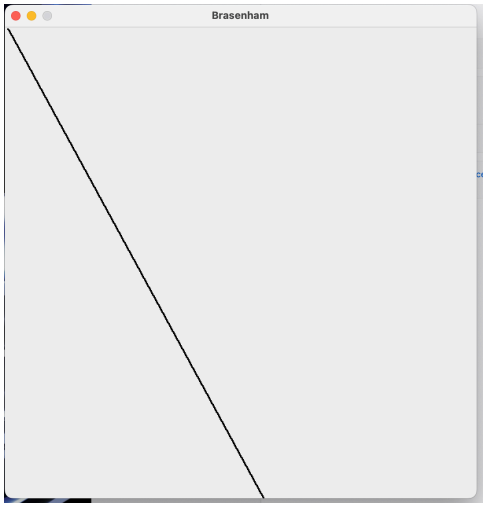
Computer Graphics Assignment

1.Brasenham Circle:

```
import time
from graphics import *
import numpy as np
import math

def draw_pixels(x,y,win):
    pt=Point(cx+x,cy+y)
    pt.draw(win)
    pt=Point(cx-x,cy+y)
    pt.draw(win)
    pt=Point(cx+x,cy-y)
    pt.draw(win)
    pt=Point(cx-x,cy-y)
    pt.draw(win)
    pt=Point(cy+y,cx+x)
    pt.draw(win)
    pt=Point(cy-y,cx+x)
    pt.draw(win)
    pt=Point(cy+y,cx-x)
    pt.draw(win)
    pt=Point(cy-y,cx-x)
    pt.draw(win)
    time.sleep(0.01)

def brasenham_circle(r):
    window=GraphWin("Brasenham Circle Drawing Algorithm",2*r,2*r)
    x=0
    y=r
    draw_pixels(x,y,window)
    d=3-2*y
    while int(x)!=int(y) and int(x-1)!=int(y):
        if d<0: #select N
            d = d+4*x+6
            x+=1
        else:   #select S
            d = d + 4*(x-y)+10
            x = x + 1
            y-=1
        draw_pixels(x,y,window)
    window.getMouse()
    window.close()

if __name__=='__main__':
    r=int(input("Enter radius of the circle : "))
    cx=r
    cy=r
    brasenham_circle(r)
```

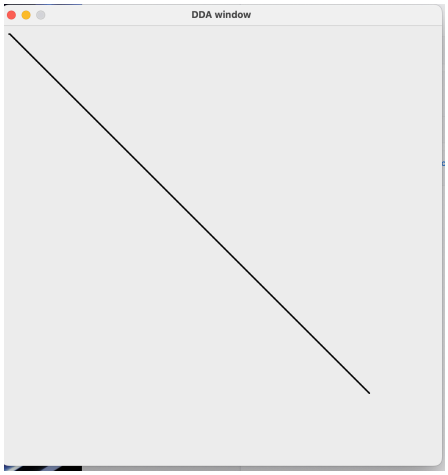
2.Brasemham Line:

```
import time
from graphics import *
def brasenham(x1,y1,x2,y2):
    dx=abs(x2-x1)
    dy=abs(y2-y1)
    p=2*dy-dx
    window=GraphWin("Brasenham",600,600)
    pt=Point(x1,y1)
    pt.draw(window)
    while x1!=x2 or y1!=y2:
        if dx>dy:
            if p<0:
                x1+=1
                p+=2*dy
            else:
                x1+=1
                y1+=1
                p+=2*dy-2*dx
        else:
            if p<0:
                y1+=1
                p+=2*dx
            else:
                x1+=1
                y1+=1
                p+=2*dx-2*dy
        pt=Point(x1,y1)
        pt.draw(window)
        time.sleep(0.01)
        if x1>x2 or y1>y2:
            break
    window.getMouse()
    window.close()

if __name__=='__main__':
    x1=int(input("Enter x1 : "))
    y1=int(input("Enter y1 : "))
    x2=int(input("Enter x2 : "))
    y2=int(input("Enter y2 : "))
    brasenham(x1,y1,x2,y2)
```



3. DDA Line:

```
import time
from graphics import *
def dda(x1,y1,x2,y2):
    xtemp=x1
    ytemp=y1
    win=GraphWin("DDA window",600,600)
    pt=Point(abs(x1),abs(y1))
    pt.draw(win)
    dx=x2-x1
    dy=y2-y1
    steps=max(abs(dx),abs(dy))
    xtemp=x1
    ytemp=y1
    xinc=dx/steps
    yinc=dy/steps
    for i in range(steps):
        xtemp+=xinc
        ytemp+=yinc
        x1=int(xtemp)
        y1=int(ytemp)
        if i==steps-1:
            x1=int(x2)
            y1=int(y2)
        pt=Point(x1,y1)
        pt.draw(win)
        time.sleep(0.01)
    win.getMouse()
    win.close()


if __name__=='__main__':
    x1=int(input("Enter x1 : "))
    y1=int(input("Enter y1 : "))
    x2=int(input("Enter x2 : "))
    y2=int(input("Enter y2 : "))
    dda(x1,y1,x2,y2)
```



