**COMPUTER GRAPHICS ASSIGNMENT-6**

**PROJECTIONS**

**Name-**ABHISHEK TIBREWAL

**Id-**2016UCP1103

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

* **How does the image change if you make the plane of projection as the principal planes and the center of projection is at infinity in the direction perpendicular to the plane of projection?**

**CODE:**

from graphics import \*

import math

f=open("input.txt","r")

c=int(f.readline())

vertex1 = []

f=open("input.txt","r")

c=int(f.readline())

vertex1=[]

vertex2=[]

for i in range(c):

number\_string=f.readline().split(' ')

number\_Array=[int(i) for i in number\_string]

point=[]

point.append(number\_Array[0])

point.append(number\_Array[1])

point.append(number\_Array[2])

vertex1.append(point)

point=[]

point.append(number\_Array[3])

point.append(number\_Array[4])

point.append(number\_Array[5])

vertex2.append(point)

win\_obj=GraphWin("PROJECTION\_1103",900,900)

win\_obj.setBackground("Light blue")

win\_obj.setCoords(-300,-300,600,600)

x\_axis=Line(Point(-300,0),Point(600,0))

y\_axis=Line(Point(0,-300),Point(0,600))

z\_axis=Line(Point(300,300),Point(-300,-300))

x\_axis.setOutline("Black")

y\_axis.setOutline("Black")

z\_axis.setOutline("Black")

x\_axis.setArrow('both')

y\_axis.setArrow('both')

z\_axis.setArrow('both')

x\_axis.draw(win\_obj)

y\_axis.draw(win\_obj)

z\_axis.draw(win\_obj)

info\_x=Text(Point(580,-10),"+x axis")

info\_x.draw(win\_obj)

info\_x=Text(Point(-280,-10),"-x axis")

info\_x.draw(win\_obj)

info\_y=Text(Point(-10,580),"+y axis")

info\_y.draw(win\_obj)

info\_y=Text(Point(-10,-280),"-y axis")

info\_y.draw(win\_obj)

info\_ny=Text(Point(-280,-280),"+z axis")

info\_ny.draw(win\_obj)

info\_ny=Text(Point(280,280),"-z axis")

info\_ny.draw(win\_obj)

origin=Text(Point(-10,-10),"origin")

origin.draw(win\_obj)

def drawLine(x0,y0,z0,x1,y1,z1,color):

ax, ay = x0-(z0\*0.3), y0-(z0\*0.3)

bx, by = x1-(z1\*0.3), y1-(z1\*0.3)

#print(ax,",",ay)

#print(bx,",",by)

line=Line(Point(ax,ay),Point(bx,by));

line.setFill(color)

line.setWidth(3)

line.draw(win\_obj)

def drawSolid(vertex1,vertex2,color):

for i in range(c):

x0 = vertex1[i][0]

y0 = vertex1[i][1]

z0 = vertex1[i][2]

x1 = vertex2[i][0]

y1 = vertex2[i][1]

z1 = vertex2[i][2]

drawLine(x0,y0,z0,x1,y1,z1,color)

drawSolid(vertex1,vertex2,"green")

tx=int(input("Enter Tx to translate:"))

ty=int(input("Enter Ty to translate:"))

tz=int(input("Enter Tz to translate:"))

vertex3=[]

vertex4=[]

for i in range(c):

point=[]

x0 = vertex1[i][0]+tx

point.append(x0)

y0 = vertex1[i][1]+ty

point.append(y0)

z0 = vertex1[i][2]+tz

point.append(z0)

vertex3.append(point)

point=[]

x1 = vertex2[i][0]+tx

point.append(x1)

y1 = vertex2[i][1]+ty

point.append(y1)

z1 = vertex2[i][2]+tz

point.append(z1)

vertex4.append(point)

drawSolid(vertex3,vertex4,"red")

vertex1=vertex3

vertex2=vertex4

vertex3=[]

vertex4=[]

print("Choose principle plane:: 1>XY 2>YZ 3>XZ")

f=int(input())

if f==1:

a1=1

b1=1

c1=0

elif f==2:

a1=0

b1=1

c1=1

elif f==3:

a1=1

b1=0

c1=1

for i in range(c):

point=[]

x0 = vertex1[i][0]\*a1

point.append(x0)

y0 = vertex1[i][1]\*b1

point.append(y0)

z0 = vertex1[i][2]\*c1

point.append(z0)

vertex3.append(point)

point=[]

x1 = vertex2[i][0]\*a1

point.append(x1)

y1 = vertex2[i][1]\*b1

point.append(y1)

z1 = vertex2[i][2]\*c1

point.append(z1)

vertex4.append(point)

