**COMPUTER GRAPHICS ASSIGNMENT-4**

**TRANSFORMATIONS**

**Name-**ABHISHEK TIBREWAL

**Id-**2016UCP1103

**Batch-**A(1,2)

**2-D**

* **TRANSLATION:**

**from graphics import \***

**#function for translation**

**def translation(tx,ty):**

**for i in range(0,s):**

**pbt[i][0]+=tx**

**pbt[i][1]+=ty**

**print(pbt)**

**#taking input from user**

**print("Enter no. of sides in polygon:")**

**s=int(input("Enter sides:"))**

**print("Enter the points:")**

**pbt=[]**

**for i in range(0,s):**

**print("enter x coordinate of point-",i+1,":")**

**x=int(input())**

**print("enter y coordinate of point-",i+1,":")**

**y=int(input())**

**a=[x,y]**

**pbt.append(a)**

**window=GraphWin("2016UCP1103\_TRANSLATION",600,600) #for viewport(device coordinates)**

**window.setCoords(-300,-300,300,300) #for window(user coordinates)**

**window.setBackground("yellow")**

**#drwing user coordinate system**

**X=Line(Point(-300,0),Point(300,0)) #for drawing X-axis**

**X.setArrow('both')**

**X.setOutline('blue')**

**X.draw(window)**

**msg=Text(Point(290,10), "+X")**

**msg.draw(window)**

**msg=Text(Point(-290,10), "-X")**

**msg.draw(window)**

**Y=Line(Point(0,-300),Point(0,300)) #for drawing Y-axis**

**Y.setArrow('both')**

**Y.setOutline('blue')**

**Y.draw(window)**

**msg=Text(Point(10,290), "+Y")**

**msg.draw(window)**

**msg=Text(Point(10,-290), "-Y")**

**msg.draw(window)**

**msg=Text(Point(0,0), "(0,0)") #for origin**

**msg.draw(window)**

**#drawing original polygon**

**for i in range(1,s):**

**x1=pbt[i-1][0]**

**y1=pbt[i-1][1]**

**x2=pbt[i][0]**

**y2=pbt[i][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("red")**

**line.draw(window)**

**Text(Point(x1,y1),"("+str(x1)+","+str(y1)+")").draw(window)**

**x1=pbt[0][0]**

**y1=pbt[0][1]**

**x2=pbt[s-1][0]**

**y2=pbt[s-1][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("red")**

**line.draw(window)**

**Text(Point(x2,y2),"("+str(x2)+","+str(y2)+")").draw(window)**

**#calling translation**

**tx=int(input("Enter Translation in x:Tx="))**

**ty=int(input("Enter Translation in y:Ty="))**

**translation(tx,ty)**

**#drawing tranlated polygon**

**for i in range(1,s):**

**x1=pbt[i-1][0]**

**y1=pbt[i-1][1]**

**x2=pbt[i][0]**

**y2=pbt[i][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("blue")**

**line.draw(window)**

**Text(Point(x1,y1),"("+str(x1)+","+str(y1)+")").draw(window)**

**x1=pbt[0][0]**

**y1=pbt[0][1]**

**x2=pbt[s-1][0]**

**y2=pbt[s-1][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

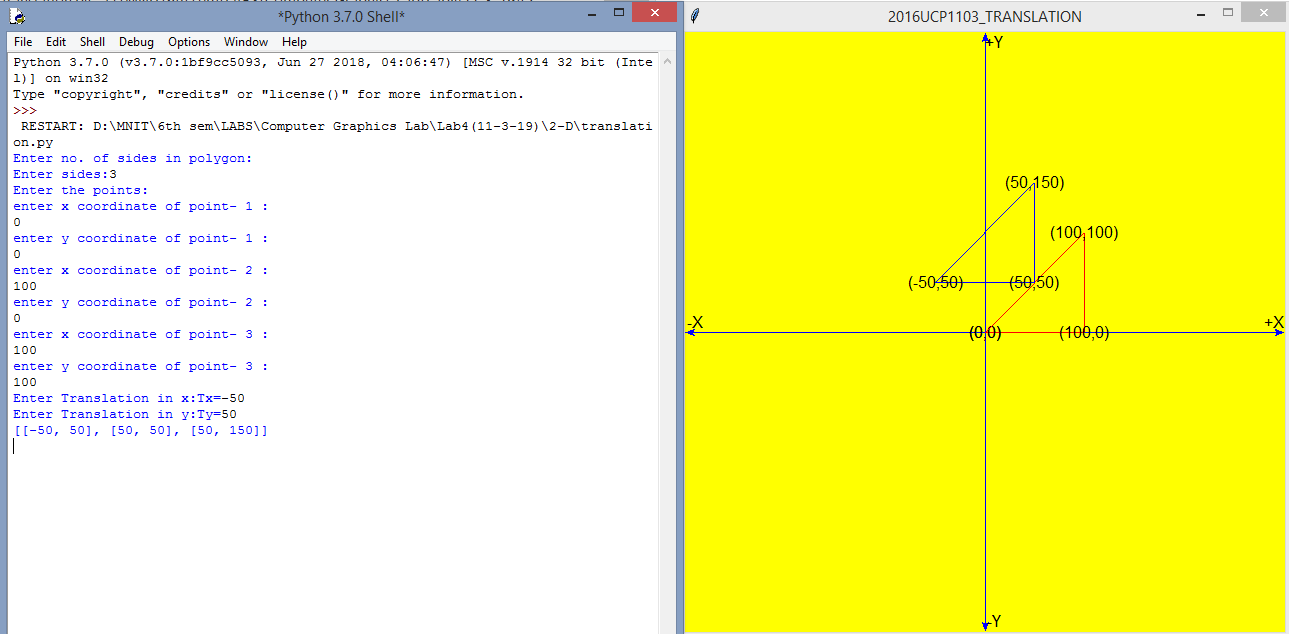
**line.setFill("blue")**

**line.draw(window)**

**Text(Point(x2,y2),"("+str(x2)+","+str(y2)+")").draw(window)**

**window.getMouse()**

**window.close()**

****

* **SCALING:**

**from graphics import \***

**#function for center of polygon**

**def getcenter():**

**global cx,cy**

**for i in range(0,s):**

**cx+=pbt[i][0]**

**cy+=pbt[i][1]**

**cx/=s**

**cy/=s**

**#function for translation**

**def translation(tx,ty):**

**for i in range(0,s):**

**pbt[i][0]=int(pbt[i][0]+tx)**

**pbt[i][1]=int(pbt[i][1]+ty)**

**print(pbt)**

**#function for scaling**

**def scaling(sx,sy):**

**for i in range(0,s):**

**pbt[i][0]=int(pbt[i][0]\*sx)**

**pbt[i][1]=int(pbt[i][1]\*sy)**

**print(pbt)**

**#taking input from user**

**print("Enter no. of sides in polygon:")**

**s=int(input("Enter sides:"))**

**print("Enter the points:")**

**pbt=[]**

**for i in range(0,s):**

**print("enter x coordinate of point-",i+1,":")**

**x=int(input())**

