LAB MANUAL
FOR
IF LAB

PROGRAM TO SHOW THE USE OF VARIOUS INBUILT FUNCTIONS OF GRAPHICS

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


void main()
{
        int gdriver=DETECT,gmode,errorcode;
        int poly[6],i,xmax,ymax;
        initgraph(&gdriver,&gmode,"c:\\tc\\bgi");

        cleardevice();
        setcolor(CYAN);
        rectangle(100,50,500,80);

        xmax=getmaxx();
        ymax=getmaxy();
        printf("Max x is %d",xmax);
        printf("\tMax y is %d",ymax);

        arc(200,400,0,90,50);

        bar(10,20,100,40);

        setcolor(RED);
        bar3d(200,200,100,150,60,20);

        circle(500,200,50);

        settextjustify(CENTER_TEXT,CENTER_TEXT);
        settextstyle(GOTHIC_FONT, HORIZ_DIR, 4);
        outtextxy(100,100,"Vikas Kapoor");

        ellipse(300,300,0,360,75,20);
        setfillstyle(CLOSE_DOT_FILL,2);
        fillellipse(400,400,60,30);
        setfillstyle(LINE_FILL,4);
        sector(200,250,0,360,75,20);
```

```
        getch();
        closegraph();
}
```

PROGRAM TO CREATE A HUT

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


void main()
{
        int gdriver=DETECT,gmode,errorcode;
        clrscr();
        initgraph(&gdriver,&gmode,"c:\\tc\\bgi");

        line(100,20,20,100);
        line(100,20,180,100);

        line(20,100,440,100);

        line(100,20,360,20);
        line(360,20,440,100);

        rectangle(20,100,440,350);

        line(180,100,180,350);

        setcolor(BROWN);
        rectangle(50,200,150,350);

        setcolor(CYAN);
        rectangle(250,175,365,275);
        line(250,175,365,275);
        line(365,175,250,275);

        setcolor(WHITE);
        fillellipse(100,70,10,10);


        getch();
        closegraph();
}
```

PROGRAM TO IMPLEMENT LINE DDA ALGORITHM

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#define ROUND(a)   ((int) (a+0.5))


void lineDDA(int xa,int ya,int xb,int yb)
{
        int dx=xb-xa;
        int dy=yb-ya;
        int steps,k;
        float xIncr,yIncr,x=xa,y=ya;

        if(abs(dx)>abs(dy))
        {
                steps=abs(dx);
        }
        else
        {
                steps=abs(dy);
        }

        xIncr=dx/(float)steps;
        yIncr=dy/(float)steps;

        putpixel(ROUND(x),ROUND(y),2);

        for(k=0;k<=steps;k++)
        {
                x+=xIncr;
                y+=yIncr;

                putpixel(ROUND(x),ROUND(y),2);
        }
}

void main()
{
        int gdriver=DETECT,gmode,errorcode;
```

```
        int a,b,c,d;
        initgraph(&gdriver,&gmode,"c:\\tc\\bgi");


        printf("\nEnter the coordinates :- ");
        scanf("%d%d",&a,&b);
        printf("\n\nEnter the coordinates of second point :- ");
        scanf("%d%d",&c,&d);

//      clrscr();

        lineDDA(a,b,c,d);

//      lineDDA(400,450,200,160);

        getch();
        closegraph();
}
```

PROGRAM TO IMPLEMENT LINE BRESENHEM'S (MID POINT LINE) ALGORITHM

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

lineBres(int xa,int ya,int xb,int yb)
{
        int dx=abs(xa-xb), dy=abs(ya-yb);
        int d=(2*dy)-dx;
        int incrE=2*dy, incrNE=2*(dy-dx);
        int x=xa, y=ya;

        putpixel(x,y,2);

        while(x<xb)
        {
                if(d<=0)
                {
                        d+=incrE;
                        x++;
                }
                else
                {
                        d+=incrNE;
                        x++;
                        y++;
                }

                putpixel(x, y, 2);
        }
        return 0;
}


void main()
{
        int gdriver=DETECT,gmode,errorcode;
        int a,b,c,d;
        initgraph(&gdriver,&gmode,"c:\\tc\\bgi");
```

```
        printf("\nEnter the coordinates :- ");
        scanf("%d%d",&a,&b);
        printf("\n\nEnter the coordinates of second point :- ");
        scanf("%d%d",&c,&d);

        lineBres(a,b,c,d);

        getch();
        closegraph();
}
```