4.Mid Point line Drawing :

Code:

```
from graphics import *
import time
def Midpoint_line(x1,y1,x2,y2):
    win=GraphWin("Bresenham line drawing",600,600)
    dx=x2-x1
    dy=y2-y1
    m=dy/dx
    d0=dy-(dx/2)
    while(x1<=x2 and y1<=y2):
        # m<1
        if(d0<=0):
            x1=x1+1
            y1=y1
            d0=d0+dy
        else:
            x1=x1+1
            y1=y1+1
            d0=d0+dy-dx
        pt=Point(x1+100,y1+100)
        pt.draw(win)
        time.sleep(0.01)
    win.getMouse()
    win.close()
def main():
    x=int(input("Enter x1: "))
    y=int(input("Enter y1: "))
    xEnd=int(input("Enter x2: "))
    yEnd=int(input("Enter y2: "))
    color="r."
    Midpoint_line(x,y,xEnd,yEnd)
if __name__=="__main__":
    main()
```

Output :

```
Enter x1: 5
Enter y1: 10
Enter x2: 100
Enter y2: 150
PS D:\NOTES\Sem 6\DS Sir\Computer Graphics Practical>
```

5. DDA Circle:

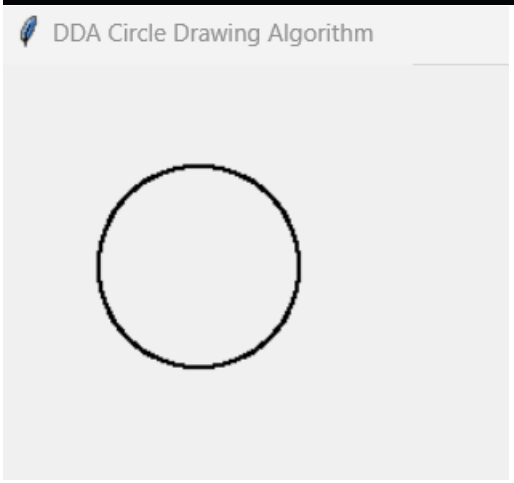
Code:

```
import time
from graphics import *
import numpy as np
import math
cx=300
cy=300
def put_pixel(x,y,win):
    pt=Point(cx+x,cy+y)
    pt.draw(win)
    pt=Point(cx-x,cy+y)
    pt.draw(win)
    pt=Point(cx+x,cy-y)
    pt.draw(win)
    pt=Point(cx-x,cy-y)
    pt.draw(win)
    pt=Point(cy+y,cx+x)
    pt.draw(win)
    pt=Point(cy-y,cx+x)
    pt.draw(win)
    pt=Point(cy+y,cx-x)
    pt.draw(win)
    pt=Point(cy-y,cx-x)
    pt.draw(win)
    time.sleep(0.01)
def dda_circle(r):
    win=GraphWin("DDA Circle Drawing Algorithm",600,600)
    x=r
    y=0
    put_pixel(x,y,win)
    n=math.ceil(np.log2(r))
    e=pow(2,-n)
    while int(x)!=-int(y):
```

```
            x+=e*y
            y-=e*x
            # print(str(x)+"   "+str(y))
            put_pixel(x,y,win)
        win.getMouse()
        win.close()

if __name__=='__main__':
    r=int(input("Enter radius of the circle : "))
    dda_circle(r)
```

Output:

```
Enter radius of the circle : 50
PS D:\NOTES\Sem 6\DS Sir\Computer Graphics Practical>
```



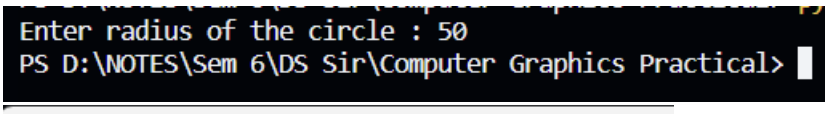
6. Bresenham Circle :