**print("enter y coordinate of point-",i+1,":")**

**y=int(input())**

**a=[x,y]**

**pbt.append(a)**

**window=GraphWin("2016UCP1103\_SCALING",600,600) #for viewport(device coordinates)**

**window.setCoords(-300,-300,300,300) #for window(user coordinates)**

**window.setBackground("yellow")**

**#drwing user coordinate system**

**X=Line(Point(-300,0),Point(300,0)) #for drawing X-axis**

**X.setArrow('both')**

**X.setOutline('blue')**

**X.draw(window)**

**msg=Text(Point(290,10), "+X")**

**msg.draw(window)**

**msg=Text(Point(-290,10), "-X")**

**msg.draw(window)**

**Y=Line(Point(0,-300),Point(0,300)) #for drawing Y-axis**

**Y.setArrow('both')**

**Y.setOutline('blue')**

**Y.draw(window)**

**msg=Text(Point(10,290), "+Y")**

**msg.draw(window)**

**msg=Text(Point(10,-290), "-Y")**

**msg.draw(window)**

**msg=Text(Point(0,0), "(0,0)") #for origin**

**msg.draw(window)**

**#drawing original polygon**

**for i in range(1,s):**

**x1=pbt[i-1][0]**

**y1=pbt[i-1][1]**

**x2=pbt[i][0]**

**y2=pbt[i][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("red")**

**line.draw(window)**

**Text(Point(x1,y1),"("+str(x1)+","+str(y1)+")").draw(window)**

**x1=pbt[0][0]**

**y1=pbt[0][1]**

**x2=pbt[s-1][0]**

**y2=pbt[s-1][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("red")**

**line.draw(window)**

**Text(Point(x2,y2),"("+str(x2)+","+str(y2)+")").draw(window)**

**#calling translation**

**sx=float(input("Enter Scaling in x:Sx="))**

**sy=float(input("Enter Scaling in y:Sy="))**

**cx=0**

**cy=0**

**getcenter()**

**translation(-cx,-cy)**

**scaling(sx,sy)**

**translation(cx,cy)**

**#drawing scaled polygon**

**for i in range(1,s):**

**x1=pbt[i-1][0]**

**y1=pbt[i-1][1]**

**x2=pbt[i][0]**

**y2=pbt[i][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("blue")**

**line.draw(window)**

**Text(Point(x1,y1),"("+str(x1)+","+str(y1)+")").draw(window)**

**x1=pbt[0][0]**

**y1=pbt[0][1]**

**x2=pbt[s-1][0]**

**y2=pbt[s-1][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

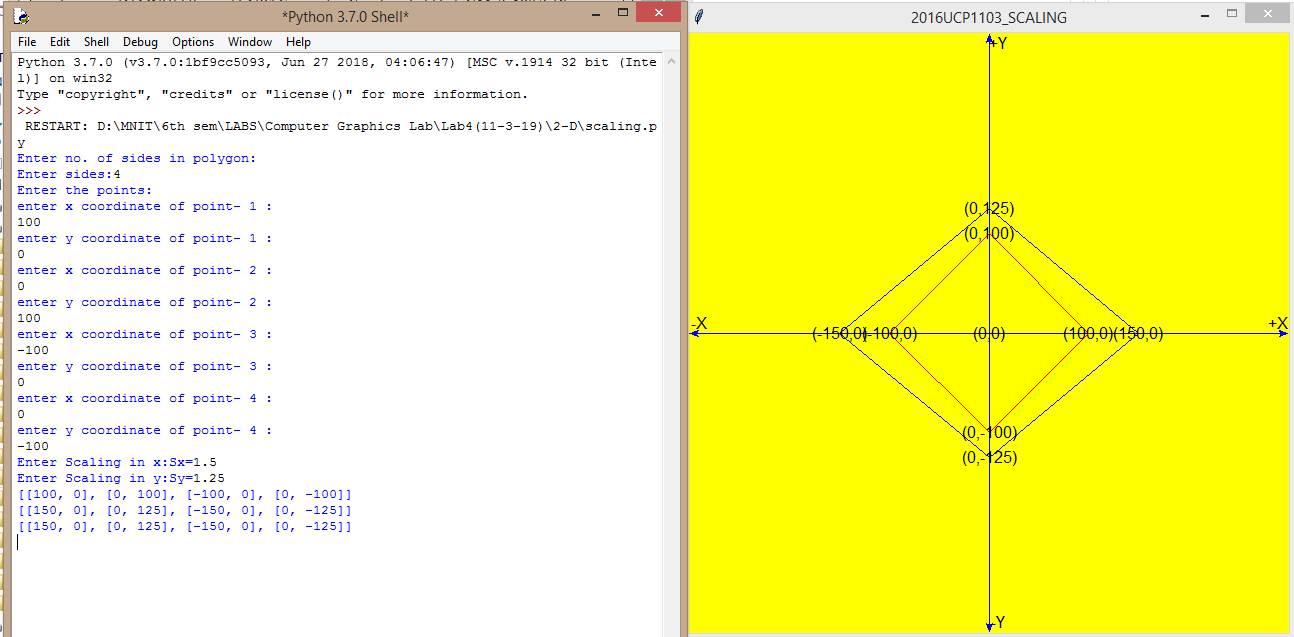
**line.setFill("blue")**

**line.draw(window)**

**Text(Point(x2,y2),"("+str(x2)+","+str(y2)+")").draw(window)**

**window.getMouse()**

**window.close()**

****

* **ROTATION:**

**from graphics import \***

**import math**

**#function for rotation**

**def rotation(phi):**

**for i in range(0,s):**

**angle=math.radians(phi)**

**x=pbt[i][0]**

**y=pbt[i][1]**

**pbt[i][0]=int((x\*math.cos(math.radians(phi)))-(y\*math.sin(math.radians(phi))))**

**pbt[i][1]=int((x\*math.sin(math.radians(phi)))+(y\*math.cos(math.radians(phi))))**

**print(pbt)**

**#function for translation**

**def translation(tx,ty):**

**for i in range(0,s):**

**pbt[i][0]=int(pbt[i][0]+tx)**

**pbt[i][1]=int(pbt[i][1]+ty)**

**print(pbt)**

**#taking input from user**

**print("Enter no. of sides in polygon:")**

**s=int(input("Enter sides:"))**

**print("Enter the points:")**

**pbt=[]**

**for i in range(0,s):**

**print("enter x coordinate of point-",i+1,":")**

**x=int(input())**

**print("enter y coordinate of point-",i+1,":")**

**y=int(input())**

**a=[x,y]**

**pbt.append(a)**

**window=GraphWin("2016UCP1103\_ROTATION ",600,600) #for viewport(device coordinates)**

**window.setCoords(-300,-300,300,300) #for window(user coordinates)**

**window.setBackground("yellow")**

**#drwing user coordinate system**

**X=Line(Point(-300,0),Point(300,0)) #for drawing X-axis**

**X.setArrow('both')**

**X.setOutline('blue')**

**X.draw(window)**