PROGRAM TO IMPLEMENT MIDPOINT CIRCLE ALGORITHM

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

void CirclePlot(int ,int ,int ,int);




void CirclePt(xc, yc, r)
{
        int x=0,y=r,p=1-r;
        while(x<y)
        {
                x++;
                if(p<0)
                {
                        p+=2*x;
                }
                else
                {
                        y--;
                        p+=2*(x-y)+1;
                }
                CirclePlot(xc,yc,x,y);
        }
}
void CirclePlot(xc ,yc ,x ,y)
{
        putpixel(xc+x, yc+y, 1);
        putpixel(xc+y, yc+x, 2);
        putpixel(xc-y, yc+x, 3);
        putpixel(xc-x, yc+y, 4);
        putpixel(xc-x, yc-y, 5);
        putpixel(xc-y, yc-x, 6);
        putpixel(xc+y, yc-x, 7);
        putpixel(xc+x, yc-y, 8);
}
```

```
void main()
{
        int gd=DETECT,gm;
        int xc, yc, r;
        clrscr();
        initgraph(&gd, &gm, "c:\\tc\\bgi");

        printf("Enter the Center coordinates of the circle : ");
        scanf("%d%d",&xc,&yc);
        printf("\n\nEnter the Radius of the circle : ");
        scanf("%d",&r);
        CirclePt(xc,yc,r);

        getch();
        closegraph();
}
```

PROGRAM TO IMPLEMENT MIDPOINT ELLIPSE ALGORITHM

```
#include<stdio.h>
#include<dos.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
  void ellipsepoints(int,int);
  void completeellipse(int r,int g,int u,int v)
  {
  float s,k,e,f;
  double p2;
  s=r;k=g;
  e=(pow((s+0.5),2));
  f=(pow((k-1),2));
  p2=(u*e)+(v*f)-(u*v);
  ellipsepoints(s,k);
  while(k>=0)
  {
  if(p2>0)
  p2=p2+v-2*v*s;
  else
  {
  p2=p2+(2*u*(s+1))-(2*v*(k-1))+v;
  s++;
  }
  k--;
  ellipsepoints(s,k);
  }
  }
  void main()
  {
  int gdriver=DETECT,gmode;
  int a,b,x,y;
  long p1,u,v;
  initgraph(&gdriver,&gmode,"c:\\tc\\bgi");
   printf( "\nenter the length of major axis");
  scanf("%d",&a);
  printf( "\nenter the length of minor axis");
  scanf("%d",&b);

  x=0;
```

```
y=b;
u=pow(b,2);
v=pow(a,2);
p1=u-(v*b)+(.25*v);
ellipsepoints(x,y);
while((2*(u*x))<=(2*(v*y)))
{
x++;
if(p1<0)
p1=(p1+(2*u*x)+u);
else
{
p1=(p1+(2*u*x)-(2*v*y)+u);
y--;
};
ellipsepoints(x,y);
}
completeellipse(x,y,u,v);
getch();
closegraph();
}
void ellipsepoints(int x,int y)
{
putpixel(x+200,y+200,2);
putpixel(-x+200,y+200,1);
putpixel(x+200,-y+200,3);
putpixel(-x+200,-y+200,7);
}
```

PROGRAM TO FILL A POLYGON

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

void main()
{
        int gd=DETECT,gm;
        int i,j;
        initgraph(&gd,&gm,"c:\\tc\\bgi");

        rectangle(70,90,570,390);
        line(320,90,320,390);
        line(70,240,570,240);

        for(i=70;i<=320;i=i+5)
        {
                for(j=90;j<=240;j+=5)
                {
                        putpixel(i,j,1);
                }
                delay(100);
        }

        for(i=320;i<=570;i=i+5)
        {
                for(j=90;j<=240;j+=3)
                {
                        putpixel(i,j,2);
                }
                delay(100);
        }
        for(i=70;i<=320;i++)
        {
                for(j=240;j<=390;j++)
                {
                        putpixel(i,j,3);
                }
                delay(100);
        }

        for(i=320;i<=570;i++)
```

```
        {
                for(j=240;j<=390;j+=10)
                {
                        putpixel(i,j,4);
                }
                delay(100);
        }

        getch();
        closegraph();
}
```

PROGRAM TO PERFORM REFLECTION OF A POINT IN THE SCREEN

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

void main()
{
        int gd=DETECT,gm;
        int xc,yc,maxx,maxy;
        int xr,yr,x1,y1;
        initgraph(&gd,&gm,"c:\\tc\\bgi");

        printf("Enter the coordinates of the Origin in the Screen : ");
        scanf("%d%d",&xc,&yc);

        maxx=getmaxx();
        maxy=getmaxy();

        cleardevice();
        line(xc,0,xc,maxy);
        line(0,yc,maxx,yc);


        printf("Enter the coordinates of the point : ");
        scanf("%d%d",&xr,&yr);

        putpixel(xr,yr,2);
        outtextxy(xr+1,yr+1,"pnt");

