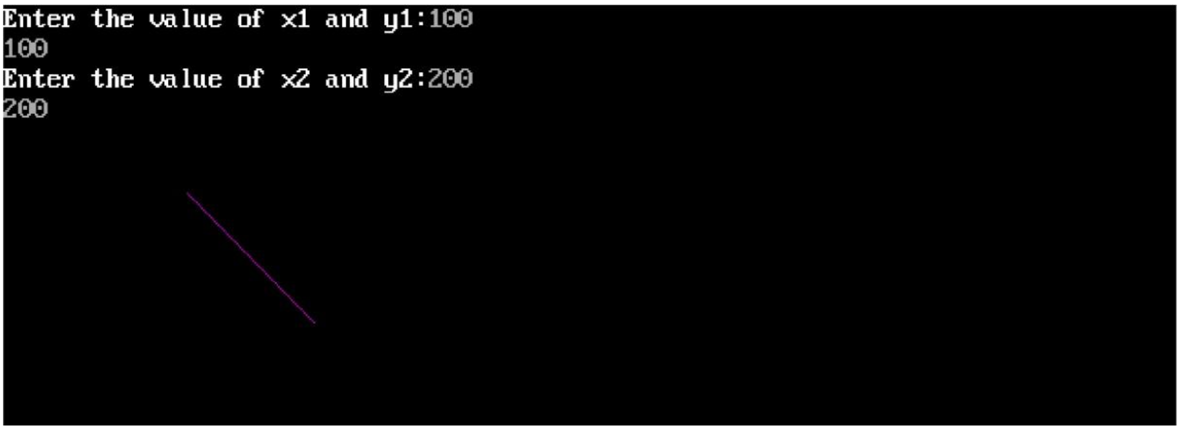
NAME - SAIVARUN KATKAM

ROLL NO - 55

Q1 - Write a Program in C for DDA Line Drawing Algorithm.

PROGRAM

```
#include <graphics.h>
#include <stdio.h>
#include<conio.h>
#include <math.h>
#include <dos.h>
void main( )
{
float x,y,x1,y1,x2,y2,dx,dy,step;
int i,gd=DETECT,gm;
initgraph(&gd,&gm,"..\\BGI ");
printf("Enter the value of x1 and y1 : ");
scanf("%f%f",&x1,&y1);
printf("Enter the value of x2 and y2: ");
scanf("%f%f",&x2,&y2);
dx=abs(x2-x1);
dy=abs(y2-y1);
if(dx>=dy)

step=dx;
else
step=dy;
dx=dx/step;
dy=dy/step;
x=x1;
y=y1;

i=1;

while(i<=step)

{

putpixel(x,y,3);

x=x+dx;

y=y+dy;

i=i+1;

delay(100);

}

getch();

closegraph();

}
```



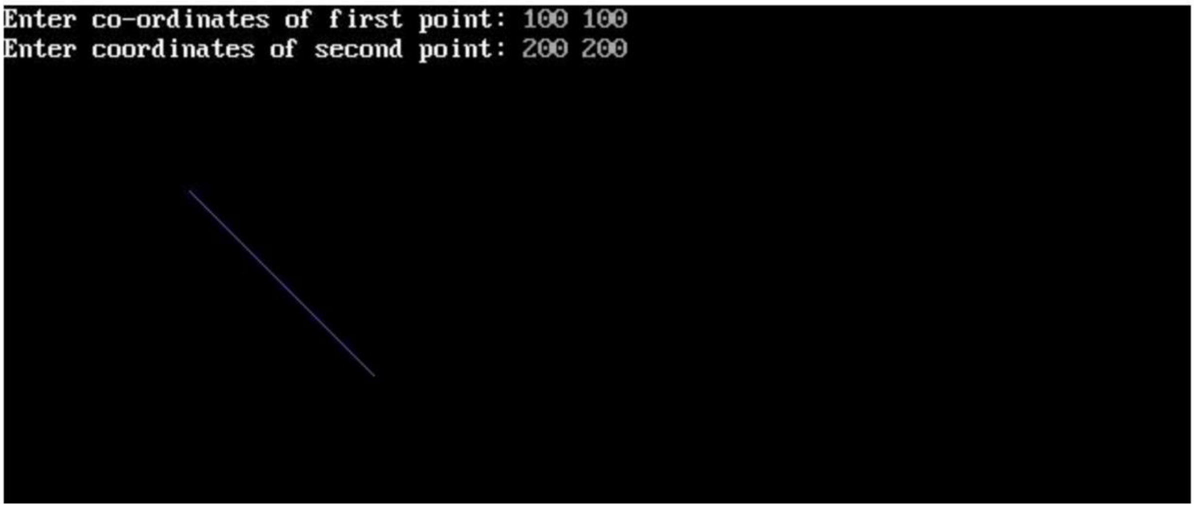
```
Enter the value of x1 and y1:100
100
Enter the value of x2 and y2:200
200
```