**msg=Text(Point(290,10), "+X")**

**msg.draw(window)**

**msg=Text(Point(-290,10), "-X")**

**msg.draw(window)**

**Y=Line(Point(0,-300),Point(0,300)) #for drawing Y-axis**

**Y.setArrow('both')**

**Y.setOutline('blue')**

**Y.draw(window)**

**msg=Text(Point(10,290), "+Y")**

**msg.draw(window)**

**msg=Text(Point(10,-290), "-Y")**

**msg.draw(window)**

**msg=Text(Point(0,0), "(0,0)") #for origin**

**msg.draw(window)**

**#drawing original polygon**

**for i in range(1,s):**

**x1=pbt[i-1][0]**

**y1=pbt[i-1][1]**

**x2=pbt[i][0]**

**y2=pbt[i][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("red")**

**line.draw(window)**

**Text(Point(x1,y1),"("+str(x1)+","+str(y1)+")").draw(window)**

**x1=pbt[0][0]**

**y1=pbt[0][1]**

**x2=pbt[s-1][0]**

**y2=pbt[s-1][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("red")**

**line.draw(window)**

**Text(Point(x2,y2),"("+str(x2)+","+str(y2)+")").draw(window)**

**#calling rotation**

**px=int(input("Enter X-coordinate of pivot point:"))**

**py=int(input("Enter Y-coordinate of pivot point:"))**

**phi=float(input("Enter angle of rotation(negative for clockwise direction):"))**

**translation(-1\*px,-1\*py)**

**rotation(phi)**

**translation(px,py)**

**#drawing rotated polygon**

**for i in range(1,s):**

**x1=pbt[i-1][0]**

**y1=pbt[i-1][1]**

**x2=pbt[i][0]**

**y2=pbt[i][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("blue")**

**line.draw(window)**

**Text(Point(x1,y1),"("+str(x1)+","+str(y1)+")").draw(window)**

**x1=pbt[0][0]**

**y1=pbt[0][1]**

**x2=pbt[s-1][0]**

**y2=pbt[s-1][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("blue")**

**line.draw(window)**

**Text(Point(x2,y2),"("+str(x2)+","+str(y2)+")").draw(window)**

**window.getMouse()**

**window.close()**

****

* **REFLECTION:**

**from graphics import \***

**import math**

**#function for rotation**

**def rotation(phi):**

**for i in range(0,s):**

**#angle=math.radians(phi)**

**x=pbt[i][0]**

**y=pbt[i][1]**

**pbt[i][0]=int((x\*math.cos(phi))-(y\*math.sin(phi)))**

**pbt[i][1]=int((x\*math.sin(phi))+(y\*math.cos(phi)))**

**print(pbt)**

**#function for translation**

**def translation(tx,ty):**

**for i in range(0,s):**

**pbt[i][0]=int(pbt[i][0]+tx)**

**pbt[i][1]=int(pbt[i][1]+ty)**

**print(pbt)**

**#function for reflection**

**def reflection(a):**

**if a=="x":**

**for i in range(0,s):**

**pbt[i][1]\*=(-1)**

**if a=="y":**

**for i in range(0,s):**

**pbt[i][0]\*=(-1)**

**if a=="o":**

**for i in range(0,s):**

**pbt[i][0]\*=(-1)**

**pbt[i][1]\*=(-1)**

**if a=="a":**

**print("Enter 2 coordinates of arbitrary line")**

**a0=float(input("Enter x0:"))**

**b0=float(input("Enter y0:"))**

**a1=float(input("Enter x1:"))**

**b1=float(input("Enter y1:"))**

**line=Line(Point(a0,b0),Point(a1,b1))**

**line.setFill("green")**

**line.draw(window)**

**if a1-a0==0:**

**phi=(90\*math.pi)/180**

**else:**

**phi=math.atan((b1-b0)/(a1-a0))**

**translation(-a0,-b0)**

**rotation(-phi)**

**for i in range(0,s):**

**pbt[i][1]\*=(-1)**

**rotation(phi)**

**translation(a0,b0)**

**#taking input from user**

**print("Enter no. of sides in polygon:")**

**s=int(input("Enter sides:"))**

**print("Enter the points:")**

**pbt=[]**

**for i in range(0,s):**

**print("enter x coordinate of point-",i+1,":")**

**x=int(input())**

**print("enter y coordinate of point-",i+1,":")**

**y=int(input())**

**a=[x,y]**

**pbt.append(a)**

**window=GraphWin("2016UCP1103\_REFLECTION",600,600) #for viewport(device