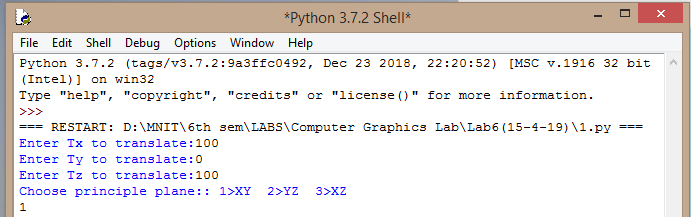
print(vertex3)

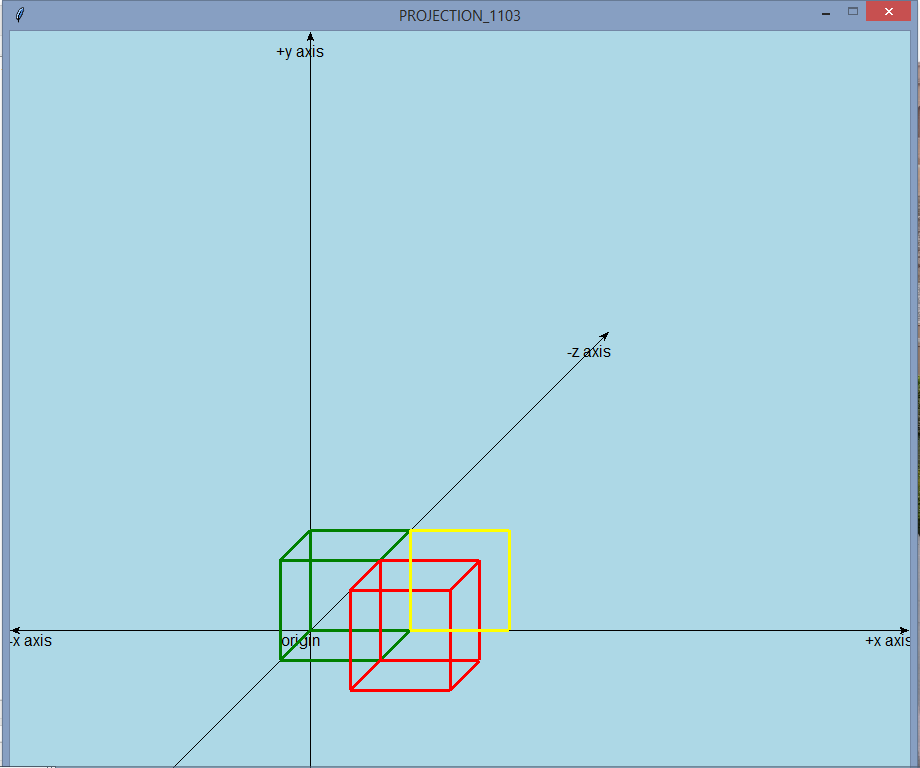
drawSolid(vertex3,vertex4,"yellow")

win\_obj.getMouse()

win\_obj.close()

**OUTPUT:**

****

****

* **How does the image change if you define the plane of projection with a reference point and normal to the plane; and the center of projection is at infinity in the direction perpendicular to the plane of projection?**

**CODE:**

from graphics import \*

import math

f=open("input.txt","r")

c=int(f.readline())

vertex1 = []

f=open("input.txt","r")

c=int(f.readline())

vertex1=[]

vertex2=[]

for i in range(c):

number\_string=f.readline().split(' ')

number\_Array=[int(i) for i in number\_string]

point=[]

point.append(number\_Array[0])

point.append(number\_Array[1])

point.append(number\_Array[2])

vertex1.append(point)

point=[]

point.append(number\_Array[3])

point.append(number\_Array[4])

point.append(number\_Array[5])

vertex2.append(point)

win\_obj=GraphWin("PROJECTION\_1103",900,900)

win\_obj.setBackground("Light blue")

win\_obj.setCoords(-300,-300,600,600)

x\_axis=Line(Point(-300,0),Point(600,0))

y\_axis=Line(Point(0,-300),Point(0,600))

z\_axis=Line(Point(300,300),Point(-300,-300))

x\_axis.setOutline("Black")

y\_axis.setOutline("Black")

z\_axis.setOutline("Black")

x\_axis.setArrow('both')

y\_axis.setArrow('both')

z\_axis.setArrow('both')

x\_axis.draw(win\_obj)

y\_axis.draw(win\_obj)

z\_axis.draw(win\_obj)

info\_x=Text(Point(580,-10),"+x axis")

info\_x.draw(win\_obj)