Q2 - Write a Program in C to implement Bresenham’s Line Algorithm.

solution -

```
#include<stdio.h>

#include<conio.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)
{
 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, "..\\BGI ");
 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);
 getch();
 closegraph();
   return 0;
   }
```



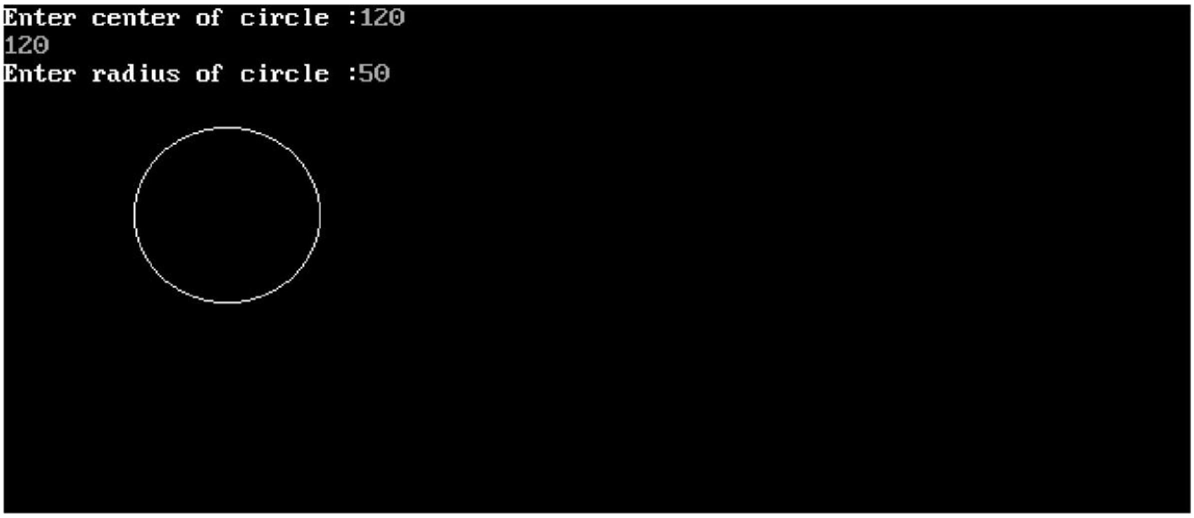
```
Enter co-ordinates of first point: 100 100
Enter coordinates of second point: 200 200
```