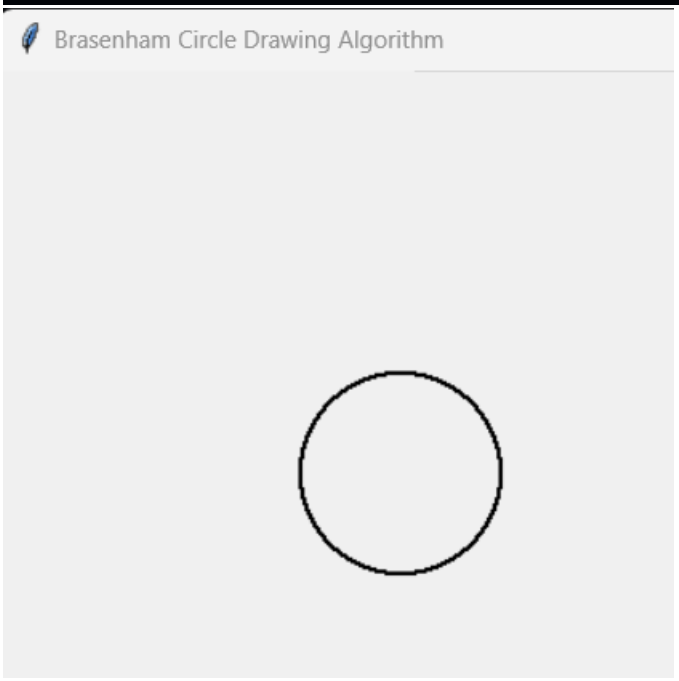
Code:

```
import time
from graphics import *
import numpy as np
import math
cx=200
cy=200
def put_pixel(x,y,win):
    pt=Point(cx+x,cy+y)
    pt.draw(win)
    pt=Point(cx-x,cy+y)
    pt.draw(win)
    pt=Point(cx+x,cy-y)
    pt.draw(win)
    pt=Point(cx-x,cy-y)
    pt.draw(win)
    pt=Point(cy+y,cx+x)
    pt.draw(win)
    pt=Point(cy-y,cx+x)
    pt.draw(win)
    pt=Point(cy+y,cx-x)
    pt.draw(win)
    pt=Point(cy-y,cx-x)
    pt.draw(win)
    time.sleep(0.01)
```

```
def brasenham_circle(r):
    win=GraphWin("Brasenham Circle Drawing Algorithm",600,600)
    x=0
    y=r
    put_pixel(x,y,win)
    d=3-2*y
    while int(x)!=int(y) and int(x-1)!=int(y):
        if d<0: #select N
            d+=4*x+6
            x+=1
        else:   #select S
            d+=4*(x-y)+10
            x+=1
            y-=1
        # print((x,y))
        put_pixel(x,y,win)
    win.getMouse()
    win.close()
if __name__=='__main__':
    r=int(input("Enter radius of the circle : "))
    brasenham_circle(r)
```

Output:





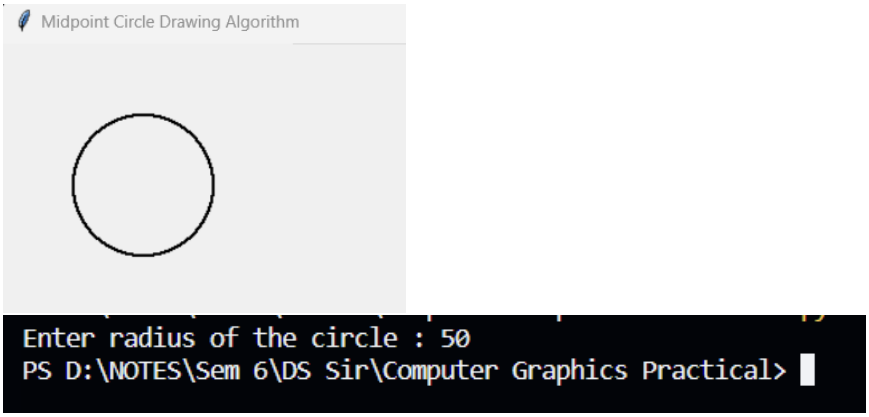
7. Midpoint Circle :

Code:

```
import time
from graphics import *
import numpy as np
import math
cx=100
cy=100
```

```
def put_pixel(x,y,win):
    pt=Point(cx+x,cy+y)
    pt.draw(win)
    pt=Point(cx-x,cy+y)
    pt.draw(win)
    pt=Point(cx+x,cy-y)
    pt.draw(win)
    pt=Point(cx-x,cy-y)
    pt.draw(win)
    pt=Point(cy+y,cx+x)
    pt.draw(win)
    pt=Point(cy-y,cx+x)
    pt.draw(win)
    pt=Point(cy+y,cx-x)
    pt.draw(win)
    pt=Point(cy-y,cx-x)
    pt.draw(win)
    time.sleep(0.01)
def mid_point_circle(r):
    win=GraphWin("Midpoint Circle Drawing Algorithm",600,600)
    x=0
    y=r
    put_pixel(x,y,win)
    d=1-y
    while int(x)!=int(y) and int(x-1)!=int(y):
        if d<0:    #select N
            d+=2*x+3
            x+=1
        else:       #select S
            d+=2*(x-y)+5
            x+=1
            y-=1
        # print((x,y))
        put_pixel(x,y,win)
    win.getMouse()
    win.close()
if __name__=='__main__':
    r=int(input("Enter radius of the circle : "))
    mid_point_circle(r)
```

