Mayank Joshi BCA 'B'
6th Semester.
Roll No - 1121083 (04).
Subject - Computer Graphics.

Question 2.

Algorithm :

1. Start.
2. Initialize the graphics mode.
3. Construct a 2D object (use Drawpoly())
   eg (x, y)
4. A. Translation
   a. Get the translation value tx, ty.
   b. Move the 2d object with tx, ty
      ($x' = x + tx$, $y' = y + ty$)
   c. Plot (x', y')
5. B. Scaling
   a. Get the scaling value Sx, Sy.
   b. Resize the object with Sx, Sy.
      ($x' = x * Sx$, $y' = y * Sy$)
   c. Plot (x', y')
6. C. Rotation
   a. Get rotation angle
   b. Rotate the object by angle ϕ
      $x' = x \cos\phi - y \sin\phi$ $\quad y' = x \sin\phi - y \cos\phi$
   c. Plot (x', y')

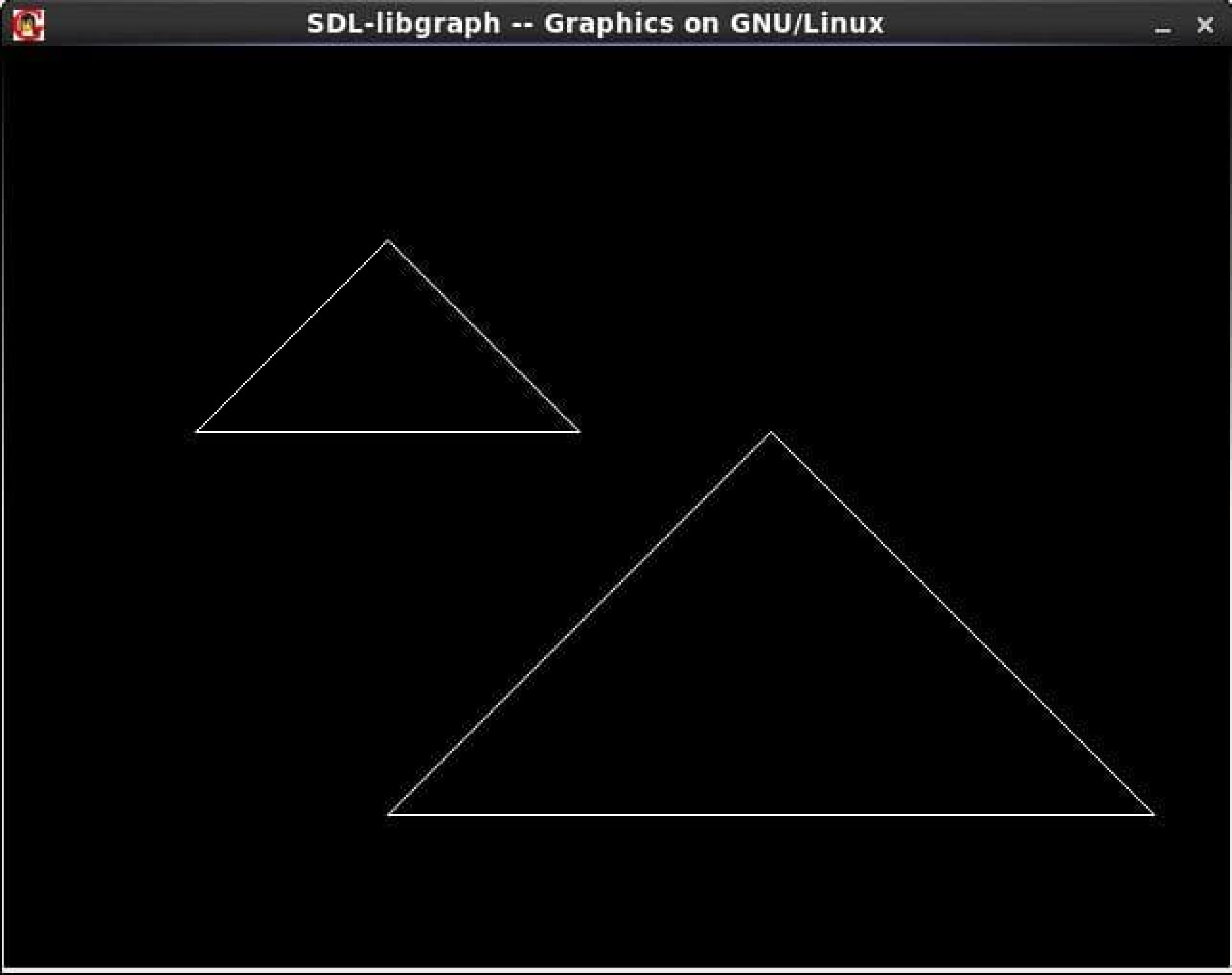
PROGRAM :