Q3 - Write a Program in C to implement Midpoint Circle algorithm

solution -

```
  #include<graphics.h>
  #include<conio.h>
  #include<stdio.h>
  void main()
  {
  int x,y,x_mid,y_mid,radius,dp;
  int g_mode,g_driver=DETECT;
  clrscr();
  initgraph(&g_driver,&g_mode,"..\\BGI ");
  printf("*********** MID POINT Circle drawing algorithm
  ********\n\n");
  printf("\nenter the coordinates= ");
  scanf("%d %d",&x_mid,&y_mid);
```

```
printf("\n now enter the radius =");


scanf("%d",&radius);
x=0;
y=radius;
dp=1-radius;
do
{
putpixel(x_mid+x,y_mid+y,YELLOW);
putpixel(x_mid+y,y_mid+x,YELLOW);
putpixel(x_mid-y,y_mid+x,YELLOW);
putpixel(x_mid-x,y_mid+y,YELLOW);
putpixel(x_mid-x,y_mid-y,YELLOW);
putpixel(x_mid-y,y_mid-x,YELLOW);
putpixel(x_mid+y,y_mid-x,YELLOW);
putpixel(x_mid+x,y_mid-y,YELLOW);
if(dp<0) {
dp+=(2*x)+1;
}
else{
y=y-1;
dp+=(2*x)-(2*y)+1;
}
x=x+1;
}
while(y>x);
getch();
}
```



```
Enter center of circle :120
120
Enter radius of circle :50
```

Q4 - Write a Program in C to implement Filling Algorithm: Boundary Fill and Flood Fill

solution -

a. Write a program to fill a circle using Flood Fill Algorithm.

solution -

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
void flodfill(int x,int y,int f,int o)
{
int c;
c=getpixel(x,y);
if(c==o)
{
setcolor(f);
putpixel (x,y,f);
delay(10);
flodfill(x+1,y,f,o);
flodfill(x,y+1,f,o);

flodfill(x+1,y+1,f,o);
flodfill(x-1,y-1,f,o);
flodfill(x-1,y,f,o);
flodfill(x,y-1,f,o);
flodfill(x-1,y+1,f,o);
flodfill(x+1,y-1,f,o);
}
}
void main()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,"..\\BGI ");
rectangle(50,50,100,100);
flodfill(51,51,4,0);
getch();
}
```

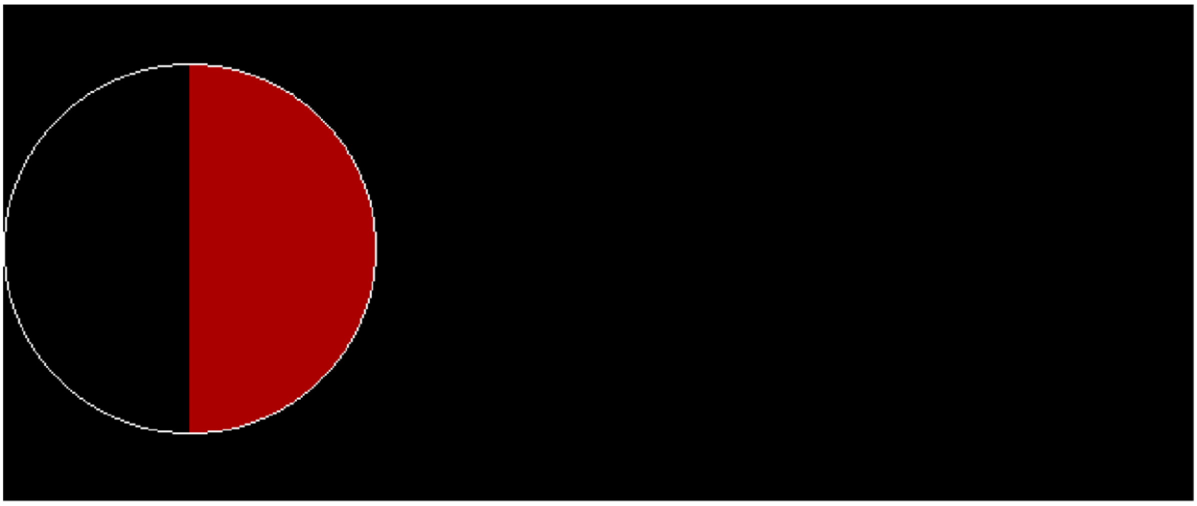
b. Write a program to fill a circle using Boundary Fill Algorithm.

Solution:-

