanf("%d",& left);

printf("Enter the value of top of a 3d bar:");
```

```c
scanf("%d",& top);
printf("Enter the value of righttside of a 3d bar:");
scanf("%d",& right);
printf("Enter the value of bottom of a 3d bar:");
scanf("%d",& bottom);
printf("Enter the value of depth of a 3d bar:");
scanf("%d",& depth);
printf("Enter the value of topflag of a 3d bar:");
scanf("%d",& topflag);
bar3d(left,top,right,bottom,depth,topflag);
getch();
closegraph();
return 0;
}
```

OUTPUT: 100
      300
      150
      50
      70

12.DRAW A ELLIPSE USING INBUILD FUNCTION

```c
#include<stdio.h>
#include<graphics.h>
int main()
{
int gd=DETECT,gm,x,y,stangle,endangle,xradius,yradius;
initgraph(&gd,&gm," ");
printf("Enter the value of x axsis:");
scanf("%d",&x);
printf("Enter the value of y axsis:");
scanf("%d",&y);
printf("Enter the value of stangle:");
scanf("%d",&stangle);
printf("Enter the value of endangle:");
scanf("%d",&endangle);
printf("Enter the value of xradius:");
```

```
scanf("%d",&xradius);

printf("Enter the value of yradius:");

scanf("%d",&yradius);

ellipse(x,y,stangle,endangle,xradius,yradius);

getch();

closegraph();

return 0;

}
```

OUTPUT: 100

100

0

360

100

65

13. OUTEXT USING INBUILD FUNCTION

```
#include<stdio.h>

#include<graphics.h>
```

```
int main()
{
int gd=DETECT,gm,text;
initgraph(&gd,&gm," ");
delay(2000);
outtext("name");
getch();
closegraph();
return 0;
}
```

OUTPUT: NAME

14. DRAW A DDA LINE ALGORITHM

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

int main()
```

```c
{
int gd= DETECT, gm, x1, y1, x2,y2,i,j, color;
float dx,dy,m;
initgraph(&gd,&gm," ");
printf("enter the value of x1 and y1;");
scanf("%d%d", &x1,&y1);
printf("enter the value of x2 and y2;");
scanf("%d%d", &x2,&y2);
dx=x2-x1;
dy=y2-y1;
m=dy/dx;
for(i=x1,j=y1;i<=x2,j<=y2;i++, j++)
{
if (m<1)
{
putpixel(x1+i,y1+(1/m),5);
putpixel(x1+2+i,y1+(1/m),6);
putpixel (x1+5 +i, y1+5+(1/m),7);
putpixel (x1+ 10+i, y1+10+(1/m),8);
```

```
}
else
{
putpixel(x1+i, y1+m, 5);
putpixel(x1+2+i,y1+m,6);
putpixel(x1+5+i,y1+5+m, 7);
putpixel (x1+10+i,y1+10+m,8);

}
}
getch();
closegraph();
return 0;
}
```

OUTPUT: 200 300

500 400

15. BRESHENHAM LINE DRAWING

```c
#include<stdio.h>
#include<graphics.h>

void drawline(int x0, int y0, int x1, int y1)
{
    int dx, dy, p, x, y;

        dx=x1-x0;
        dy=y1-y0;

        x=x0;
        y=y0;

        p=2*dy-dx;

        while(x<x1)
        {
            if(p>=0)
            {
```

```c
            putpixel(x,y,7);

            y=y+1;

            p=p+2*dy-2*dx;

        }

        else

        {

            putpixel(x,y,7);

            p=p+2*dy;

        }

        x=x+1;

    }

}

int main()
{
    int gdriver=DETECT, gmode, error, x0, y0, x1, y1;
    initgraph(&gdriver, &gmode, " ");

    printf("Enter co-ordinates of first point: ");
    scanf("%d%d", &x0, &y0);
```

```
        printf("Enter co-ordinates of second point: ");

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

        drawline(x0, y0, x1, y1);


        return 0;
}
```

