(1) WAP to do a line & increase its length 3 times

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>
Void ScaledLine(int x1, int y1, int x2, int y2, int Sx, int Sy){
x1 = x1*Sx;
y1 = y1*Sy;
x2 = x2*Sx;
y2 = y2*Sy;
Cout<<"Co-ordinates after scaling are"<<x1<<""<<y1<<""<<x2<<""<<y2;
x1+=90;
x2+=90;
y1+=90;
y2+=90;
Setcolor(RED);
Line(x1, y1, x2, y2);
}
void main(){
int gd=DETECT, gm;
initgraph(&gd, &gm, "..\\BGI");
int x1, y1, x2, y2, Sx, Sy;
Cout<<"Enter 2 Co-ordinates of line";
Cin>>x1>>y1>>x2>>y2;
Cout<<"Enter the scaling factors x & y";
Cin>>Sx>>Sy
Line(x1, y1, x2, y2);
ScaledLine(x1, y1, x2, y2, Sx, Sy);
getch();
closegraph();
}
```

(2) WAP to draw a triangle & translate it:

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
void drawTriangle(int x1, int y1, int x2, int y2, int x3, int y3){
  line(x1, y1, x2, y2);
  line(x1, y1, x3, y3);
  line(x2, y2, x3, y3);
}
void translateTriangle(int x1, int y1, int x2, int y2, int x3,
        int y3, int Tx, int Ty){
  x1 = x1 + Tx;
  x2 = x2 + Tx;
  x3 = x3 + Tx;
  y1 = y1 + Ty;
  y2 = y2 + Ty;
  y3 = y3 + Ty;
  setcolor(RED);
  drawTriangle(x1, y1, x2, y2, x3, y3);
}
void main(){
int gd = DETECT, gm;
int x1, y1, x2, y2, x3, y3, Tx, Ty;
initgraph(&gd, &gm, "..\\BGI");
cout<<"Enter 3 co-ordinates of Traiangle";
cin>>x1>>y1>>x2>>y2>>x3>>y3;
cout<<"Enter Translation factor for x & y";
cin>>Tx>>Ty;
drawTriangle(x1, y1, x2, y2, x3, y3);
translateTriangle(x1, y1, x2, y2, x3, y3, Tx, Ty);
getch();
closegraph();
}
```

(3) Scaling of a Triangle.

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
void main(){
 int gd=DETECT,gm;
 int x1,y1,x2,y2,sx,sy;x,y;
 initgraph(&gd,&gm,"..\\BGI");
 cout<<"Enter co-ordinates of Triangle";
 ~~cin>>x1>>y1>>x2>>y2>>x3>>y3;~~
 cin>>x>>y>>x1>>y1>>x2>>y2;
 line(x,y,x1,y1);
 line(x1,y1,x2,y2);
 line(x2,y2,x,y);
 cout<<"Enter the scaling factors x & y";
 cin>>sx>>sy;
 x=x*sx;
 x1=x1*sx;
 x2=x2*sx;
 y=y*sy;
 y1=y1*sy;
 y2=y2*sy;
 setcolor(RED);
 line(x,y,x1,y1);
 line(x1,y1,x2,y2);
 line(x2,y2,x,y);
 closegraph();
```

(4) Program to do a square & rotate it 45 degree

```
#include <conio.h>
#include <iostream.h>
#include <graphics.h>
#include <math.h>
#define PI 3.14159
Void main(){
  int gd=DETECT,gm;
  initgraph(&gd,&gm,"..\\BGI");
  float x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6,x7,y7,
      x8,y8,rad;
  Cout<<"Enter the co-ordinates of the square";
  Cin>>x1>>y1>>x2>>y2>>x3>>y3>>x4>>y4;
  int arr[]={x1,y1,x2,y2,x3,y3,x4,y4,x1,y1};
  drawpoly(5,arr);
  rad = PI/4;
  x5 = x1*cos(rad) - y1*sin(rad);
  y5 = x1*sin(rad) + y1*cos(rad);
  x6 = x2*cos(rad) - y2*sin(rad);
  y6 = y2*sin(rad) + y2*cos(rad);
  x7 = x3*cos(rad) - x3*sin(rad);
  y7 = y3*sin(rad) + y3*cos(rad);
  x8 = x4*cos(rad) - x4*sin(rad);
  y8 = y4*sin(rad) + y4*cos(rad);
  int arr1[]={x5,y5,x6,y6,x7,y7,x8,y8};
  Setcolor(RED);
  drawpoly(5,arr1);
  getch();
  closegraph();
```

(5) WAP to do a Composite Translation:

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>
Void trans(int *x1, int *y1, int *x2, int *y2, int *x3, int *y3,
int *tx1, int *ty1, int *tx2, int *ty2){
*x1 = *x1 + *tx1 + *tx2;
*x2 = *x2 + *tx1 + *tx2;
*x3 = *x3 + *tx1 + *tx2;
*y1 = *y1 + *ty1 + *ty2;
*y2 = *y2 + *ty1 + *ty2;
*y3 = *y3 + *ty1 + *ty2;
}
void main(){
int gd=DETECT, gm;
initgraph(&gd, &gm, "..\\BGI");
int x1=100, y1=100, x2=150, y2=50, x3=200, y3=100,
  tx1, ty1, tx2, ty2;
cout<<" Enter Translation factor";
cin>> tx1 >> tx2 >> ty1 >> ty2;
line(x1, y1, x2, y2);
line(x2, y2, x3, y3);
line(x3, y3, x1, y1);
trans(&x1, &y1, &x2, &y2, &y3, &y3, &tx1, &ty1, &tx2,
        &ty2);
 line(x1, y1, x2, y2);
 line(x2, y2, x3, y3);
 line(x3, y3, x1, y1);
 getch();
 closegraph();
}
```