```
#include<graphics.h>
#include<dos.h>
#include<conio.h>
void boundaryFill8(int x, int y, int fill_color,int
boundary_color)
{
```

```
if(getpixel(x, y) != boundary_color && getpixel(x, y) !=fill_color)
{
putpixel(x, y, fill_color);
boundaryFill8(x + 1, y, fill_color, boundary_color);
boundaryFill8(x, y + 1, fill_color, boundary_color);
boundaryFill8(x - 1, y, fill_color, boundary_color);
boundaryFill8(x, y - 1, fill_color, boundary_color);
boundaryFill8(x - 1, y - 1, fill_color, boundary_color);
boundaryFill8(x - 1, y + 1, fill_color, boundary_color);
boundaryFill8(x + 1, y - 1, fill_color, boundary_color);
boundaryFill8(x + 1, y + 1, fill_color, boundary_color);
}
}

void main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "..\\BGI ");// Rectangle function
rectangle(50, 50, 100, 100);// Function calling
boundaryFill8(55, 55, 4, 15);
delay(10000);
getch();
   /*closegraph function closes the graphics mode and deallocates all memory allocated by
graphics system .*/
closegraph();
```

```
}
```



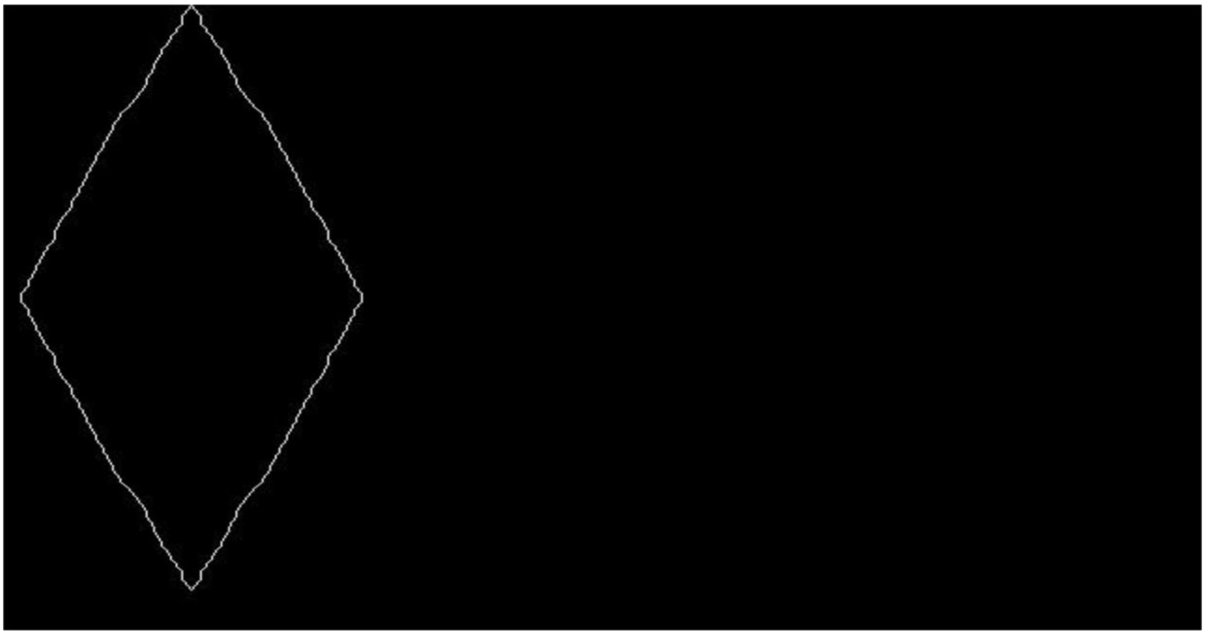
Q5 - Write a Program in C to implement Midpoint Ellipse algorithm.

solution -