coordinates)**

**window.setCoords(-300,-300,300,300) #for window(user coordinates)**

**window.setBackground("yellow")**

**#drwing user coordinate system**

**X=Line(Point(-300,0),Point(300,0)) #for drawing X-axis**

**X.setArrow('both')**

**X.setOutline('blue')**

**X.draw(window)**

**msg=Text(Point(290,10), "+X")**

**msg.draw(window)**

**msg=Text(Point(-290,10), "-X")**

**msg.draw(window)**

**Y=Line(Point(0,-300),Point(0,300)) #for drawing Y-axis**

**Y.setArrow('both')**

**Y.setOutline('blue')**

**Y.draw(window)**

**msg=Text(Point(10,290), "+Y")**

**msg.draw(window)**

**msg=Text(Point(10,-290), "-Y")**

**msg.draw(window)**

**msg=Text(Point(0,0), "(0,0)") #for origin**

**msg.draw(window)**

**#drawing original polygon**

**for i in range(1,s):**

**x1=pbt[i-1][0]**

**y1=pbt[i-1][1]**

**x2=pbt[i][0]**

**y2=pbt[i][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("red")**

**line.draw(window)**

**Text(Point(x1,y1),"("+str(x1)+","+str(y1)+")").draw(window)**

**x1=pbt[0][0]**

**y1=pbt[0][1]**

**x2=pbt[s-1][0]**

**y2=pbt[s-1][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("red")**

**line.draw(window)**

**Text(Point(x2,y2),"("+str(x2)+","+str(y2)+")").draw(window)**

**#calling reflection**

**print("Enter x for reflection about x axis")**

**print("Enter y for reflection about y axis")**

**print("Enter o for reflection about origin")**

**print("Enter a for reflection about arbitrary line")**

**opt=input("Enter option:")**

**reflection(opt)**

**#drawing reflected polygon**

**for i in range(1,s):**

**x1=pbt[i-1][0]**

**y1=pbt[i-1][1]**

**x2=pbt[i][0]**

**y2=pbt[i][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("blue")**

**line.draw(window)**

**Text(Point(x1,y1),"("+str(x1)+","+str(y1)+")").draw(window)**

**x1=pbt[0][0]**

**y1=pbt[0][1]**

**x2=pbt[s-1][0]**

**y2=pbt[s-1][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

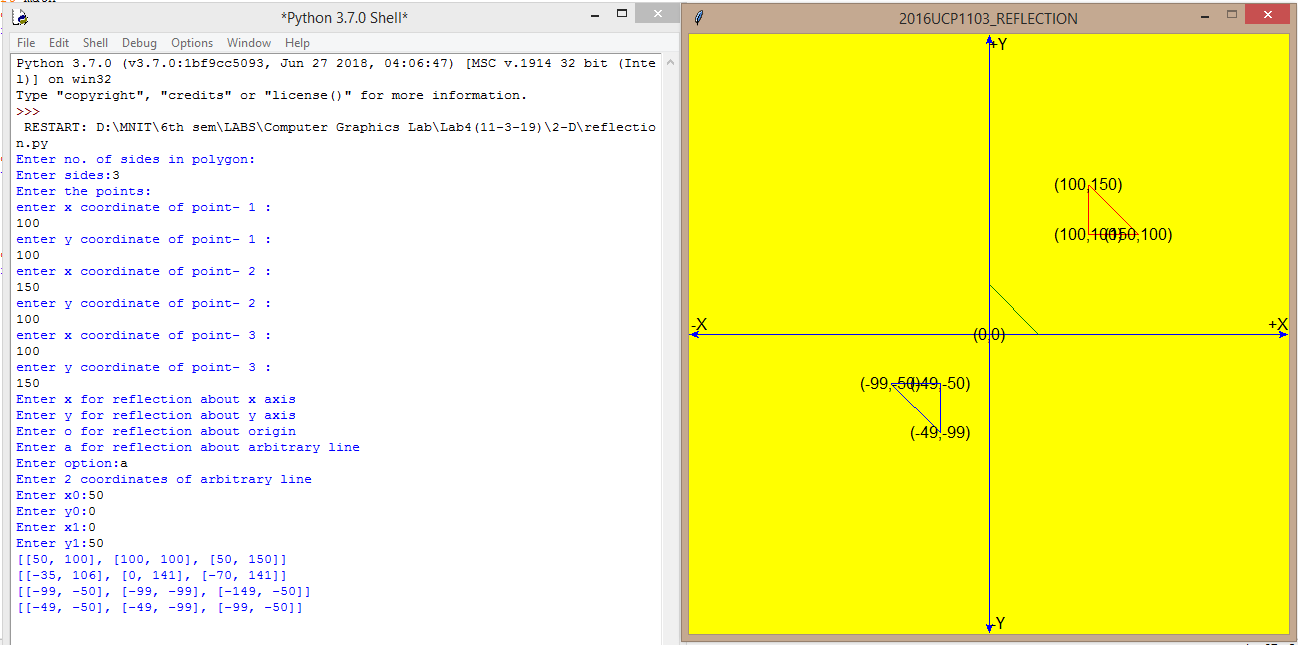
**line.setFill("blue")**

**line.draw(window)**

**Text(Point(x2,y2),"("+str(x2)+","+str(y2)+")").draw(window)**

**window.getMouse()**

**window.close()**

****

* **SHEARING:**

**from graphics import \***

**#function for center of polygon**

**def getcenter():**

**global cx,cy**

**for i in range(0,s):**

**cx+=pbt[i][0]**

**cy+=pbt[i][1]**

**cx/=s**

**cy/=s**

**#function for translation**

**def translation(tx,ty):**

**for i in range(0,s):**

**pbt[i][0]=int(pbt[i][0]+tx)**

**pbt[i][1]=int(pbt[i][1]+ty)**

**print(pbt)**

**#function for scaling**

**def scaling(sx,sy):**

**for i in range(0,s):**

**pbt[i][0]=int(pbt[i][0]\*sx)**

**pbt[i][1]=int(pbt[i][1]\*sy)**

**print(pbt)**

**#taking input from user**

**print("Enter no. of sides in polygon:")**

**s=int(input("Enter sides:"))**

**print("Enter the points:")**

**pbt=[]**

**for i in range(0,s):**

**print("enter x coordinate of point-",i+1,":")**

**x=int(input())**

**print("enter y coordinate of point-",i+1,":")**

**y=int(input())**

**a=[x,y]**

**pbt.append(a)**

**window=GraphWin("2016UCP1103\_SCALING",600,600) #for viewport(device coordinates)**

**window.setCoords(-300,-300,300,300) #for window(user coordinates)**

**window.setBackground("yellow")**

**#drwing user coordinate system**

**X=Line(Point(-300,0),Point(300,0)) #for drawing X-axis**

**X.setArrow('both')**

**X.setOutline('blue')**

**X.draw(window)**

**msg=Text(Point(290,10), "+X")**

**msg.draw(window)**

**msg=Text(Point(-290,10), "-X")**

**msg.draw(window)**