Output:



8. Polygon Drawing :

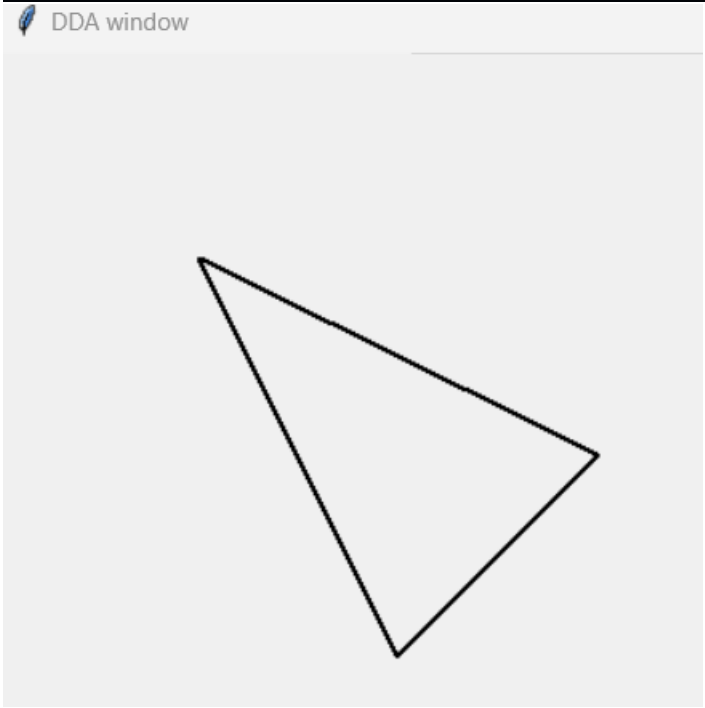
Code:

```
import time
from graphics import *
def dda(x1,y1,x2,y2,win):
    pt=Point(x1,y1)
    pt.draw(win)
    dx=x2-x1
    dy=y2-y1
    steps=max(abs(dx),abs(dy))
    if steps==0:
        return
    xtemp=x1
    ytemp=y1
    xinc=dx/steps
    yinc=dy/steps
    for i in range(steps):
        xtemp+=xinc
        ytemp+=yinc
        x1=int(xtemp)
        y1=int(ytemp)
        if i==steps-1:
            x1=int(x2)
            y1=int(y2)
        pt=Point(x1+100,y1+100)
        pt.draw(win)
        time.sleep(0.001)
def polygon_draw(op,px,py,win):
    n=op[0]
    x1=px[0]
    y1=py[0]
    for i in range(1,n+1):
        x=px[i]
        y=py[i]
        if op[i]==1:
            dda(x1,y1,x,y,win)
            x1=x
            y1=y
if __name__=='__main__':
    px=[]
    py=[]
    op=[]
    n=int(input("Enter number of points : "))
    x1=int(input("Enter initial point x : "))
    y1=int(input("Enter initial point y : "))
    op.append(n)
    px.append(x1)
    py.append(y1)
    for i in range(1,n):
        # p=int(input("1. Line : "))
        x=int(input("Enter next point x : "))
        y=int(input("Enter next point y : "))
        op.append(1)
        px.append(x)
        py.append(y)
```

```
    op.append(1)
    px.append(x1)
    py.append(y1)
    win=GraphWin("DDA window",600,600)
    polygon_draw(op,px,py,win)
    win.getMouse()
    win.close()
```

Output:

```
Enter number of points : 4
Enter initial point x : 1
Enter initial point y : 2
Enter next point x : 100
Enter next point y : 200
Enter next point x : 200
Enter next point y : 100
Enter next point x : 1
Enter next point y : 2
PS D:\NOTES\Sem 6\DS Sir\Computer Graphics Practical>
```