```
#include<conio.h>
#include<dos.h>
#include<stdio.h>
#include<graphics.h>
void main(){
long x,y,x_center,y_center;
long a_sqr,b_sqr, fx,fy, d,a,b,tmp1,tmp2;
int g_driver=DETECT,g_mode;
clrscr();

initgraph(&g_driver,&g_mode,"..\\BGI ");
printf("********* MID POINT ELLIPSE ALGORITHM *********");
printf("\n\n Enter coordinate x and y = ");

scanf("%ld%ld",&x_center,&y_center);
printf("\n Now enter constants a and b = ");
scanf("%ld%ld",&a,&b);
```

```
x=0;
y=b;
a_sqr=a*a;
b_sqr=b*b;
fx=2*b_sqr*x;
fy=2*a_sqr*y;
d=b_sqr-(a_sqr*b)+(a_sqr*0.25);
do
{
 putpixel(x_center+x,y_center+y,1);
 putpixel(x_center-x,y_center-y,1);
 putpixel(x_center+x,y_center-y,1);
 putpixel(x_center-x,y_center+y,1);

 if(d<0)
 {
 d=d+fx+b_sqr;
 }
 else

 {
  y=y-1;
  d=d+fx+-fy+b_sqr;
  fy=fy-(2*a_sqr);
  }
  x=x+1;
  fx=fx+(2*b_sqr);
  delay(10);

  }
  while(fx<fy);
```

```
tmp1=(x+0.5)*(x+0.5);
tmp2=(y-1)*(y-1);
d=b_sqr*tmp1+a_sqr*tmp2-(a_sqr*b_sqr);
do
{
putpixel(x_center+x,y_center+y,1);
putpixel(x_center-x,y_center-y,1);
putpixel(x_center+x,y_center-y,1);
putpixel(x_center-x,y_center+y,1);

if(d>=0)

d=d-fy+a_sqr;
else

{
x=x+1;
d=d+fx-fy+a_sqr;
fx=fx+(2*b_sqr);
}
y=y-1;
fy=fy-(2*a_sqr);
}
while(y>0);
getch();
closegraph();
}
```



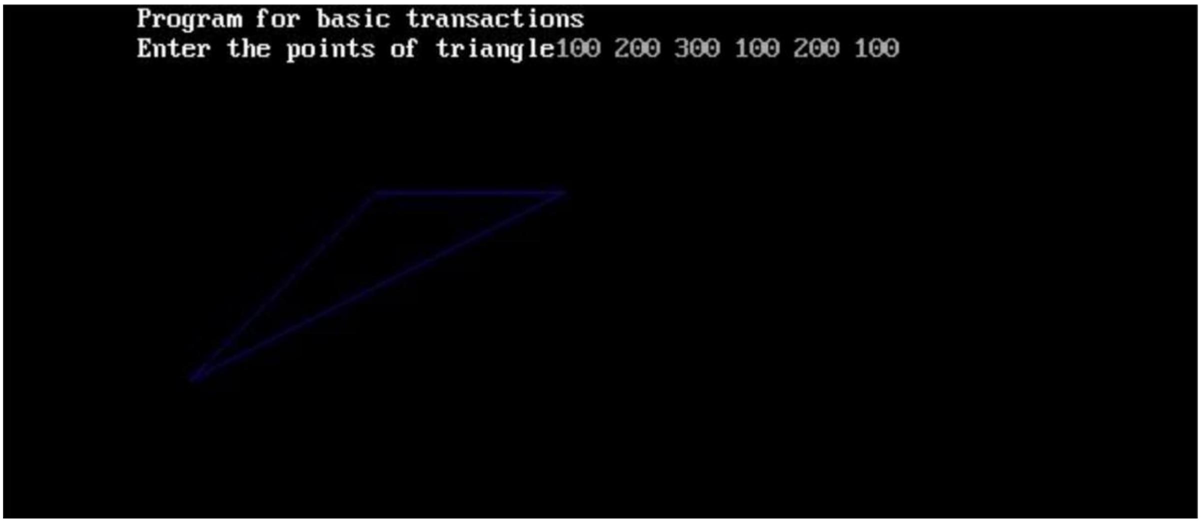
Q6 - Write a Program in C to implement 2D transformations:

1. Translation 2. Rotation 3. Scaling

1. Write a program to perform 2D translation.

Solution:-

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
void main()
{
int i;
int gd=DETECT,gm;

int x2,y2,x1,y1,x,y;
initgraph(&gd,&gm,"..\\BGI ");
printf("Enter the two endpoints of a line:x1,y1,x2,y2:\n");
scanf("%d\n%d\n%d\n%d",&x1,&y1,&x2,&y2);
line(x1,y1,x2,y2);
printf("\n Enter the translation coordinates:x y");
scanf("%d%d",&x,&y);
x1=(x1+x);
y1=(y1+y);

x2=(x2+x);
y2=(y2+y);
printf("line after Translation");
line(x1,y1,x2,y2);
getch();
closegraph();
}
```



```
Program for basic transactions
Enter the points of triangle100 200 300 100 200 100
```