OUTPUT: 200 200

     400 200

16.DRAW TRANSLATION FIGURE

```
#include <stdio.h>

#include <conio.h>

#include <graphics.h>

void main()

{
```

```c
    int
gd=DETECT,gm,x1,y1,x2,y2,x3,y3,tx,ty,x11,y11,x22,y22,
x33,y33,a;
    initgraph (&gd,&gm," ");
    Printf("Enter the value of x1 and y1:");
    Scanf("%d%d",&x1,&y1);
    Printf("Enter the value of x2 and y2:");
    Scanf("%d%d",&x2,&y2);
    Printf("Enter the value of x3 and y3:");
    Scanf("%d%d",&x3,&y3);
    line(x1,y1,x2,y2);
    line(x2,y2,x3,y3);
    line(x3,y3,x1,y1);
    Printf ("1 for translate xy 2 for translate X 3 for
translate y or 4 for other's");
    Printf("Enter the value of a:");
    Scanf("%d",&a);
    if(a==1)
{
    Printf("Enter the value of tx and ty:");
    Scanf("%d%d",&tx,&ty);
```

```c
    x11=x1+tx;

    y11=y1+ty;

    x22=x2+tx;

    y22=y2+ty;

    x33=x3+tx;

    y33=y3+ty;

}

else if(a==2)

{

    Printf("Enter the value of tx and ty:");

    scanf("%d%d",&tx,&ty);

    x11=x1+tx;

    y11=y1+0;

    x22=x2+tx;

    y22=y2+0;

    x33=x3+tx;

    y33=y3+0;

}

else if(a==3)

{
```

```c
Printf("Enter the value of tx and ty:");
scanf("%d%d",&tx,&ty);
x11=x1+0;
y11=y1+ty;
x22=x2+0;
y22=y2+ty;
x33=x3+0;
y33=y3+ty;
}
else
{
 Printf("invalid");
}
line (x11,y11,x22,y22);
line (x22,y22,x33,y33);
line (x33,y33,x11,y11);
getch();
closegnaph();
return 0;
}
```

OUTPUT: 300 300

400 300

350 200

VALUE OF A : 1

40 40

17.DRAW A ROTATION FIGURE

#include <stdio.h>

#include <conio.h>

#include <graphics.h>

#include <math.h> // Include math.h for sin() and cos() functions

int main()

{

   int gd = DETECT, gm;

   int x1, y1, x2, y2, x3, y3; // Initial triangle vertices

   int x11, y11, x22, y22, x33, y33; // Rotated triangle vertices

```c
float angle, c, s; // Angle of rotation, cosine, sine

initgraph(&gd, &gm, "C:\\Turboc3\\BGI"); // Initialize
graphics mode (use correct path)

// Input the coordinates of the triangle vertices
printf("Enter the value of x1 and y1: ");
scanf("%d%d", &x1, &y1);
printf("Enter the value of x2 and y2: ");
scanf("%d%d", &x2, &y2);
printf("Enter the value of x3 and y3: ");
scanf("%d%d", &x3, &y3);

// Draw the original triangle
line(x1, y1, x2, y2);
line(x2, y2, x3, y3);
line(x3, y3, x1, y1);

// Input the angle of rotation in degrees
printf("Enter the angle of rotation in degrees: ");
```

```c
scanf("%f", &angle);

// Calculate cosine and sine of the angle
c = cos(angle * 3.14 / 180);
s = sin(angle * 3.14 / 180);

// Perform rotation for each vertex of the triangle
x11 = floor(x1 * c + y1 * s);
y11 = floor(-x1 * s + y1 * c);

x22 = floor(x2 * c + y2 * s);
y22 = floor(-x2 * s + y2 * c);

x33 = floor(x3 * c + y3 * s);
y33 = floor(-x3 * s + y3 * c);

// Draw the rotated triangle
line(x11, y11, x22, y22);
line(x22, y22, x33, y33);
line(x33, y33, x11, y11);
```

```c
    getch(); // Wait for a key press
    closegraph(); // Close graphics mode
    return 0;
}
```