9. Translation :

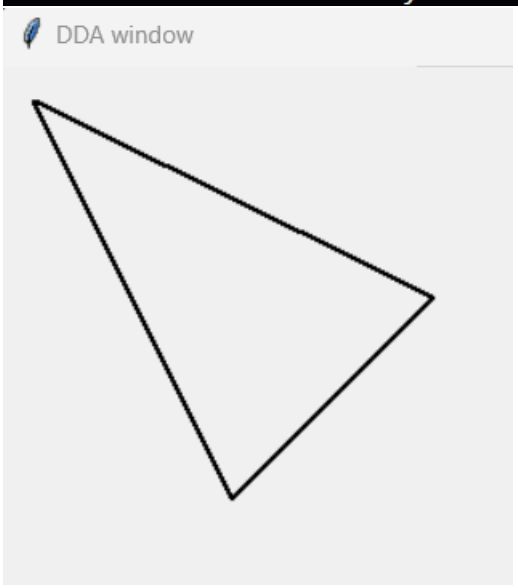
Code:

```
import time
from graphics import *
def translation(x,y,tx,ty):
    return Point(x+tx,y+ty)

def dda(x1,y1,x2,y2,tx,ty,win):
    pt=translation(x1,y1,tx,ty)
    pt.draw(win)
    dx=x2-x1
    dy=y2-y1
    steps=max(abs(dx),abs(dy))
    if steps==0:
        return
```

```
    xtemp=x1
    ytemp=y1
    xinc=dx/steps
    yinc=dy/steps
    for i in range(steps):
        xtemp+=xinc
        ytemp+=yinc
        x1=int(xtemp)
        y1=int(ytemp)
        if i==steps-1:
            x1=int(x2)
            y1=int(y2)
        pt=translation(x1,y1,tx,ty)
        pt.draw(win)
        time.sleep(0.001)


def polygon_draw(op,px,py,tx,ty,win):
    n=op[0]
    x1=px[0]
    y1=py[0]
    for i in range(1,n+1):
        x=px[i]
        y=py[i]
        if op[i]==1:
            dda(x1,y1,x,y,tx,ty,win)
            x1=x
            y1=y

if __name__=='__main__':
    px=[]
    py=[]
    op=[]
    n=int(input("Enter number of points : "))
    x1=int(input("Enter initial point x : "))
    y1=int(input("Enter initial point y : "))
    op.append(n)
    px.append(x1)
    py.append(y1)
    for i in range(1,n):
        # p=int(input("1. Line : "))
        x=int(input("Enter next point x : "))
        y=int(input("Enter next point y : "))
        op.append(1)
        px.append(x)
        py.append(y)
    op.append(1)
    px.append(x1)
    py.append(y1)
    tx=int(input("Enter translation value of x : "))
    ty=int(input("Enter translation value of y : "))
    win=GraphWin("DDA window",600,600)
    polygon_draw(op,px,py,tx,ty,win)
    win.getMouse()
    win.close()
```

Output:

```
Enter number of points : 4
Enter initial point x : 1
Enter initial point y : 2
Enter next point x : 100
Enter next point y : 200
Enter next point x : 200
Enter next point y : 100
Enter next point x : 1
Enter next point y : 2
Enter translation value of x : 15
Enter translation value of y : 15
```



10. Rotation :

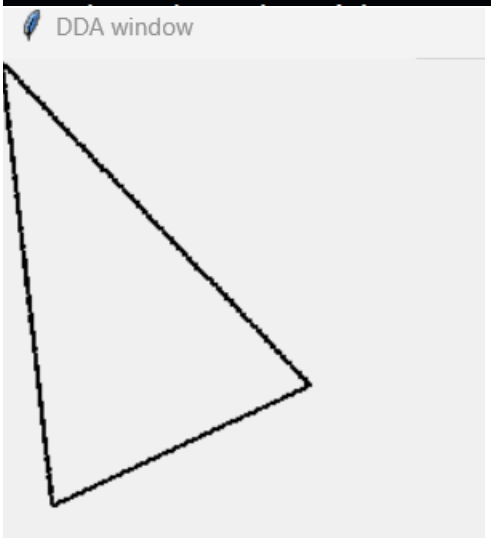
Code:

```
import time
from graphics import *
import numpy as np
import math
def rotation(x,y,theta,clk):
    pt=np.array([x,y])
    if clk==1:
        rot=np.array([[np.cos(theta),np.sin(theta)],[-
np.sin(theta),np.cos(theta)]])
    if clk==0:
        rot=np.array([[np.cos(theta),-
np.sin(theta)],[np.sin(theta),np.cos(theta)]])
    res=pt@rot
    return Point(res[0],res[1])

def dda(x1,y1,x2,y2,theta,clk,win):
    pt=rotation(x1,y1,theta,clk)
    pt.draw(win)
    dx=x2-x1
    dy=y2-y1
    steps=max(abs(dx),abs(dy))
    xtemp=x1
    ytemp=y1
```

```
    xinc=dx/steps
    yinc=dy/steps
    for i in range(steps):
        xtemp+=xinc
        ytemp+=yinc
        x1=int(xtemp)
        y1=int(ytemp)
        if i==steps-1:
            x1=int(x2)
            y1=int(y2)
        pt=rotation(x1,y1,theta,clk)
        pt.draw(win)
        time.sleep(0.001)


def polygon_draw(op,px,py,theta,clk,win):
    n=op[0]
    x1=px[0]
    y1=py[0]
    for i in range(1,n+1):
        x=px[i]
        y=py[i]
        if op[i]==1:
            dda(x1,y1,x,y,theta,clk,win)
            x1=x
            y1=y
if __name__=='__main__':
    px=[]
    py=[]
    op=[]
    n=int(input("Enter number of points : "))
    x1=int(input("Enter initial point x : "))
    y1=int(input("Enter initial point y : "))
    op.append(n)
    px.append(x1)
    py.append(y1)
    for i in range(1,n):
        p=int(input("1. Line : "))
        x=int(input("Enter next point x : "))
        y=int(input("Enter next point y : "))
        op.append(p)
        px.append(x)
        py.append(y)
    op.append(1)
    px.append(x1)
    py.append(y1)
    theta=int(input("Enter rotation angle : "))
    theta=math.radians(theta)
    clk=int(input("1. Clockwise,  0. Anti-Clockwise : "))
    win=GraphWin("DDA window",600,600)
    polygon_draw(op,px,py,theta,clk,win)
    win.getMouse()
    win.close()
```