2. write a program Perform 2D Rotation

.

Solution:-

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
```

```
{
 int i;
 int gd=DETECT,gm;
 int x2,y2,x1,y1,x,y,xn,yn;
 double r11,r12,r21,r22,th;
 initgraph(&gd,&gm,"..\\BGI ");
 printf("Enter the two endpoints of a line:x1,y1,x2,y2:\n");

 scanf("%d\n%d\n%d\n%d\n",&x1,&y1,&x2,&y2);
 line(x1,y1,x2,y2);
 printf("\n\n Enter the angle");
 scanf("%lf",&th);
 r11=cos((th*3.14)/180);
 r12=sin((th*3.14)/180);
 r21=(-sin((th*3.14)/180));
 r22=cos((th*3.14)/180);
 xn=((x2*r11)-(y2*r21));
 yn=((x2*r12)+(y2*r22));
 line(x1,y1,xn,yn);
 getch();
 closegraph();
 }
```

3. Write a program to implement 2D scaling.

solution -

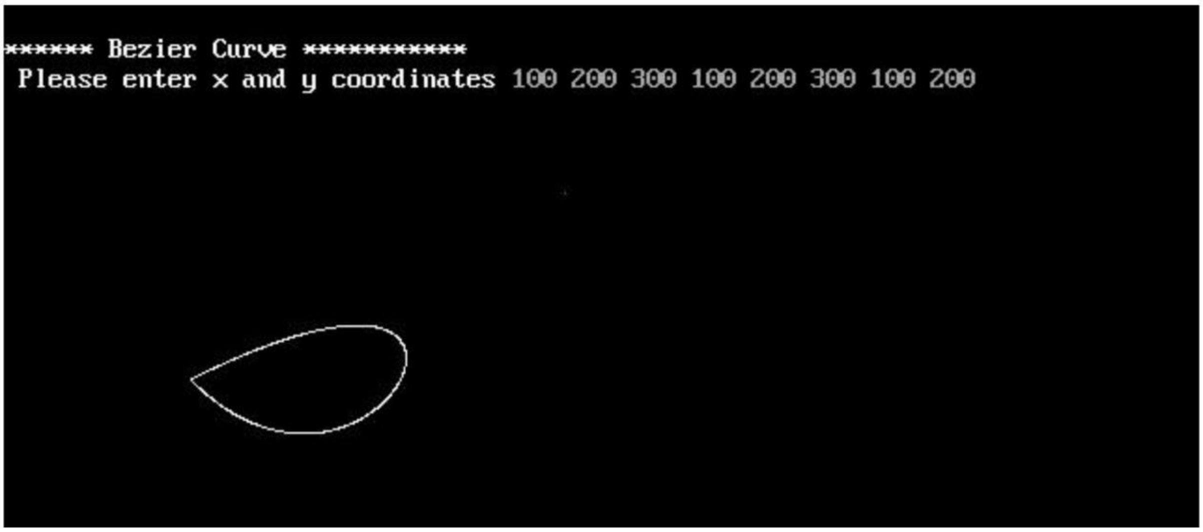
```
 #include<graphics.h>
 #include<stdio.h>
 #include<conio.h>
```