info\_x=Text(Point(-280,-10),"-x axis")

info\_x.draw(win\_obj)

info\_y=Text(Point(-10,580),"+y axis")

info\_y.draw(win\_obj)

info\_y=Text(Point(-10,-280),"-y axis")

info\_y.draw(win\_obj)

info\_ny=Text(Point(-280,-280),"+z axis")

info\_ny.draw(win\_obj)

info\_ny=Text(Point(280,280),"-z axis")

info\_ny.draw(win\_obj)

origin=Text(Point(-10,-10),"origin")

origin.draw(win\_obj)

def drawLine(x0,y0,z0,x1,y1,z1,color):

ax, ay = x0-(z0\*0.3), y0-(z0\*0.3)

bx, by = x1-(z1\*0.3), y1-(z1\*0.3)

#print(ax,",",ay)

#print(bx,",",by)

line=Line(Point(ax,ay),Point(bx,by));

line.setFill(color)

line.setWidth(3)

line.draw(win\_obj)

def drawSolid(vertex1,vertex2,color):

for i in range(c):

x0 = vertex1[i][0]

y0 = vertex1[i][1]

z0 = vertex1[i][2]

x1 = vertex2[i][0]

y1 = vertex2[i][1]

z1 = vertex2[i][2]

drawLine(x0,y0,z0,x1,y1,z1,color)

drawSolid(vertex1,vertex2,"green")

tx=int(input("Enter Tx to translate:"))

ty=int(input("Enter Ty to translate:"))

tz=int(input("Enter Tz to translate:"))

vertex3=[]

vertex4=[]

for i in range(c):

point=[]

x0 = vertex1[i][0]+tx

point.append(x0)

y0 = vertex1[i][1]+ty

point.append(y0)

z0 = vertex1[i][2]+tz

point.append(z0)

vertex3.append(point)

point=[]

x1 = vertex2[i][0]+tx

point.append(x1)

y1 = vertex2[i][1]+ty

point.append(y1)

z1 = vertex2[i][2]+tz

point.append(z1)

vertex4.append(point)

drawSolid(vertex3,vertex4,"red")

vertex1=vertex3

vertex2=vertex4

vertex3=[]

vertex4=[]

print("Enter Normal vector of the plane ai+bj+ck::")

a1,b1,c1=input().split()

a1=int(a1)

b1=int(b1)

c1=int(c1)

print("Enter reference point of plane x0 y0 z0 ::")

p,q,r=input().split()

p=int(p)

q=int(q)

r=int(r)

d1=a1\*a1+b1\*b1+c1\*c1

d0=p\*a1+q\*b1+r\*c1

for i in range(c):

point=[]

x0 = (vertex1[i][0]\*(d1-a1\*a1)-vertex1[i][1]\*a1\*b1-vertex1[i][2]\*a1\*c1+a1\*d0)/d1

x0=int(x0)

point.append(x0)

y0 = (vertex1[i][1]\*(d1-b1\*b1)-vertex1[i][0]\*b1\*a1-vertex1[i][2]\*b1\*c1+b1\*d0)/d1

y0=int(y0)

point.append(y0)

z0 = (vertex1[i][2]\*(d1-c1\*c1)-vertex1[i][0]\*c1\*a1-vertex1[i][1]\*c1\*b1+c1\*d0)/d1

z0=int(z0)

point.append(z0)

vertex3.append(point)

point=[]

x1 = (vertex2[i][0]\*(d1-a1\*a1)-vertex2[i][1]\*a1\*b1-vertex2[i][2]\*a1\*c1+a1\*d0)/d1

x1=int(x1)

point.append(x1)

y1 = (vertex2[i][1]\*(d1-b1\*b1)-vertex2[i][0]\*b1\*a1-vertex2[i][2]\*b1\*c1+b1\*d0)/d1

y1=int(y1)

point.append(y1)

z1 = (vertex2[i][2]\*(d1-c1\*c1)-vertex2[i][0]\*c1\*a1-vertex2[i][1]\*c1\*b1+c1\*d0)/d1

z1=int(z1)

point.append(z1)