Output:

```
Enter number of points : 4
Enter initial point x : 1
Enter initial point y : 2
1. Line : 1
Enter next point x : 100
Enter next point y : 200
1. Line : 1
Enter next point x : 200
Enter next point y : 100
1. Line : 1
Enter next point x : 1
Enter next point y : 2
Enter rotation angle : 20
1. Clockwise,  0. Anti-Clockwise : 1
```



11. Scaling:

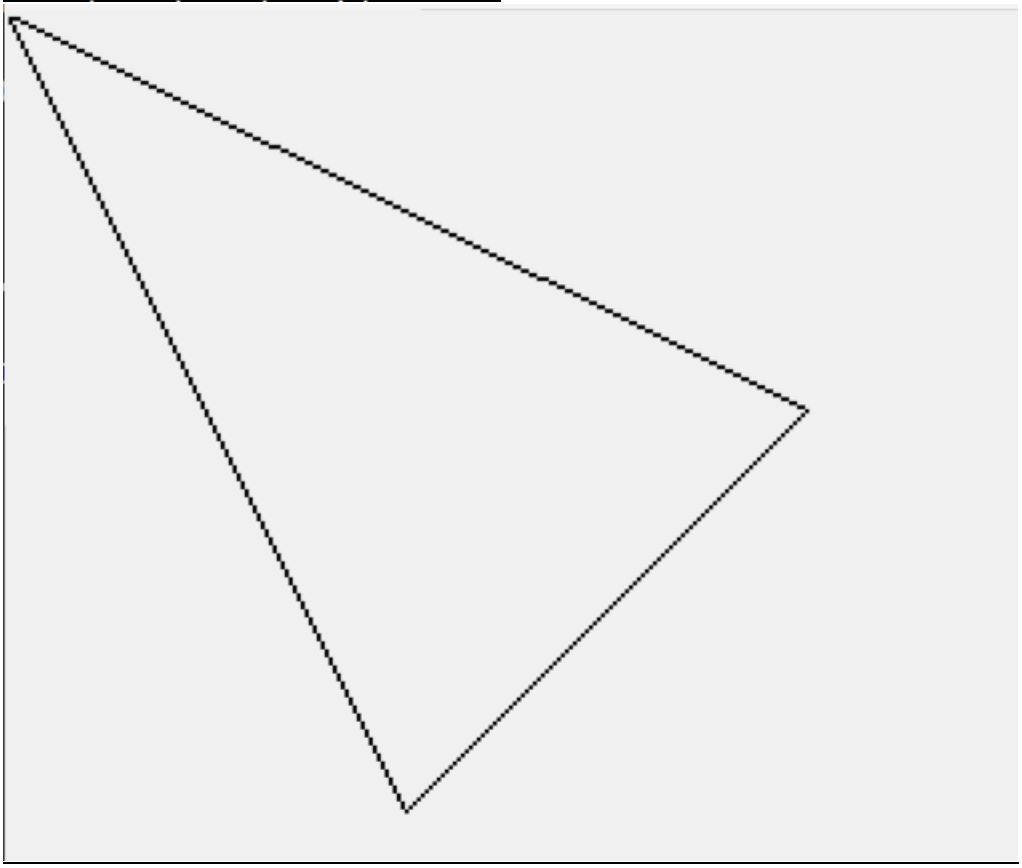
Code:

```
import time
from graphics import *
import numpy as np

def scaling(x,y,sx,sy):
    pt=np.array([x,y,1])
    scale=np.array([[sx,0,0],[0,sy,0],[0,0,1]])
    res=pt@scale
    return Point(res[0],res[1])

def dda(x1,y1,x2,y2,sx,sy,win):
    pt=scaling(x1,y1,sx,sy)
    pt.draw(win)
    dx=x2-x1
    dy=y2-y1
    steps=max(abs(dx),abs(dy))
    if steps==0:
        return
    xtemp=x1
    ytemp=y1
    xinc=dx/steps
    yinc=dy/steps
    for i in range(steps):
        xtemp+=xinc
```

```
        ytemp+=yinc
        x1=int(xtemp)
        y1=int(ytemp)
        if i==steps-1:
            x1=int(x2)
            y1=int(y2)
        pt=scaling(x1,y1,sx,sy)
        pt.draw(win)
        time.sleep(0.001)


def polygon_draw(op,px,py,sx,sy,win):
    n=op[0]
    x1=px[0]
    y1=py[0]
    for i in range(1,n+1):
        x=px[i]
        y=py[i]
        if op[i]==1:
            dda(x1,y1,x,y,sx,sy,win)
            x1=x
            y1=y

if __name__=='__main__':
    px=[]
    py=[]
    op=[]
    n=int(input("Enter number of points : "))
    x1=int(input("Enter initial point x : "))
    y1=int(input("Enter initial point y : "))
    op.append(n)
    px.append(x1)
    py.append(y1)
    for i in range(1,n):
        p=int(input("1. Line : "))
        x=int(input("Enter next point x : "))
        y=int(input("Enter next point y : "))
        op.append(p)
        px.append(x)
        py.append(y)
    op.append(1)
    px.append(x1)
    py.append(y1)
    sx=int(input("Enter scaling value of x : "))
    sy=int(input("Enter scaling value of y : "))
    win=GraphWin("DDA window",1000,1000)
    polygon_draw(op,px,py,sx,sy,win)
    win.getMouse()
    win.close()
```