```
void main()
{
int i;
int gd=DETECT,gm;
int x2,y2,x1,y1,x,y;
initgraph(&gd,&gm,"..\\BGI ");
printf("Enter the two endpoints of a line:x1,y1,x2,y2:\n");
scanf("%d\n%d\n%d\n%d",&x1,&y1,&x2,&y2);
line(x1,y1,x2,y2);
printf("Enter the scaling coordinates:x\t y\t");
scanf("%d%d",&x,&y);
x1=(x1*x);
y1=(y1*y);
x2=(x2*x);

y2=(y2*y);
printf("line after scaling");
line(x1,y1,x2,y2);
getch();
closegraph();
}
```

Q7 - Write a Program in C to implement Bezier Curve

solution -

```
#include<graphics.h>
#include<math.h>
#include<conio.h>
#include<stdio.h>
int main()
{
    int x[4],y[4],i;

    double put_x,put_y,t;

    int gr=DETECT,gm;

    initgraph(&gr,&gm,"..\\BGI ");

    printf("\n*** Bezier Curve ****");

    printf("\n Please enter x and y coordinates ");

    for(i=0;i<4;i++)

    {

    scanf("%d%d",&x[i],&y[i]);

    putpixel(x[i],y[i],3);

    }

    for(t=0.0;t<=1.0;t=t+0.001)

    {

    put_x = pow(1-t,3)x[0] + 3*t*pow(1-t,2)*x[1] + 3*t*t(1-t)*x[2] + pow(t,3)*x[3];// Formula to
draw curve

    put_y = pow(1-t,3)y[0] + 3*t*pow(1-t,2)*y[1] + 3*t*t(1-t)*y[2] + pow(t,3)*y[3];
putpixel(put_x,put_y, WHITE);

    }

    getch();

    closegraph();

    }
```

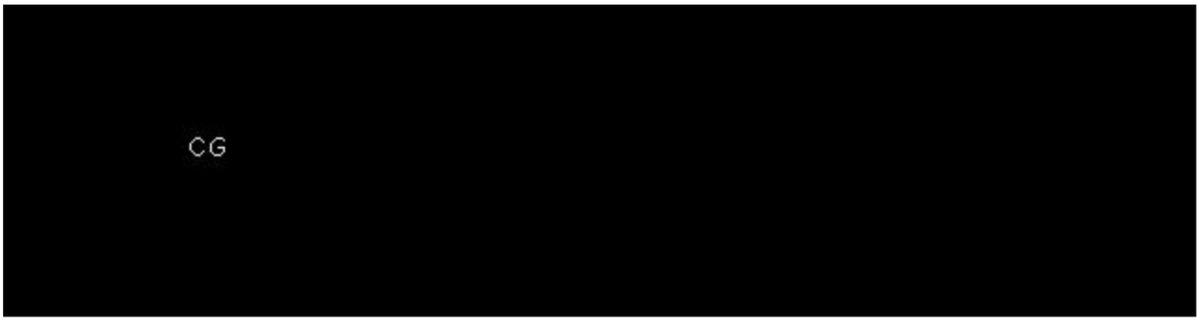


```
****** Bezier Curve ************
 Please enter x and y coordinates 100 200 300 100 200 300 100 200
```

Q8 - Write a Program in C for Character Generation

solution -

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int main()
{
int gd=DETECT,gm,i,j;
int a[20][20]={{0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,0,0},
{0,0,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0},
{0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,1},
{1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0},
{1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0},
{1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1,1,1,0},
{1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0},
{0,1,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,1,0},
{0,0,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0},
{0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,0,0,0}};
initgraph(&gd,&gm,"..\\BGI ");
for(i=0;i<19;i++)
{
for(j=0;j<19;j++)
{
if(a[i][j]==1)
putpixel(100+j,200+i,WHITE);
}
}
getch();
return 0;
}
```



```
CG
```