OUTPUT: 100 100

200 100

150 50

90

18.SCALING FIGURE

```c
#include <stdio.h>
#include <graphics.h>

int main()
{
    int gd = DETECT, gm;
    int x1, y1, x2, y2, x3, y3, x4, y4;
```

```c
int x11, y11, x22, y22, x33, y33, x44, y44;
int d, sf;

initgraph(&gd, &gm, " "); // Initialize graphics mode (use correct path)

// Input the coordinates of the square
printf("Enter the value of x1, y1: ");
scanf("%d%d", &x1, &y1);
printf("Enter the value of x2, y2: ");
scanf("%d%d", &x2, &y2);
printf("Enter the value of x3, y3: ");
scanf("%d%d", &x3, &y3);
printf("Enter the value of x4, y4: ");
scanf("%d%d", &x4, &y4);

// Draw the initial square
line(x1, y1, x2, y2);
line(x2, y2, x3, y3);
line(x3, y3, x4, y4);
```

```c
    line(x4, y4, x1, y1);

    // Prompt user for scaling choice and factor
    printf("Do you want to enlarge the square? (1 for yes, 0 for no): ");
    scanf("%d", &d);
    printf("Enter the scaling factor: ");
    scanf("%d", &sf);

    if (d == 1)
    {
        // Enlarge the square
        x11 = x1 * sf;
        y11 = y1 * sf;
        x22 = x2 * sf;
        y22 = y2 * sf;
        x33 = x3 * sf;
        y33 = y3 * sf;
        x44 = x4 * sf;
        y44 = y4 * sf;
```

```
    // Draw the enlarged square

    line(x11, y11, x22, y22);

    line(x22, y22, x33, y33);

    line(x33, y33, x44, y44);

    line(x44, y44, x11, y11);

}

else if (d == 0)

{

    // Shrink the square

    x11 = x1 / sf;

    y11 = y1 / sf;

    x22 = x2 / sf;

    y22 = y2 / sf;

    x33 = x3 / sf;

    y33 = y3 / sf;

    x44 = x4 / sf;

    y44 = y4 / sf;


    // Draw the shrunken square
```

```c
        line(x11, y11, x22, y22);

        line(x22, y22, x33, y33);

        line(x33, y33, x44, y44);

        line(x44, y44, x11, y11);

    }

    else

    {

        printf("Invalid choice.\n");

    }


    getch(); // Wait for a key press

    closegraph(); // Close graphics mode

    return 0;

}
```
output:

100 100

150 100

150 150

100 150

1

19.GEOMETRIC SHAPE....

```c
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int main()
{
int gd=DETECT,gm,x1,y1,x2,y2,x3,y3,x4,y4,shf,d,x11,y11,x22,y22,x33,y33,x44,y44;
initgraph(&gd,&gm," ");
printf("Enter the first coordinate value of rectangle:");
scanf("%d%d",&x1,&y1);
printf("Enter the second coordinate value of rectangle:");
scanf("%d%d",&x2,&y2);
printf("Enter the third coordinate value of rectangle:");
scanf("%d%d",&x3,&y3);
```

```c
printf("Enter the forth coordinate value of rectangle:");
scanf("%d%d",&x4,&y4);
line(x1,y1,x2,y2);
line(x2,y2,x3,y3);
line(x3,y3,x4,y4);
line(x4,y4,x1,y1);
printf("enter the value of sharing factor:");
scanf("%d",&shf);
printf("1 for first coordinate_/n");
printf("2 for second coordinate_/n");
printf("3 for third coordinate_/n");
printf("4 for forth coordinate_/n");
printf("enter your decision sharing:");
scanf("%d",&d);
if(d==1)
{
    x11=x1*shf;
    y11=y2*shf;
    x22=x2;
    y22=y2;
```

```
        x33=x3;

        y33=y3;

        x44=x4;

        y44=y4;

    }

    else if(d==2)

    {

        x11=x1;

        y11=y1;

        x22=x2;

        y22=y2;

        x33=x3;

        y33=y3*shf;

        x44=x4;

        y44=y4;

    }

    else if(d==3)

    {

        x11=x1;

        y11=y1;
```

```
        x22=x2;

        y22=y2;

        x33=x3;

        y33=y3;

        x44=x4;

        y44=y4;

    }

    else if(d==4)

    {

        x11=x1;

        y11=y1;

        x22=x2;

        y22=y2;

        x33=x3;

        y33=y3;

        x44=x4*shf;

        y44=y4*shf;

    }

    line(x11,y11,x22,y22);

    line(x22,y22,x33,y33);
```

line(x33,y33,x44,y44);

line(x44,y44,x11,y11);

getch();

closegraph();

return 0;

}

OUTPUT: 100 100

200 100

200 200

100 200

5

4

20.COHEN SUTHERLAND...

#include<graphics.h>

#include<conio.h>

#include<stdio.h>

#include<math.h>

#include<dos.h>

void main()