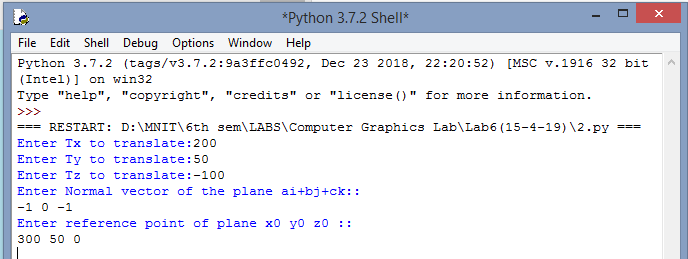
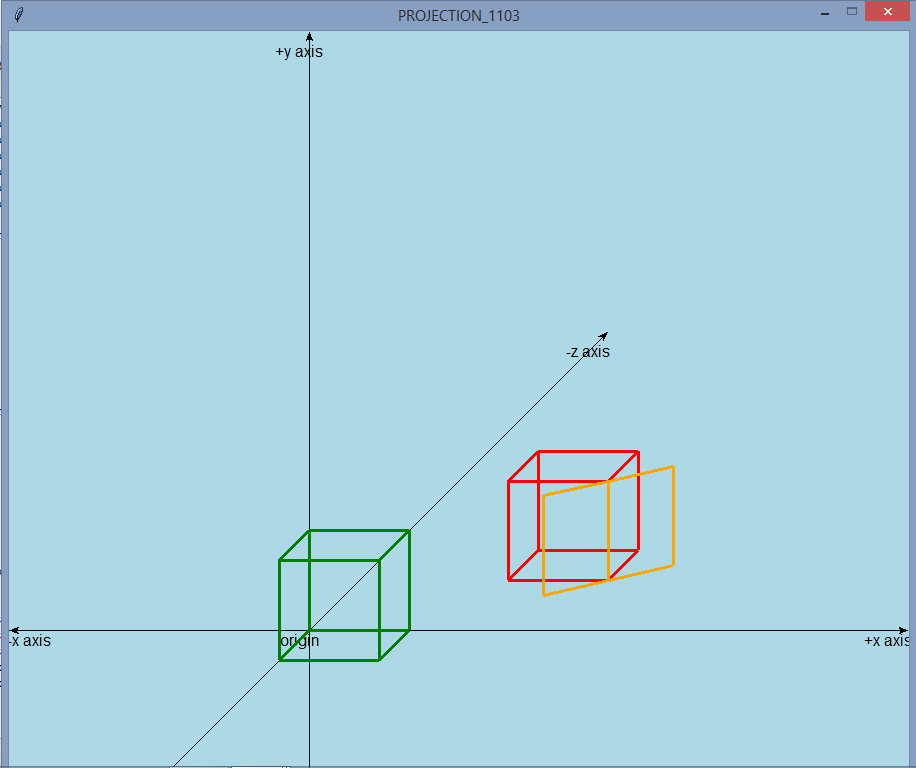
vertex4.append(point)

drawSolid(vertex3,vertex4,"orange")

win\_obj.getMouse()

win\_obj.close()

**OUTPUT:**

****

* **How does the image change if you define the plane of projection with a reference point and normal to the plane; and the center of projection is at a point**

**CODE:**

from graphics import \*

import math

f=open("input.txt","r")

c=int(f.readline())

vertex1 = []

f=open("input.txt","r")

c=int(f.readline())

vertex1=[]

vertex2=[]

for i in range(c):

number\_string=f.readline().split(' ')

number\_Array=[int(i) for i in number\_string]

point=[]

point.append(number\_Array[0])

point.append(number\_Array[1])

point.append(number\_Array[2])

vertex1.append(point)

point=[]

point.append(number\_Array[3])

point.append(number\_Array[4])

point.append(number\_Array[5])

vertex2.append(point)

win\_obj=GraphWin("PROJECTION\_1103",900,900)

win\_obj.setBackground("Light blue")

win\_obj.setCoords(-300,-300,600,600)

x\_axis=Line(Point(-300,0),Point(600,0))

y\_axis=Line(Point(0,-300),Point(0,600))

z\_axis=Line(Point(300,300),Point(-300,-300))

x\_axis.setOutline("Black")

y\_axis.setOutline("Black")

z\_axis.setOutline("Black")

x\_axis.setArrow('both')

y\_axis.setArrow('both')

z\_axis.setArrow('both')

x\_axis.draw(win\_obj)

y\_axis.draw(win\_obj)

z\_axis.draw(win\_obj)

info\_x=Text(Point(580,-10),"+x axis")

info\_x.draw(win\_obj)

info\_x=Text(Point(-280,-10),"-x axis")

info\_x.draw(win\_obj)

info\_y=Text(Point(-10,580),"+y axis")

info\_y.draw(win\_obj)

info\_y=Text(Point(-10,-280),"-y axis")

info\_y.draw(win\_obj)

info\_ny=Text(Point(-280,-280),"+z axis")

info\_ny.draw(win\_obj)

info\_ny=Text(Point(280,280),"-z axis")

info\_ny.draw(win\_obj)

origin=Text(Point(-10,-10),"origin")

origin.draw(win\_obj)

def drawLine(x0,y0,z0,x1,y1,z1,color):

ax, ay = x0-(z0\*0.3), y0-(z0\*0.3)

bx, by = x1-(z1\*0.3), y1-(z1\*0.3)