**Y=Line(Point(0,-300),Point(0,300)) #for drawing Y-axis**

**Y.setArrow('both')**

**Y.setOutline('blue')**

**Y.draw(window)**

**msg=Text(Point(10,290), "+Y")**

**msg.draw(window)**

**msg=Text(Point(10,-290), "-Y")**

**msg.draw(window)**

**msg=Text(Point(0,0), "(0,0)") #for origin**

**msg.draw(window)**

**#drawing original polygon**

**for i in range(1,s):**

**x1=pbt[i-1][0]**

**y1=pbt[i-1][1]**

**x2=pbt[i][0]**

**y2=pbt[i][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("red")**

**line.draw(window)**

**Text(Point(x1,y1),"("+str(x1)+","+str(y1)+")").draw(window)**

**x1=pbt[0][0]**

**y1=pbt[0][1]**

**x2=pbt[s-1][0]**

**y2=pbt[s-1][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("red")**

**line.draw(window)**

**Text(Point(x2,y2),"("+str(x2)+","+str(y2)+")").draw(window)**

**#calling translation**

**sx=float(input("Enter Scaling in x:Sx="))**

**sy=float(input("Enter Scaling in y:Sy="))**

**cx=0**

**cy=0**

**getcenter()**

**translation(-cx,-cy)**

**scaling(sx,sy)**

**translation(cx,cy)**

**#drawing scaled polygon**

**for i in range(1,s):**

**x1=pbt[i-1][0]**

**y1=pbt[i-1][1]**

**x2=pbt[i][0]**

**y2=pbt[i][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

**line.setFill("blue")**

**line.draw(window)**

**Text(Point(x1,y1),"("+str(x1)+","+str(y1)+")").draw(window)**

**x1=pbt[0][0]**

**y1=pbt[0][1]**

**x2=pbt[s-1][0]**

**y2=pbt[s-1][1]**

**line=Line(Point(x1,y1),Point(x2,y2))**

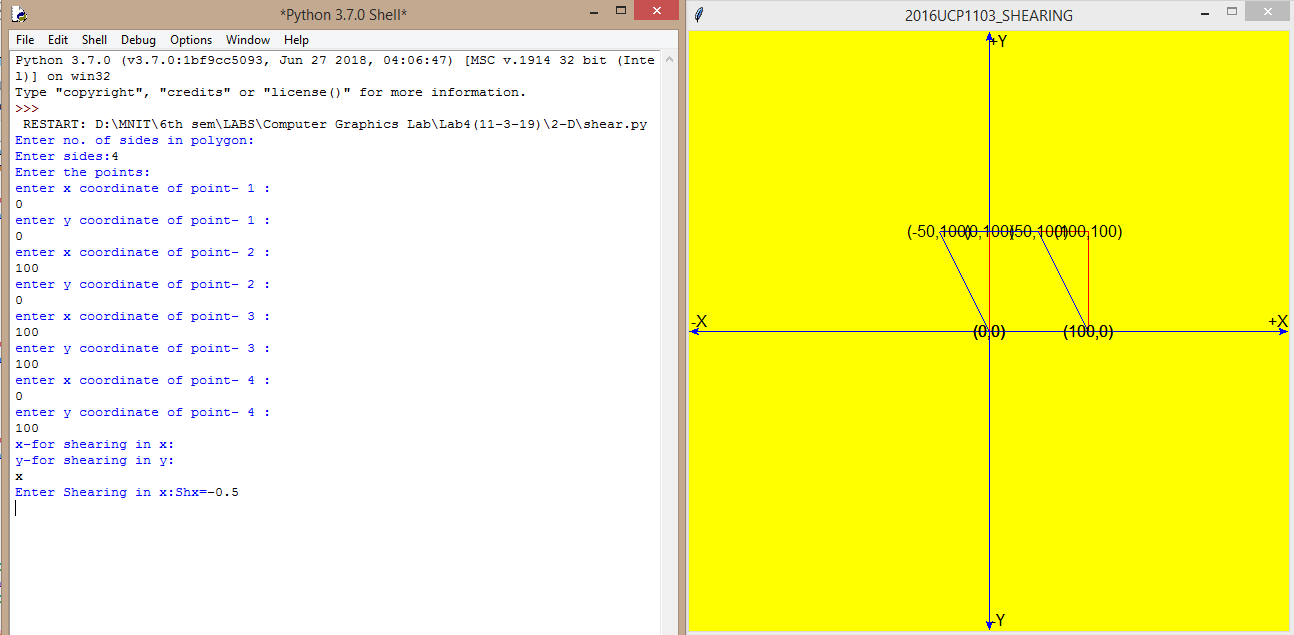
**line.setFill("blue")**

**line.draw(window)**

**Text(Point(x2,y2),"("+str(x2)+","+str(y2)+")").draw(window)**

**window.getMouse()**

**window.close()**

****

**3-D**

**CODE:**

**import numpy as np**

**import math**

**import time**

**import matplotlib.pyplot as plt**

**from mpl\_toolkits.mplot3d import Axes3D**

**def Trans(q,tx,ty,tz,centerx,centery,centerz):**

**shift=q.tolist()**

**for i in range(len(shift)):**

**shift[i][0]=shift[i][0]-centerx**

**shift[i][1]=shift[i][1]-centery**

**shift[i][2]=shift[i][2]-centerz**

**q=np.array(shift)**

**t=np.array([[1,0,0,0],[0,1,0,0],[0,0,1,0],[tx,ty,tz,1]])**

**tr=q.dot(t)**

**l=tr.tolist()**

**for i in range(len(l)):**

**l[i][0]=l[i][0]+centerx**

**l[i][1]=l[i][1]+centery**

**l[i][2]=l[i][2]+centerz**

**tr=np.array(l)**

**return tr**

**def Scale(q,sx,sy,sz,centerx,centery,centerz):**

**shift=q.tolist()**

**for i in range(len(shift)):**

**shift[i][0]=shift[i][0]-centerx**

**shift[i][1]=shift[i][1]-centery**

**shift[i][2]=shift[i][2]-centerz**

**q=np.array(shift)**

**s=np.array([[sx,0,0,0],[0,sy,0,0],[0,0,sz,0],[0,0,0,1]])**

**sc=q.dot(s)**

**l=sc.tolist()**

**for i in range(len(l)):**

**l[i][0]=l[i][0]+centerx**

**l[i][1]=l[i][1]+centery**

**l[i][2]=l[i][2]+centerz**

**sc=np.array(l)**

**return sc**

**def Rotate(q,theta,centerx,centery,centerz):**

**shift=q.tolist()**

**ch=int(input("enter 1-> x-axis 2-> y-axis 3-> z-axis "))**

**for i in range(len(shift)):**

**shift[i][0]=shift[i][0]-centerx**

**shift[i][1]=shift[i][1]-centery**

**shift[i][2]=shift[i][2]-centerz**

**q=np.array(shift)**

**if(ch == 3):**

**s=np.array([[math.cos(theta),math.sin(theta),0,0],[(-1\*math.sin(theta)),math.cos(theta),0,0],[0,0,1,0],[0,0,0,1]])**

**elif(ch == 1):**

**s=np.array([[1,0,0,0],[0,math.cos(theta),math.sin(theta),0],[0,(-1\*math.sin(theta)),math.cos(theta),0],[0,0,0,1]])**

**elif(ch == 2):**

**s=np.array([[math.cos(theta),0,(-1\*math.sin(theta)),0],[0,1,0,0],[(math.sin(theta)),0,math.cos(theta),0],[0,0,0,1]])**