//Reflection along x-axis

        y1=yr-yc;
        putpixel(xr,(yc-y1),3);
        outtextxy(xr+1,(yc-y1)+1,"in x");

//Reflection along y-axis

        x1=xr-xc;
        putpixel((xc-x1),yr,4);
```

```
        outtextxy((xc-x1)+1,yr+1,"in y");

//Reflection along origin

        putpixel((xc-x1),(yc-y1),1);
        outtextxy((xc-x1)+1,(yc-y1)+1,"in origin");

        getch();
        closegraph();
}
```

PROGRAM TO PERFORM SHEARING OF A RECTANGLE

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

void main()
{
        int gd=DETECT,gm;
        int l,t,r,b,sx,sy;
        int x1,y1,x2,y2,x3,y3,x4,y4;
        initgraph(&gd,&gm,"c:\\tc\\bgi");

        printf("Enter the coordinates of rectangle : ");
        scanf("%d%d%d%d",&l,&t,&r,&b);

//      cleardevice();
        rectangle(l,t,r,b);


//Shear along x-axis
        printf("Enter the x-direction shear factor : ");
        scanf("%d",&sx);
        x1=l+sx*t-sx*t;
        y1=t;

        x2=r+sx*t-sx*t;
        y2=t;

        x3=l+sx*b-sx*t;
        y3=b;

        x4=r+sx*b-sx*t;
        y4=b;

        line(x1,y1,x2,y2);
        line(x2,y2,x4,y4);
        line(x4,y4,x3,y3);
        line(x1,y1,x3,y3);

        x1=y1=x2=y2=x3=y3=x4=y4=0;

//Shear along y-axis
```

```
//      cleardevice();
  //    rectangle(l,t,r,b);
        printf("Enter the y-direction shear factor : ");
        scanf("%d",&sy);
        x1=l;
        y1=t+sy*l-sy*l;

        x2=r;
        y2=t+sy*r-sy*l;

        x3=r;
        y3=b+sy*r-sy*l;

        x4=l;
        y4=b+sy*l-sy*l;

        line(x1,y1,x2,y2);
        line(x2,y2,x3,y3);
        line(x4,y4,x3,y3);
        line(x1,y1,x4,y4);



        getch();
        closegraph();
}
```