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

 void main ()
  {
    int gm ;
    int gd = DETECT ;
    int x1 , x2 , x3 , y1 , y2 , y3 , nx1 , nx2 , nx3 , ny1 ,
        ny2 , ny3 , c ;
    int sx , sy , xt , yt , r ;
    float t ;
    initgraph (& gd , & gm , "c:\tc\bg:") ;
    printf ("\t Program for basic transactions") ;
    printf ("\n\t Enter the points of triangle") ;
    setcolor (1) ;

  scanf ("%d %d %d %d %d %d", & x1 , & y1 , & x2 , & y2 , & x3 , & y3 ) ;
```

```
line (x1, y1, x2, y2);
line (x2, y2, x3, y3);
line (x3, y3, x1, y1);
getch ();
printf ("\n 1. Translation \n 2. Rotation \n 3. Scaling
        \n 4. exit");
printf ("Enter your choice:");
scanf ("%d", &c);
switch (c)
 {
  case 1:
   printf ("\n Enter the translation factor");
   scanf ("%d %d", &xt, &yt);
   nx1 = x1 + xt;
   ny1 = y1 + yt;
   nx2 = x2 + xt;
   ny2 = y2 + yt;
   nx3 = x3 + xt;
   ny3 = y3 + yt;
   line (nx1, ny1, nx2, ny2);
   line (nx2, ny2, nx3, ny3);
```

```
line (nx3, ny3, nx1, ny1);
getch();

case 2:
printf("\n Enter the angle of rotation");
  scanf("%d", &r);
  t = 3.14 * r / 180;
  nx1 = abs(x1 * cos(t) - y1 * sin(t));
  ny1 = abs(x1 * sin(t) + y1 * cos(t));
  nx2 = abs(x2 * cos(t) - y2 * sin(t));
  ny2 = abs(x2 * sin(t) + y2 * cos(t));
  nx3 = abs(x3 * cos(t) - y3 * sin(t));
  ny3 = abs(x3 * sin(t) + y3 * cos(t));

      line(nx1, ny1, nx2, ny2);
      line(nx2, ny2, nx3, ny3);
      line(nx3, ny3, nx1, ny1);
      getch();

case 3:
  printf("\n Enter the scaling factor");
```

```
scanf("%d %d", &sx, &sy);
nx1 = x1 * sx;
ny1 = y2 * sy;
nx2 = x2 * sx;
ny2 = y2 * sy;
nx3 = x3 * sx;
ny3 = y3 * sy;
line(nx1, ny1, nx2, ny2);
line(nx2, ny2, nx3, ny3);
line(nx3, ny3, nx1, ny1);
getch()

case 4:
    break;

default:
    printf("Enter the correct choice");
}
closegraph();
}
```


SDL-libgraph -- Graphics on GNU/Linux

Question-3.

Algorithm:

Step 1 - Start.

Step 2 - Declare p, q, x, y, r, d variables. P, q are coordinates of centre of circle; r is the radius.

step - 3 - Enter the value of r.

step 4 - Calculate d = 3 - 2r.

step 5 - Initialize x = 0 & nbsy = r.

step 6 - Check if the whole circle is scan converted
if x >= y
stop.

step 7 - Plot eight points by using concepts of eight-way symmetry. The centre is at (p, q).
Current active pixel is (x, y)
putpixel (x+p, y+q)
putpixel (y+p, x+q)
putpixel (-y+p, x+q)
putpixel (-x+p, y+q)

putpixel (-x+p, -y+q)

putpixel (-y+p, -x+q)

putpixel (y+p, -x+q)

putpixel (x+p, -y-q)

PROGRAM:

```
#include <graphics.h>
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>
void EightWaySymmetricPlot(int xc, int yc, intx, int y)
{
 putpixel(x+xc, y+yc, RED);
 putpixel(x+xc, -y+yc, YELLOW);
 putpixel(-x+xc, -y+yc, GREEN);
 putpixel(-x+xc, y+yc, YELLOW);
 putpixel(y+xc, x+yc, 12);
 putpixel(y+xc, -x+yc, 14);
```

```
putpixel(-y+xc,-x+yc,15);
putpixel(-y+xc,x+yc,6);
}
void Bresenhamcircle(int xc, int yc, int r)
{
int x=0, y=r, d=3-(2*r);
EightWaySymmetricPlot(xc,yc,x,y);
while(x<=y)
{
if(d<=0)
{
d=d+(4*x)+6;
}
else
{
d=d+(4*x)-(4*y)+10)
y=y-1;
}
x=x+1;
EightWaySymmetricPlot(xc,yc,x,y);
}
}
```

```
int main (void)
{
 int xc, yc, r, gdriver = DETECT, gmode, errorcode;
 initgraph (& gdriver, & gmode, "C:\\TURBOC3\\BGI");
 errorcode = graphresult ();
 if (errorcode != grok)
   {
   printf ("Graphics error: %s \n", grapherrormsg
                                    (errorcode));
   printf ("Press any key to halt:");
   getch ();
   exit (1);
  }
 printf ("Enter the values of xc & yc:");
 scanf ("%d %d", & xc, & yc);
 printf ("Enter the value of radius:");
 scanf ("%d", & r);
  BresenhamCircle (xc, yc, r);
 getch ();
 closegraph ();
 return 0;
}
```

*** Mid-Point Subdivision algorithm of circle ***
Enter the value of Xc 400
Enter the value of Yc 140
Enter the Radius of circle 97