Ex. No. 1(A)	IMPLEMENTATION OF BRESENHAM'S LINE DRAWING ALGORITHM

AIM

To draw a line using Bresenham’s line drawing algorithm using C program.

ALGORITHM

Step 1: Start the program.

Step 2: Declare the necessary variables.

Step 3: Initialize the graph using dx, dy, gd, gm.

Step 4: Assign the values of x1, y1 to x, y respectively.

Step 5: Similarly, absolute values to dx, dy.

Step 6: put pixel and set 15 to pixel position.

Step 7: Using do-while loop, put e,x,y,I values.

Step 8: Stop the program.

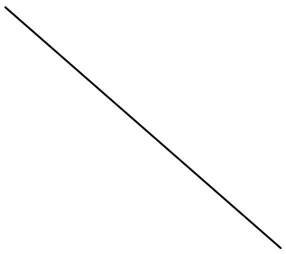
PROGRAM

```
#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)&&(y<=y1))
   {
      if(p>=0)
```

```
        {
            putpixel(x,y,7);
            If(y!=y1) y=y+1;
            p=p+2*dy-2*dx;
        }
        else
        {
            putpixel(x,y,7);
            p=p+2*dy;
        }
            if(x!=x1)
            x=x+1;
            if((x==x1)&&(y==y1))
                        break;
    }
}
int main()
{
    int gdriver=DETECT, gmode, error, x0, y0, x1, y1;
    initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");
    printf("Enter first co-ordinates point: ");
    scanf("%d%d", &x0, &y0);
    printf("Enter second co-ordinates point: ");
    scanf("%d%d", &x1, &y1);
    drawline(x0, y0, x1, y1);
    return 0;
```

OUTPUT

```
Enter first co-ordinates point: 100 100

Enter the second co-ordinates point: 300 300
```



RESULT

Thus the program to draw a line using Bresenham’s algorithm is implemented and executed successfully.

Ex. No. 1(B)	**IMPLEMENTATION OF BRESENHAM'S CIRCLE DRAWING ALGORITHM**

AIM

To draw a circle using Bresenham's circle drawing algorithm using C program.

ALGORITHM

Step 1: Start the program.

Step 2: Declare the necessary variables.

Step 3: Create the function for Circle.

Step 4: Enter the radius and center values.

Step 5: Initialize the graph with gd, gm and assign y<-radius.

Step 6: Start the circle function and p<- 1- radius.

Step 7: Check the while loop until the condition is satisfied.

Step 8: Check the if –else condition until the condition is satisfied**.**

Step 9: Assign all operation for circle function and the values.

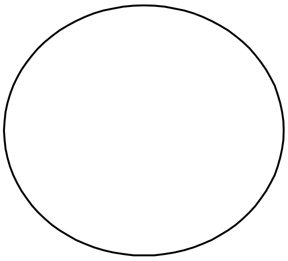
Step 10: Stop the Program.

PROGRAM

```
#include<stdio.h>

#include<graphics.h>

void main()

{

int gd=DETECT,gm=0,xa,ya,r,p,k,x,y;

initgraph(&gd,&gm,"D:\\TC\\BGI");

printf("Enter the coordinates");

scanf("%d%d",&xa,&ya,);

printf("Enter the radius");

scanf("%d",&r);

p=1-r;
x=0;
y=r;
for(k=0;x<y;k++)
{
if(p<0)
{
p=p+(2*x)+2+1;
x=x+1;
}
else
{
p=p+(2*x)+2+1-(2*y)+2;
x=x+1;
y=y-1;
}
putpixel(xa+x,ya+y,1);
putpixel(xa-x,ya+y,2);
putpixel(xa+x,ya-y,3);
putpixel(xa-x,ya-y,4);
putpixel(xa+y,ya+x,5);
putpixel(xa-y,ya+x,6);
putpixel(xa+y,ya-x,7);
putpixel(xa-y,ya-x,8);
}
}
```

OUTPUT

```
Enter the coordinates: 75 50

Enter the radius: 40
```



RESULT

Thus the C program to draw a circle using Bresenham’s algorithm is implemented and executed successfully.

Ex. No. 2	IMPLEMENTATION OF BRESENHAM'S ELLIPSE DRAWING ALGORITHM

AIM

To draw an ellipse using Bresenham's ellipse drawing algorithm using C progam.

ALGORITHM

Step 1: Start

Step 2: Declare the necessary variables .

Step 3: Initialize the gd, gm.

Step 4: Get the input, X radius of the ellipse and Y radius of the ellipse.

Step 5: Calculate rxsq,rysq, tworxsq, tworxsq values.

Step 6: Initialize the x value to 0 and y to ry.

Step 7 : Calculate the dx and dy values.

Step 8: In do loop, create a function to draw ellipse.

Step 9: Check the if –else conditions, and assign all the values to draw the ellipse.

Step 10: Stop the program.

PROGRAM

```
#include <stdio.h>

#include <conio.h>

#include <graphics.h>

#include <math.h>

#include  <dos.h>

void main()

{

long int d1,d2;

int i,gd=DETECT ,gm,x,y;

long int rx,ry,rxsq,rysq,tworxsq,tworysq,dx,dy;

printf("Enter the x Radius of the ellipse");
```

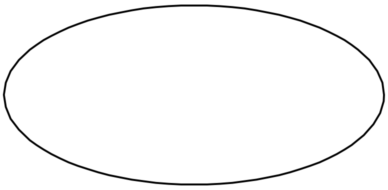
```
scanf("%ld",&rx);
printf("Enter the y Radius of the ellipse");
scanf("%ld",&ry);
initgraph(&gd,&gm," ");
rxsq=rx*rx;
rysq=ry*ry;
tworxsq=2*rxsq;
tworysq=2*rysq;
x=0;
y=ry;
d1=rysq - (rxsq * ry) + (0.25 * rxsq);
dx= tworysq * x;
dy= tworxsq * y;
do
{
  putpixel(200+x,200+y,15);
  putpixel(200-x,200-y,15);
  putpixel(200+x,200-y,15);
  putpixel(200-x,200+y,15);
  if (d1 < 0)
   {
    x=x+1;
    y=y;
     dx=dx + tworysq;
     d1=d1 + dx + rysq;
     }
   else
```

```
 {
  x=x+1;
  y=y-1;
  dx= dx + tworysq;
  dy= dy - tworxsq;
  d1= d1 + dx - dy + rysq;
  }
  delay(50);
  }while (dx < dy);
 d2 = rysq * ( x + 0.5) * ( x + 0.5 ) + rxsq * (y - 1) * (y-1) - rxsq * rysq;
  do
 {
 putpixel(200+x,200+y,15);
putpixel(200-x,200-y,15);
putpixel(200+x,200-y,15);
putpixel(200-x,200+y,15);
if (d2 >0)
{
x=x;
y=y-1;
dy = dy - tworxsq;
d2 = d2 - dy + rxsq;
}
else
{
x= x+1;
y=y-1;
```

```
  dy=dy - tworxsq;
  dx= dx + tworysq;
  d2 = d2 + dx -dy + rxsq;
  }
  delay(50);
} while ( y> 0);
getch();
closegraph();
}
```

OUTPUT

```
Enter the x Radius of the ellipse: 100
Enter the y Radius of the ellipse: 50
```



RESULT

Thus the program to draw an ellipse using Bresenham’s algorithm is implemented and executed successfully.