{

```c
int
rcode_begin[4]={0,0,0,0},rcode_end[4]={0,0,0,0},region
_code[4];

int W_xmax,W_ymax,W_xmin,W_ymin,flag=0;

float slope;

int x,y,x1,y1,i, xc,yc;

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

printf("\n Cohen Sutherlsnd Line Clipping
algorithm\n");


//Reading coordinates of window

printf("\n Enter Window XMin, YMin =");

scanf("%d %d",&W_xmin,&W_ymin);

printf("\n Enter Window XMax, YMax =");

scanf("%d %d",&W_xmax,&W_ymax);


//Reading coordinates of any line

printf("\n Please enter intial point x and y= ");

scanf("%d %d",&x,&y);

printf("\n Now, enter final point x1 and y1= ");
```

```c
scanf("%d %d",&x1,&y1);
//clearing all the text in the screen
cleardevice();

//creating window
rectangle(W_xmin,W_ymin,W_xmax,W_ymax);

//creating line
line(x,y,x1,y1);

//Code for giving 4 bit code to first end point of line
// Top
if(y>W_ymax)  {
rcode_begin[0]=1;
flag=1 ;
}
// Bottom
if(y<W_ymin) {
rcode_begin[1]=1;
flag=1;
```

```
}
// Right
if(x>W_xmax) {
rcode_begin[2]=1;
flag=1;
}
//Left
if(x<W_xmin)  {
rcode_begin[3]=1;
flag=1;
}

//Code for giving 4 bit code to lat end point of line
// Top
if(y1>W_ymax){
rcode_end[0]=1;
flag=1;
}
// Bottom
if(y1<W_ymin) {
```

```c
rcode_end[1]=1;
flag=1;
}
// Right
if(x1>W_xmax){
rcode_end[2]=1;
flag=1;
}
 //Left
if(x1<W_xmin){
rcode_end[3]=1;
flag=1;
 }

 //Check condition for clipping
if(flag==0)
{
printf("No need of clipping as it is already in window");
}
flag=1;
```

```c
for(i=0;i<4;i++){
region_code[i]= rcode_begin[i] && rcode_end[i] ;
if(region_code[i]==1)
 flag=0;
}
if(flag==0)
{
printf("\n Line is completely outside the window");
}
else{
//calculating slope
slope=(float)(y1-y)/(x1-x);
//first point left
if(rcode_begin[2]==0 && rcode_begin[3]==1)
{
y=y+(float) (W_xmin-x)*slope ;
x=W_xmin;
}

//first point right
```

```c
if(rcode_begin[2]==1 && rcode_begin[3]==0)
{
y=y+(float) (W_xmax-x)*slope ;
x=W_xmax;
}

//first point top
if(rcode_begin[0]==1 && rcode_begin[1]==0)
{
x=x+(float) (W_ymax-y)/slope ;
y=W_ymax;
}

//first point bottom
if(rcode_begin[0]==0 && rcode_begin[1]==1)
{
x=x+(float) (W_ymin-y)/slope ;
y=W_ymin;
}
```

```c
// end points left
if(rcode_end[2]==0 && rcode_end[3]==1)
{
y1=y1+(float) (W_xmin-x1)*slope ;
x1=W_xmin;


}
// end points right
if(rcode_end[2]==1 && rcode_end[3]==0)
{
y1=y1+(float) (W_xmax-x1)*slope ;
x1=W_xmax;


}
//end points top
if(rcode_end[0]==1 && rcode_end[1]==0)
{
x1=x1+(float) (W_ymax-y1)/slope ;
y1=W_ymax;
```

```
}
//end points bottom
if(rcode_end[0]==0 && rcode_end[1]==1)
{
x1=x1+(float) (W_ymin-y1)/slope ;
y1=W_ymin;

}
}
//closing of else condition
delay(1000);
clearviewport();
rectangle(W_xmin,W_ymin,W_xmax,W_ymax);
setcolor(10);
line(x,y,x1,y1);
getch();
closegraph();
}
OUTPUT:
```