Output:

```
Enter number of points : 4
Enter initial point x : 1
Enter initial point y : 2
1. Line : 1
Enter next point x : 100
Enter next point y : 200
1. Line : 1
Enter next point x : 200
Enter next point y : 100
1. Line : 1
Enter next point x : 1
Enter next point y : 2
Enter scaling value of x : 2
Enter scaling value of y : 2
```



12. Shearing :

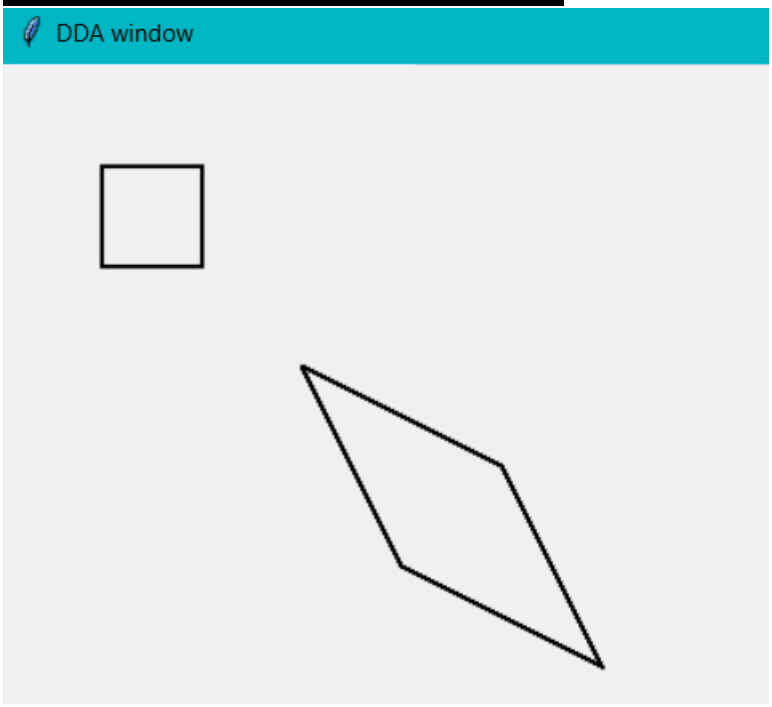
Code:

```
import time
from graphics import *
import numpy as np
ans = []
def shearing(x,y,shx,shy):
    pt=np.array([x,y,1])
    shear=np.array([[1,shy,0],[shx,1,0],[0,0,1]])
    res=pt@shear
    ans.append(res[0])
    print(res[0] , res[1])
    return (res[0],res[1])
def dda(x1,y1,x2,y2,win):
    pt=Point(x1,y1)
    pt.draw(win)
    dx=x2-x1
    dy=y2-y1
```

```
    steps=max(abs(dx),abs(dy))
    xtemp=x1
    ytemp=y1
    xinc=dx/steps
    yinc=dy/steps
    for i in range(steps):
        xtemp+=xinc
        ytemp+=yinc
        x1=int(xtemp)
        y1=int(ytemp)
        if i==steps-1:
            x1=int(x2)
            y1=int(y2)
        pt=Point(x1,y1)
        pt.draw(win)
        time.sleep(0.001)
def polygon_draw(n,px,py,win):
    x1=px[0]
    y1=py[0]
    for i in range(1,n+1):
        x=px[i]
        y=py[i]
        dda(x1,y1,x,y,win)
        x1=x
        y1=y
if __name__=='__main__':
    px=[]
    py=[]
    op=[]
    n=int(input("Enter number of points : "))
    x1=int(input("Enter initial point x : "))
    y1=int(input("Enter initial point y : "))
    px.append(x1)
    py.append(y1)
    for i in range(1,n):
        x=int(input("Enter next point x : "))
        y=int(input("Enter next point y : ")
        px.append(x)
        py.append(y)
    px.append(x1)
    py.append(y1)
    shx=int(input("Enter Shearing value of x : "))
    shy=int(input("Enter Shearing value of y : "))
    win=GraphWin("DDA window",1000,1000)
    polygon_draw(n,px,py,win)
    for i in range(n):
        x1,y1=shearing(px[i],py[i],shx,shy)
        x2,y2=shearing(px[i+1],py[i+1],shx,shy)
        dda(x1,y1,x2,y2,win)
    # ans = np.array(ans)
    print(ans)
    win.getMouse()
    win.close()
```

Output:

```
Enter number of points : 4