**else:**

**print("wrong choice");**

**sc=q.dot(s)**

**l=sc.tolist()**

**for i in range(len(l)):**

**l[i][0]=l[i][0]+centerx**

**l[i][1]=l[i][1]+centery**

**l[i][2]=l[i][2]+centerz**

**sc=np.array(l)**

**return sc**

**def RotateAA(q,theta,centerx,centery,centerz,aax1,aay1,aaz1,A,B,C):**

**V=math.sqrt((B\*B+C\*C))**

**L=math.sqrt((A\*A+B\*B+C\*C))**

**shift=q.tolist()**

**c=C/V**

**s=B/V**

**c1=A/L**

**s1=V/L**

**for i in range(len(shift)):**

**shift[i][0]=shift[i][0]-centerx**

**shift[i][1]=shift[i][1]-centery**

**shift[i][2]=shift[i][2]-centerz**

**q=np.array(shift)**

**ss=np.array([[1,0,0,(-1\*aax1)],[0,1,0,(-1\*aay1)],[0,0,1,(-1\*aaz1)],[0,0,0,1]]) #Translation to pass from origin**

**sc=q.dot(ss)**

**ss=np.array([[1,0,0,0],[0,c,s,0],[0,(-1\*s),c,0],[0,0,0,1]]) #To bring in x-z plane**

**sc=sc.dot(ss)**

**ss=np.array([[s1,0,c1,0],[0,1,0,0],[(-1\*c1),0,s1,0],[0,0,0,1]]) #To make it z-axis**

**sc=sc.dot(ss)**

**ss=np.array([[math.cos(theta),math.sin(theta),0,0],[(-1\*math.sin(theta)),math.cos(theta),0,0],[0,0,1,0],[0,0,0,1]])#Actual Rotation**

**sc=sc.dot(ss)**

**ss=np.array([[s1,0,(-1\*c1),0],[0,1,0,0],[(c1),0,s1,0],[0,0,0,1]]) #Inverse of making it z-axis**

**sc=sc.dot(ss)**

**ss=np.array([[1,0,0,0],[0,c,(-1\*s),0],[0,(s),c,0],[0,0,0,1]]) #Inverse of bringing in x-z plane**

**sc=sc.dot(ss)**

**ss=np.array([[1,0,0,(aax1)],[0,1,0,(aay1)],[0,0,1,(aaz1)],[0,0,0,1]]) # Inverse of Translation**

**sc=sc.dot(ss)**

**l=sc.tolist()**

**for i in range(len(l)):**

**l[i][0]=l[i][0]+centerx**

**l[i][1]=l[i][1]+centery**

**l[i][2]=l[i][2]+centerz**

**sc=np.array(l)**

**n = len(polygon)**

**for i in range(0,n-1):**

**ax.plot([polygon[i][0], polygon[i+1][0]], [polygon[i][1], polygon[i+1][1]],[polygon[i][2], polygon[i+1][2]])**

**ax.plot([polygon[n-1][0], polygon[0][0]], [polygon[n-1][1], polygon[0][1]],[polygon[n-1][2], polygon[0][2]])**

**plt.show()**

**return sc**

**polygon = [[0,0,0,1],[1,0,0,1],[1,0,1,1],[0,0,1,1],[0,0,0,1],[0,1,0,1],[1,1,0,1],[1,1,1,1],[0,1,1,1],[0,1,0,1],[1,1,0,1],[1,0,0,1],[1,0,1,1],[1,1,1,1],[0,1,1,1],[0,0,1,1]]**

**n2=8**

**for kj in range(0,0):**

**x11=int(input("x"+str(kj+1)+" "));**

**y11=int(input("y"+str(kj+1)+" "));**

**z11=int(input("z"+str(kj+1)+" "));**

**polygon.append([x11,y11,z11,1])**

**fig = plt.figure("INITIAL")**

**ax = fig.add\_subplot(111, projection='3d')**

**n = len(polygon)**

**for i in range(0,n-1):**

**ax.plot([polygon[i][0], polygon[i+1][0]], [polygon[i][1], polygon[i+1][1]],[polygon[i][2], polygon[i+1][2]])**

**ax.plot([polygon[n-1][0], polygon[0][0]], [polygon[n-1][1], polygon[0][1]],[polygon[n-1][2], polygon[0][2]])**

**plt.show()**

**xm=[]**

**ym=[]**

**zm=[]**

**for i in range(len(polygon)):**

**xm.append(polygon[i][0])**

**ym.append(polygon[i][1])**

**zm.append(polygon[i][2])**

**centerx=sum(xm[:-1])/n2**

**centery=sum(ym[:-1])/n2**

**centerz=sum(zm[:-1])/n2**

**polyarr=np.array(polygon)**

**traask=int(input("enter 1 if want to do translation else 0"));**

**if(traask == 1):**

**tx=int(input("enter tx "))**

**ty=int(input("enter ty "))**

**tz=int(input("enter tz "))**

**polyarr=Trans(polyarr,tx,ty,tz,centerx,centery,centerz)**

**polygon=polyarr.tolist()**

**fig = plt.figure("AFTER TRANSLATION")**

**ax = fig.add\_subplot(111, projection='3d')**

**n = len(polygon)**

**for i in range(0,n-1):**

**ax.plot([polygon[i][0], polygon[i+1][0]], [polygon[i][1], polygon[i+1][1]],[polygon[i][2], polygon[i+1][2]])**

**ax.plot([polygon[n-1][0], polygon[0][0]], [polygon[n-1][1], polygon[0][1]],[polygon[n-1][2], polygon[0][2]])**

**plt.show()**

**scaask=int(input("enter 1 if want to do scaling else 0"));**

**if(scaask == 1):**

**sx=int(input("enter sx "))**

**sy=int(input("enter sy "))**

**sz=int(input("enter sz "))**

**polyarr=Scale(polyarr,sx,sy,sz,centerx,centery,centerz)**

**polygon=polyarr.tolist()**

**fig = plt.figure("AFTER SCALING")**

**ax = fig.add\_subplot(111, projection='3d')**

**n = len(polygon)**

**for i in range(0,n-1):**

**ax.plot([polygon[i][0], polygon[i+1][0]], [polygon[i][1], polygon[i+1][1]],[polygon[i][2], polygon[i+1][2]])**

**ax.plot([polygon[n-1][0], polygon[0][0]], [polygon[n-1][1], polygon[0][1]],[polygon[n-1][2], polygon[0][2]])**

**plt.show()**

**rotask=int(input("enter 1 if want to do rotation else 0"));**

**if(rotask == 1):**

**theta=int(input("enter angle of rotation "))**

**theta=(theta\*math.pi)/180;**

**polyarr=Rotate(polyarr,theta,centerx,centery,centerz)**

**polygon=polyarr.tolist()**

**fig = plt.figure("AFTER ROTATION")**

**ax = fig.add\_subplot(111, projection='3d')**

**n = len(polygon)**

**for i in range(0,n-1):**

**ax.plot([polygon[i][0], polygon[i+1][0]], [polygon[i][1], polygon[i+1][1]],[polygon[i][2], polygon[i+1][2]])**

**ax.plot([polygon[n-1][0], polygon[0][0]], [polygon[n-1][1], polygon[0][1]],[polygon[n-1][2], polygon[0][2]])**

**plt.show()**

**rotaask=int(input("enter 1 if want to do rotation about arbitrary axis else 0"));**

**if(rotaask == 1):**

**aax1=int(input("enter x1 of arbitray axis "))**

**aay1=int(input("enter y1 of arbitray axis "))**

**aaz1=int(input("enter z1 of arbitray axis "))**

**A=int(input("enter x direction ratio "))**

**B=int(input("enter y direction ratio "))**

**C=int(input("enter z direction ratio "))**

**theta=int(input("enter angle of rotation "))**

**theta=(theta\*math.pi)/180**

**polyarr=RotateAA(polyarr,theta,centerx,centery,centerz,aax1,aay1,aaz1,A,B,C)**

**polygon=polyarr.tolist()**

**fig = plt.figure("AFTER ROTATION ABOUT AXIS PASSING THROUGH ("+str(aax1)+","+str(aay1)+","+str(aaz1)+") and dirction Ratios ("+str(A)+","+str(B)+","+str(C)+")")**