PROGRAM TO ROTATE A CIRCLE AROUND ANOTHER CIRCLE

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
#include<dos.h>
void main()
{
clrscr();
int gm,gd=DETECT;
initgraph(&gd,&gm,"c:\\tc\\bgi");
int x1,x2,x3,y1,y2,y3;
float t=0.0;
cout<<"  output  "<<"\n" ;
cout<<"\nEnter coordinates of first point and second point ";
cin>>x1>>y1;
circle(x1,y1,100);
x3=100;
y3=100;
while(!kbhit())
{
t=t-0.1;
x2=x1+(x3*cos(t)+y3*sin(t));
y2=y1+(x3*sin(t)-y3*cos(t));
circle(x2,y2,40);
delay(100);
cleardevice();
circle(x1,y1,100);
}
getch();
closegraph();
}
```

PROGRAM TO PERFORM VARIOUS TRANSFORMATIONS

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

void Scaling();
void Translation();
void Rotation();

void main()
{
        int gd=DETECT,gm;
        int ch;
        clrscr();
        initgraph(&gd,&gm,"d:\\tc\\bgi");

        printf("\t|||||||||TRANSFORMATIONS|||||||||");
        printf("\n\n\t\t1. Scaling\n\t\t2. Translation\n\t\t3. Rotation");
        printf("\n\n\t\tEnter the choice : ");
        scanf("%d",&ch);
        //ch=getch();

        switch(ch)
        {
                case 1:
                        Scaling();
                        break;

                case 2:
                        Translation();
                        break;

                case 3:
                        Rotation();
                        break;

                default:
                        printf("Wrong Choice");
        }

        getch();
        closegraph();
}
```

```
void Scaling()
{
        int gd=DETECT,gm;
        int x1,y1,x2,y2,xa,ya,xb,yb,sx,sy,chs;
        clrscr();
        initgraph(&gd,&gm,"d:\\tc\\bgi");

        printf("\t||||||||||SCALING||||||||||");
        printf("\n\n\t\t1. Point Scaling\n\t\t2. Line Scaling");
        printf("\n\n\tEnter the choice : ");
        scanf("%d",&chs);
        //chs=getch();

        switch(chs)
        {
                case 1:
                        printf("\n\n\tEnter the co-ordinates of the Point : ");
                        scanf("%d%d",&x1,&y1);

                        putpixel(x1,y1,1);

                        printf("\n\n\tEnter the x direction Scaling factor (Sx) : ");
                        scanf("%d",&sx);

                        printf("\n\n\tEnter the y direction Scaling factor (Sy) : ");
                        scanf("%d",&sy);

                        xa=x1*sx;
                        ya=y1*sy;

                        putpixel(xa,ya,4);

                        break;

                case 2:
                        printf("\n\n\tEnter the co-ordinates of the Line : ");
                        scanf("%d%d%d%d",&x1,&y1,&x2,&y2);

                        setcolor(1);
                        line(x1,y1,x2,y2);

                        printf("\n\n\tEnter the x direction Scaling factor (Sx) : ");
                        scanf("%d",&sx);
```

```
                printf("\n\n\tEnter the y direction Scaling factor (Sy) : ");
                scanf("%d",&sy);

                xa=x1*sx;
                ya=y1*sy;
                xb=x2*sx;
                yb=y2*sy;

                setcolor(4);
                line(xa,ya,xb,yb);

                break;

        default:
                printf("Wrong Choice");
    }
}

void Translation()
{
    int gd=DETECT,gm;
    int x1,y1,x2,y2,xa,ya,xb,yb,tx,ty,cht;
    clrscr();
    initgraph(&gd,&gm,"d:\\tc\\bgi");

    printf("\t|||||||||TRANSLATION|||||||||");
    printf("\n\n\t\t1. Point Translation\n\t\t2. Line Translation");
    printf("\n\n\tEnter the choice : ");
    scanf("%d",&cht);
    //cht=getch();

    switch(cht)
    {
        case 1:
                printf("\n\n\tEnter the co-ordinates of the Point : ");
                scanf("%d%d",&x1,&y1);

                putpixel(x1,y1,1);

                printf("\n\n\tEnter the x direction Translation factor (tx) : ");
                scanf("%d",&tx);

                printf("\n\n\tEnter the y direction Translation factor (ty) : ");
                scanf("%d",&ty);
```

```
                xa=x1+tx;
                ya=y1+ty;

                putpixel(xa,ya,4);

                break;

        case 2:
                printf("\n\n\tEnter the co-ordinates of the Line : ");
                scanf("%d%d%d%d",&x1,&y1,&x2,&y2);

                setcolor(1);
                line(x1,y1,x2,y2);

                printf("\n\n\tEnter the x direction Translarion factor (tx) : ");
                scanf("%d",&tx);

                printf("\n\n\tEnter the y direction Translation factor (ty) : ");
                scanf("%d",&ty);

                xa=x1+tx;
                ya=y1+ty;
                xb=x2+tx;
                yb=y2+ty;

                setcolor(4);
                line(xa,ya,xb,yb);

                break;

        default:
                printf("Wrong Choice");
        }
}

void Rotation()
{
        int gd=DETECT,gm;
        float x1,y1,x2,y2,xa,ya,xb,yb;
        int q,chr,i;
        double ang;
        clrscr();
        initgraph(&gd,&gm,"d:\\tc\\bgi");

        printf("\t|||||||||ROTATION|||||||||");
```

```
printf("\n\n\t\t1. Point Translation\n\t\t2. Line Translation");
printf("\n\n\tEnter the choice : ");
scanf("%d",&chr);
//chr=getch();

switch(chr)
{
        case 1:
                printf("\n\n\tEnter the co-ordinates of the Point : ");
                scanf("%f%f",&x1,&y1);

                putpixel(x1,y1,1);

                printf("\n\n\tEnter the angle : ");
                scanf("%d",&q);

                ang=(q*3.14)/180;

                for(i=0;i<50;i++)
                {
                        xa=x1*cos(ang)-y1*sin(ang);
                        ya=y1*sin(ang)+x1*cos(ang);
                        x1=xa;
                        y1=ya;

                        putpixel(xa,ya,4);
                        delay(100);
                }

                break;

        case 2:
                printf("\n\n\tEnter the co-ordinates of the Line : ");
                scanf("%f%f%f%f",&x1,&y1,&x2,&y2);

                setcolor(1);
                line(x1,y1,x2,y2);

                printf("\n\n\tEnter the angle : ");
                scanf("%d",&q);

                ang=(q*3.14)/180;

                for(i=0;i<50;i++)
                {
```

```
                        xa=x1*cos(ang)-y1*sin(ang);
                        ya=y1*sin(ang)+x1*cos(ang);
                        xb=x2*cos(ang)-y2*sin(ang);
                        yb=y2*sin(ang)+x2*cos(ang);

                        setcolor(4);
                        line(xa,ya,xb,yb);
                        delay(100);
                }

                break;

        default:
                printf("Wrong Choice");
    }
}
```