Enter initial point x : 50
Enter initial point y : 50
Enter next point x : 100
Enter next point y : 50
Enter next point x : 100
Enter next point y : 100
Enter next point x : 50
Enter next point y : 100
Enter Shearing value of x : 2
Enter Shearing value of y : 2
```



13. Reflection :

Code:

```
import time
from graphics import *
import numpy as np
ans = []
def x_reflect(x,y):
    pt=np.array([x,y,1])
    x_ref=np.array([[1,0,0],[0,-1,0],[0,0,1]])
    res=pt@x_ref
    ans.append(res[0])
    print(res[0] , res[1])
    return (res[0],res[1])
def dda(x1,y1,x2,y2,win):
    pt=Point(x1+250,y1+250)
    pt.draw(win)
    dx=x2-x1
    dy=y2-y1
    steps=max(abs(dx),abs(dy))
    xtemp=x1
    ytemp=y1
    xinc=dx/steps
    yinc=dy/steps
```

```
    for i in range(steps):
        xtemp+=xinc
        ytemp+=yinc
        x1=int(xtemp)
        y1=int(ytemp)
        if i==steps-1:
            x1=int(x2)
            y1=int(y2)
        pt=Point(x1+250,y1+250)
        pt.draw(win)
        time.sleep(0.001)
def polygon_draw(n,px,py,win):
    x1=px[0]
    y1=py[0]
    for i in range(1,n+1):
        x=px[i]
        y=py[i]
        dda(x1,y1,x,y,win)
        x1=x
        y1=y
if __name__=='__main__':
    px=[]
    py=[]
    op=[]
    n=int(input("Enter number of points : "))
    x1=int(input("Enter initial point x : "))
    y1=int(input("Enter initial point y : "))
    px.append(x1)
    py.append(y1)
    for i in range(1,n):
        x=int(input("Enter next point x : "))
        y=int(input("Enter next point y : "))
        px.append(x)
        py.append(y)
    px.append(x1)
    py.append(y1)
    win=GraphWin("DDA window",1000,1000)
    polygon_draw(n,px,py,win)
    for i in range(n):
        x1,y1=x_reflect(px[i],py[i])
        x2,y2=x_reflect(px[i+1],py[i+1])
        dda(x1,y1,x2,y2,win)
    # ans = np.array(ans)
    print(ans)
    win.getMouse()
    win.close()
```

Output:

```
Enter number of points : 4
Enter initial point x : 50
Enter initial point y : 50
Enter next point x : 100
Enter next point y : 50
Enter next point x : 100
Enter next point y : 100
Enter next point x : 50
Enter next point y : 100
```