(6) WAP to do Composite Scaling:

```
#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
#include <graphics.h>
Void main(){
  int gd=DETECT, gm, x, y, r;
  initgraph(&gd, &gm, "..\\BGI");
  x= getmaxx()/2;
  y= getmaxy()/2;
  r=0;
  while(!kbhit()){
    setcolor(BLACK);
    circle(x,y,r);
    r=r+1;
    setcolor(WHITE);
    circle(x,y,r);
    if(x+r>=getmaxx || y-r>=getmaxy())
    break;
  }
  getch();
}
```

(7) WAP to draw Triangle & Reflect along x, y & x=y axis.

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>
Void main(){
  int gd= DETECT, gm;
  int x1, y1, x2, y2, x3, y3, c;
  initgraph(&gd, &gm, "..\\BGI");
  Cout << "Enter The Co-ordinates";
  Cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3;
```

(6) DDA Line Drawing Algo.

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>
Void main(){
int gd= DETECT,gm;
float x1,y1,x2,y2,dx,dy, xinc, yinc, ;
initgraph(&gd,&gm,"..\\BGI");
Cout<<" Enter the Starting point";
Cin >> x1 >> y1;
Cout<<" Enter the End point";
Cin >> x2 >> y2;
dx = x2 - x1;
dy = y2-y1;
if (dy>dx){
    steps= dx;
}
else{
    steps = dy;
}
xinc = dx/steps;
yinc = dy/steps;
for (i=0, i<=steps, i++){
    putpixel (x1, y1, 2);
    x = x1+xinc;
    y= y1+yinc;
}
  getch();
  closegraph();
}
```

(9) Bresenham Line Drawing Algo

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
Void main(){
int gd=DETECT, gm;
float x1, y1, x2, y2, d, dx, dy;
initgraph(&gd, &gm, "..\\BGI");
Cout<<" Enter the co-ordinates of the line.
Cin>>x1>>y1>>x2>>y2
dx=x2-x1;
dy=y2-y1;
                → d=2*(dy-dx)
x=x1;
y=y1;
While(x<x2){
    putpixel(x, y, 4);
    if(d>=0){
        x++;
        y++;
        d=d+2*(dy-dx)
    }
    else{
        x++
        d=d+2*dy
    }
}
getch();
closegraph();
}
```

(10) DDA Circle drawing Algo:

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>
#include <math.h>
Void main(){
int gd=DETECT, gm;
int x, y, xc, yc, t, s, r;
initgraph (&gd, &gm, "..\\BGI")
Cout<<" Enter the co-ordinates of center "
Cin>>xc>>yc
Cout<<" Enter the radius "
Cin>>r;
for (t=0, i<=(2*3.14), t+=0.01)
{
  x = xc + r* Cos(t);
  y = yc + r* Sin(t);
  putpixel (x, y, 3);
}
  getch();
  closegraph();
}
```

(11) Bresenham's Circle Drawing Algo.

```
# include <iostream.h>
# include <conio.h>
# include <graphics.h>
Void plot (int a, int b, int xc, int yc)
{
putpixel (xc+a, yc+b, 1);
putpixel (xc+b, yc+a, 2);
putpixel (xc-a, yc+b, 3);
putpixel (xc+a, yc-b, 4);
putpixel (xc-b, yc+a, 5);
putpixel (xc+b, yc-a, 6);
putpixel (xc-a, yc-b, 7);
putpixel (xc-b, yc-a, 8);
}
Void main(){
int gd = DETECT, gm;
float xc, yc, r, p, x, y;
initgraph (&gd, &gm, "..\\BGI");
Cout << " Enter the Center Co-ordinates & radius.. ")
cin >> xc >> yc >> r;
x=0
y=r
p=1-r;
While (x<y){
    if (p<0){
        x++
        plot (x, y, xc, yc)
        p = p + 2*x+1;
}
```