**ax = fig.add\_subplot(111, projection='3d')**

**n = len(polygon)**

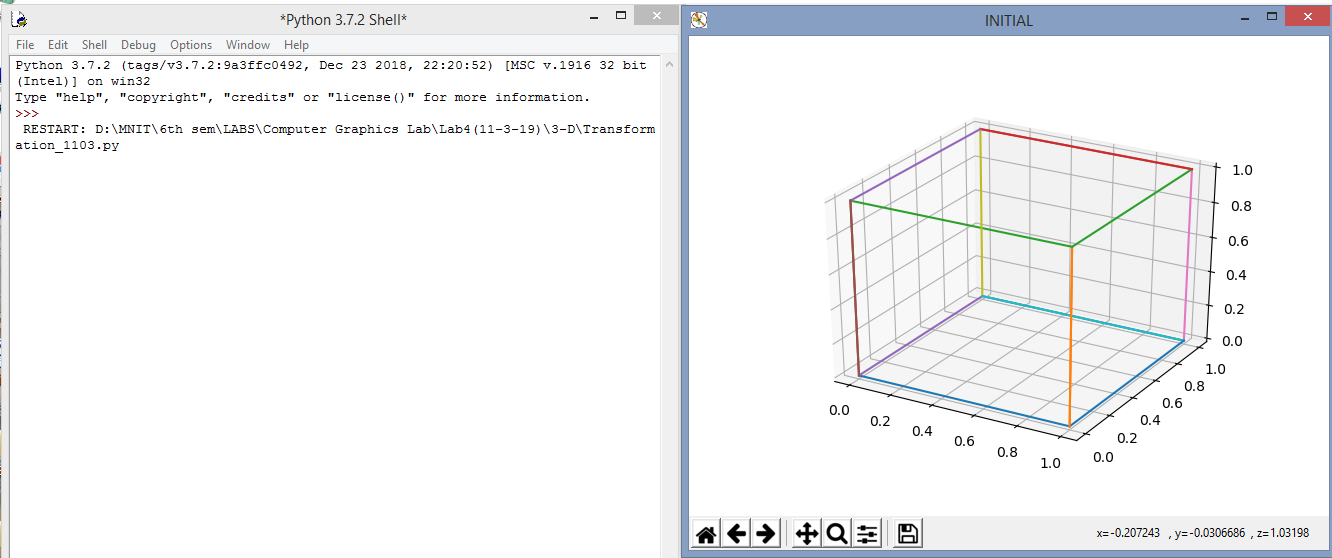
**for i in range(0,n-1):**

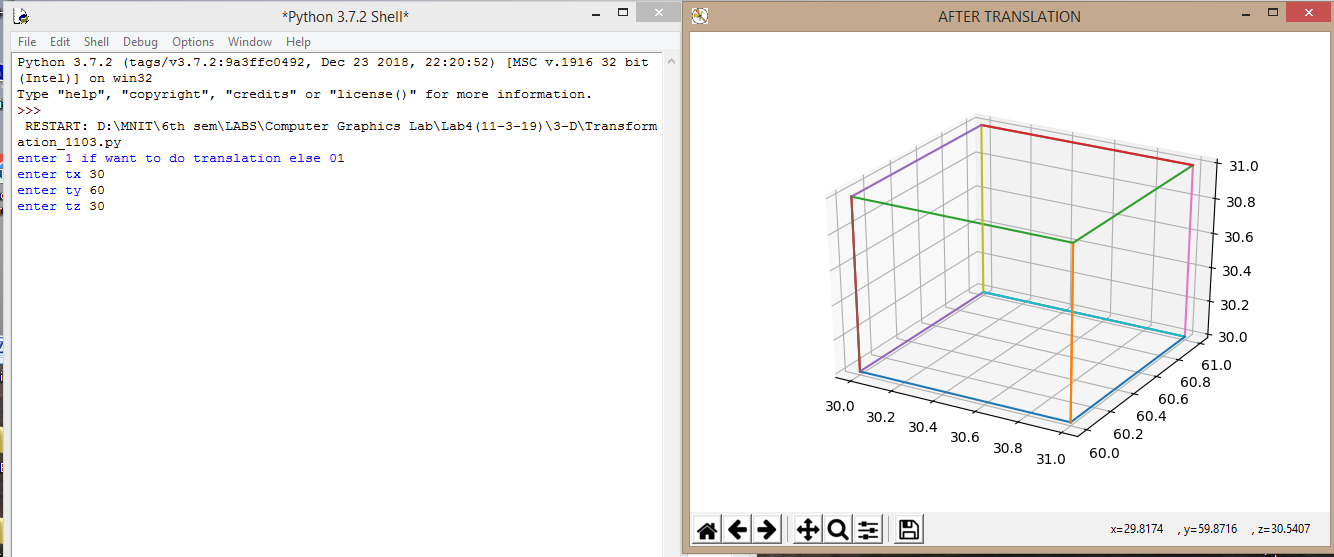
**ax.plot([polygon[i][0], polygon[i+1][0]], [polygon[i][1], polygon[i+1][1]],[polygon[i][2], polygon[i+1][2]])**