200 200

500 500

150 150

350 350

21.COMPOSITE FIGURE....

```
#include<stdio.h>

#include <conio.h>

#include <graphics.h>

int main()

{

int gd=DETECT,gm,x1,y1,x2,y2,x3,y3,dv,tx,ty,x11,y11,x22,y22,x33,y33,d,x41,y41,x42,y42,x43,y43,sf;

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

Printf("Enter the values of x1 and y1:");

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

Printf("Enter the values of x2 and y2:");

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

Printf("Enter the values of x3 and y3:");

Scanf("%d%d",&x3,&y3);
```

```
line(x1,y1,x2,y2);
line(x2,y2,x3,y3);
line(x3,y3,x1,y1);
Printf("Enter 1 for translate tx & ty:\n Enter 2 for
translate tx:\n Enter 3 for translate ty; ");
Printf("Enter your decision value: ");
Scanf("%d",&dv);
if (dv==1)
{
Printf("Enter the values of tx and ty:");
Scanf("%d%d",&tx,&ty);
x11=x1+tx;
y11=y1+ty;
x22=x2+tx;
y22=y2+ty;
X33=X3+tx;
y33=y3+ty;
}
else if(dv==2)
{
```

```c
Printf("Enter the value of tx: ");
Seanf("%d", &tx);
x11=x1+tx;
y11=y1+0;
x22=x2+tx;
y22=y2+0;
X33=X3+tx;
y33=y3+0;
}
else if (dv==3)
{
Printf("Enter the value of ty:");
Scanf("%d",&ty);
x11=x1+0;
y11=y1+ty;
x22=x2+0;
y22=y2+ty;
x33=x3+0;
y33=y3+ty;
}
```

```
else
{
Printf("Given value is invalid");
}
line(x11,y11,x22,y22);
line(x22,y22,x33,y33);
line(x33,y33,x11,y11);
Printf("Enter 1 to see larger triangle; \n Enter  to see smaller triangle");
Printf(" Enter your decision value : ");
Scanf("%d", &d);
if(d==1)
{
Printf ("Enter the scaling factors: ");
Scanf("%d",&sf);
x41=x11*sf;
y41=y11*sf;
x42=x22*sf;
y42=y22*sf;
x43=x33*sf;
```

```c
y43=y33*sf;
}
else if(d==0)
{
Printf("Enter the scaling factor");
Scanf("%d",&sf);
x41=x11/sf;
y41=y11/Sf;
x42=x22/sf;
y42=y22/sf;
x43=x33/sf;
y43=y33/Sf;
}
else
{
Printf("Given value is invalid");
}
line(x41,y41,x42,y42);
line(x42,y42,x43,y43);
line(x43,y43,x41,y41);
```

```
getch();

closegraph();

return 0;

}
```

OUTPUT: 55

1

2

22. **FILL ALGORITHM…**

```
#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main()

{

 int gd= DETECT,gm,i,j;

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

 for (i=100; i<=200;i++)

{

   for(J=100;J<=200;j++)

   {
```

```
        putpixel(i,j,RED);

    }

}

getch();

closegraph();

return 0;

}


OUTPUT:
```

23.MID POINT ALGORITHM....

```
#include<stdio.h>

#include<graphics.h>

#include<conio.h>

void Bcircle(int r);

void main()

{

  int gd=DETECT,gm;          /* request auto detection */

  int r,i;
```

```c
    initgraph(&gd,&gm,"C:\\TC\\BGI");    /* initialize
graphic mode */
  for(i=0;i<2;i++)
  {
    printf("Enter the radius of circle:");
    scanf("%d",&r);
    Bcircle(r);
  }
    getch();
}
void Bcircle(int r)
{
  int x,y,xc=200,yc=200,p;    /*Predefined center of
circle*/
  x=0;
  y=r;


  p=3-(2*r);
  for(x=0;x<=y;x++)
  {
    putpixel(xc+x,yc-y,1);
```

```c
if(p<0)
{
  y=y;
  p=(p+(4*x)+1);
}
else
{ y=y-1;
  p=(p+(4*(x-y)+1));
}
/*plot points on all eight octants circle centered*/



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);
  }
}
```

BRAINWARE UNIVERSITY

Ref No.:- BWU/FNC/STD/2024/003   Date: July 05, 2024

NOTICE

Sub: <u>Payment of Odd Semester Fees July 2024</u>

Odd Semester fees may be deposited through the following modes
1. In **Cash** *or* **Debit/Credit card** *or* by **Draft** (in favour of **Brainware University, payable at Kolkata**) at **The University campus** Accounts Department during office hours (10.30 am – 4.30 pm)

2. **Through Online Banking**

Relevant details of bank transfer are given below:	
Bank Name:	HDFC Bank Ltd.
Account Holder's Name:	BRAINWARE UNIVERSITY
Account No:	50100246296609
Account Type:	Savings
IFS Code:	HDFC0000352
Branch Name:	BARASAT
Branch Address:	N/39, 1st Floor, KNC Rd, opposite Bidhan Cinema, Haritala, Gupta Colony, Barasat, Kolkata, West Bengal 700124

3. Through Student Self Service portal

Process:Select:
https://www.brainwareuniversity.ac.in/studentselfservice
→ Selfservice → Payment.

4. Through UPI (QR Code)

You are requested to inform about your online / UPI payment details to the following e-mail Id: accounts@brainwareuniversity.ac.in

Please provide the following information while you are sharing your payment information to the above-mentioned e-mail id.

1. Student Name
2. Student Code
3. Transaction ID
4. Bank Name of the University
5. Amount

- Last date of payment is July 31st, 2024 (Wednesday).
- ***Delayed payment will attract penalty as per University rules (Rs.70/- per day including Sundays & Holidays)***
- Students are advised to inform their parents /guardians accordingly.
- Any query related to payment contact in **033-69010504**.

Stay Safe, Stay Healthy!

Sd/-

Abir Saha

Finance Officer

Brainware University