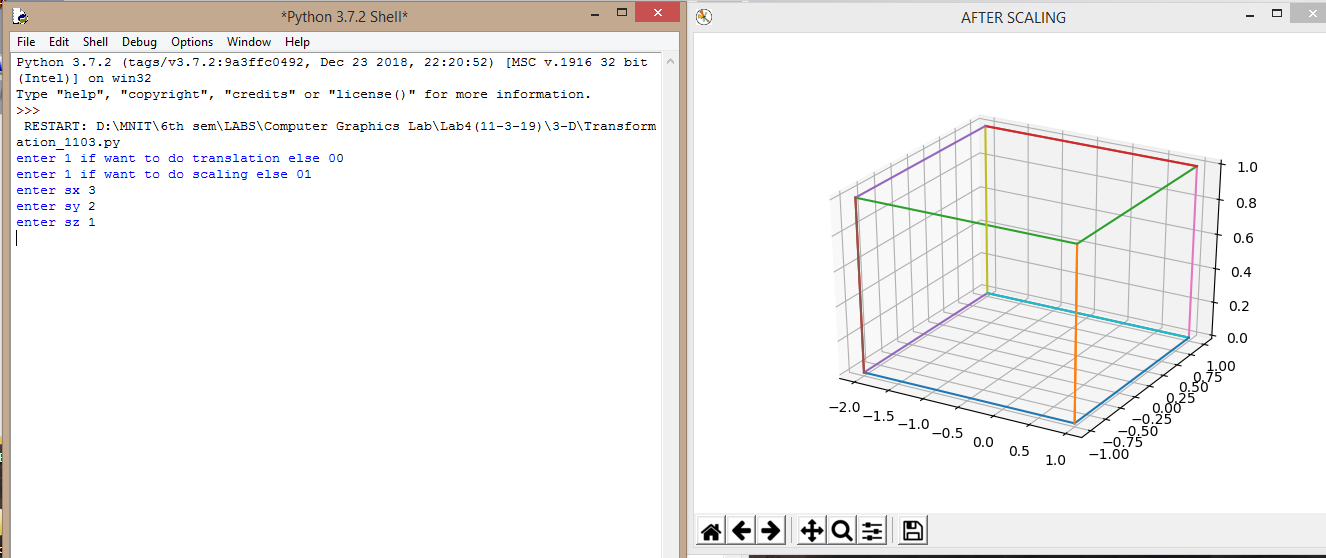
**ax.plot([polygon[n-1][0], polygon[0][0]], [polygon[n-1][1], polygon[0][1]],[polygon[n-1][2], polygon[0][2]])**

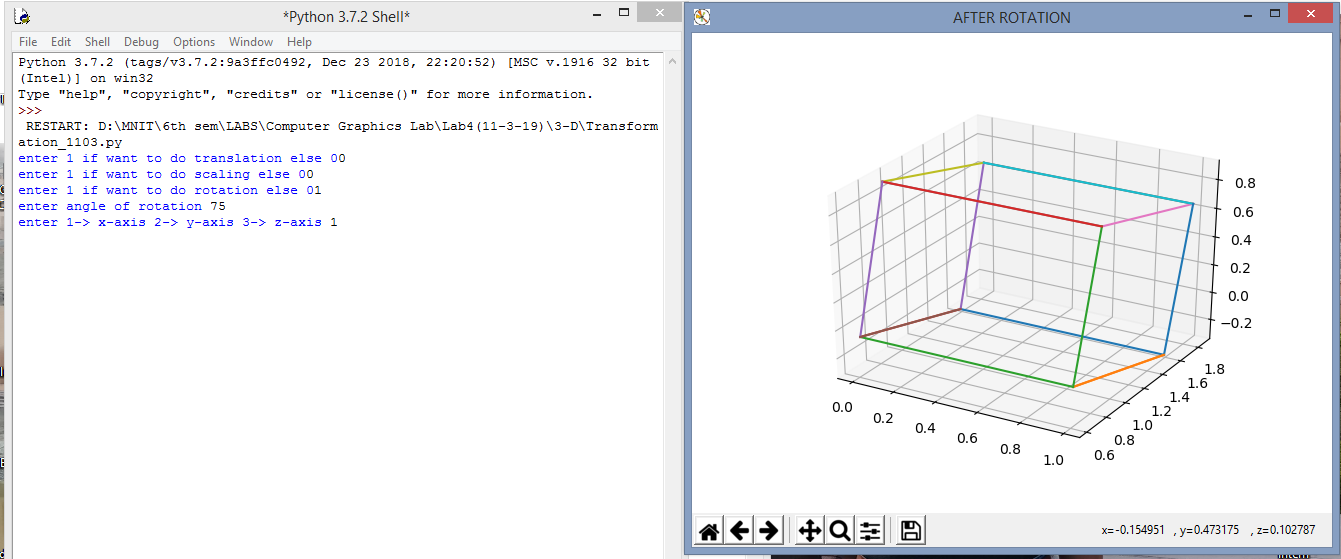
**plt.show()**

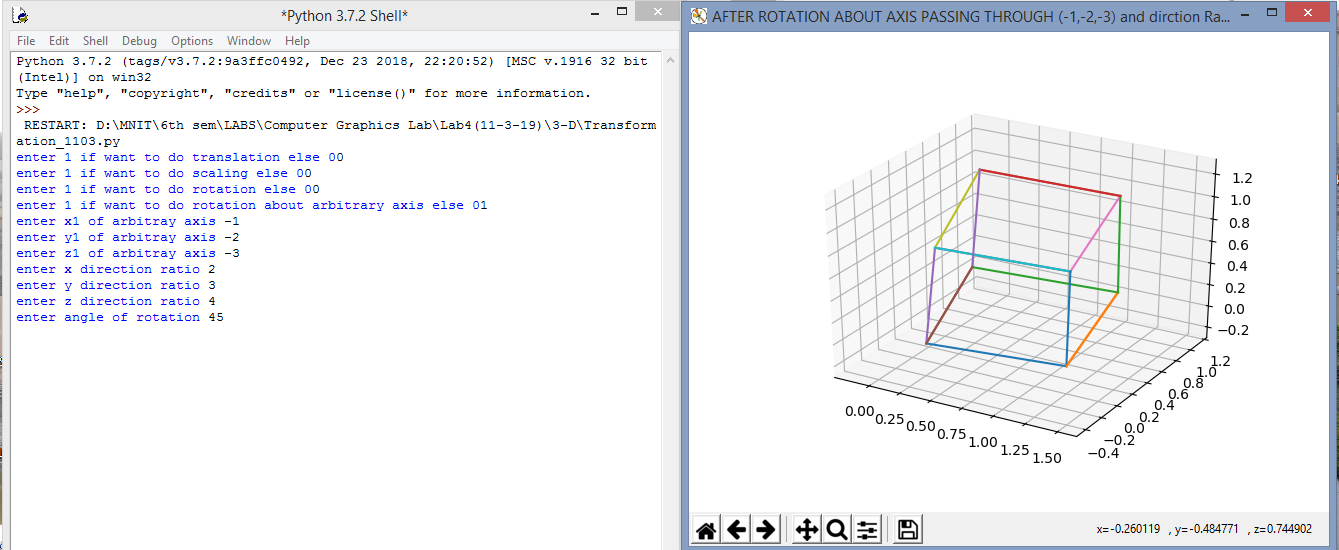
**OUTPUT:**

****

****

****

****

****