#print(ax,",",ay)

#print(bx,",",by)

line=Line(Point(ax,ay),Point(bx,by));

line.setFill(color)

line.setWidth(3)

line.draw(win\_obj)

def drawSolid(vertex1,vertex2,color):

for i in range(c):

x0 = vertex1[i][0]

y0 = vertex1[i][1]

z0 = vertex1[i][2]

x1 = vertex2[i][0]

y1 = vertex2[i][1]

z1 = vertex2[i][2]

drawLine(x0,y0,z0,x1,y1,z1,color)

drawSolid(vertex1,vertex2,"green")

tx=int(input("Enter Tx to translate:"))

ty=int(input("Enter Ty to translate:"))

tz=int(input("Enter Tz to translate:"))

vertex3=[]

vertex4=[]

for i in range(c):

point=[]

x0 = vertex1[i][0]+tx

point.append(x0)

y0 = vertex1[i][1]+ty

point.append(y0)

z0 = vertex1[i][2]+tz

point.append(z0)

vertex3.append(point)

point=[]

x1 = vertex2[i][0]+tx

point.append(x1)

y1 = vertex2[i][1]+ty

point.append(y1)

z1 = vertex2[i][2]+tz

point.append(z1)

vertex4.append(point)

drawSolid(vertex3,vertex4,"red")

vertex1=vertex3

vertex2=vertex4

vertex3=[]

vertex4=[]

print("Enter Normal vector of the plane n1i+n2j+n3k ::")

n1,n2,n3=input().split()

n1=int(n1)

n2=int(n2)

n3=int(n3)

print("Enter reference point of plane x0 y0 z0 ::")

p,q,r=input().split()

p=int(p)

q=int(q)

r=int(r)

print("Enter center of projection a0 b0 c0 ::")

a0,b0,c0=input().split()

a0=int(a0)

b0=int(b0)

c0=int(c0)

d1=a0\*n1+b0\*n2+c0\*n3

d0=p\*n1+q\*n2+r\*n3

d=d0-d1

for i in range(c):

x=vertex1[i][0]

y=vertex1[i][1]

z=vertex1[i][2]

point=[]

x0 =(x\*(a0\*n1+d)+y\*a0\*n2+z\*a0\*n3-a0\*d0)/(x\*n1+y\*n2+z\*n3-d1)

x0=int(x0)

point.append(x0)

y0 =(x\*b0\*n1+y\*(b0\*n2+d)+z\*b0\*n3-b0\*d0)/(x\*n1+y\*n2+z\*n3-d1)

y0=int(y0)

point.append(y0)

z0 =(x\*c0\*n1+y\*c0\*n2+z\*(c0\*n3+d)-c0\*d0)/(x\*n1+y\*n2+z\*n3-d1)

z0=int(z0)

point.append(z0)

vertex3.append(point)

x=vertex2[i][0]

y=vertex2[i][1]

z=vertex2[i][2]

point=[]

x1 = (x\*(a0\*n1+d)+y\*a0\*n2+z\*a0\*n3-a0\*d0)/(x\*n1+y\*n2+z\*n3-d1)

x1=int(x1)

point.append(x1)

y1 =(x\*b0\*n1+y\*(b0\*n2+d)+z\*b0\*n3-b0\*d0)/(x\*n1+y\*n2+z\*n3-d1)

y1=int(y1)

point.append(y1)

z1 = (x\*c0\*n1+y\*c0\*n2+z\*(c0\*n3+d)-c0\*d0)/(x\*n1+y\*n2+z\*n3-d1)

z1=int(z1)

point.append(z1)

vertex4.append(point)

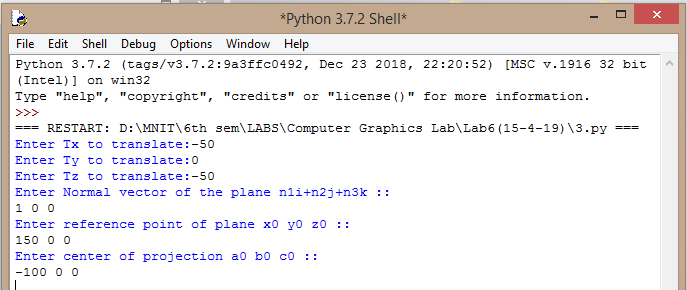
drawSolid(vertex3,vertex4,"orange")

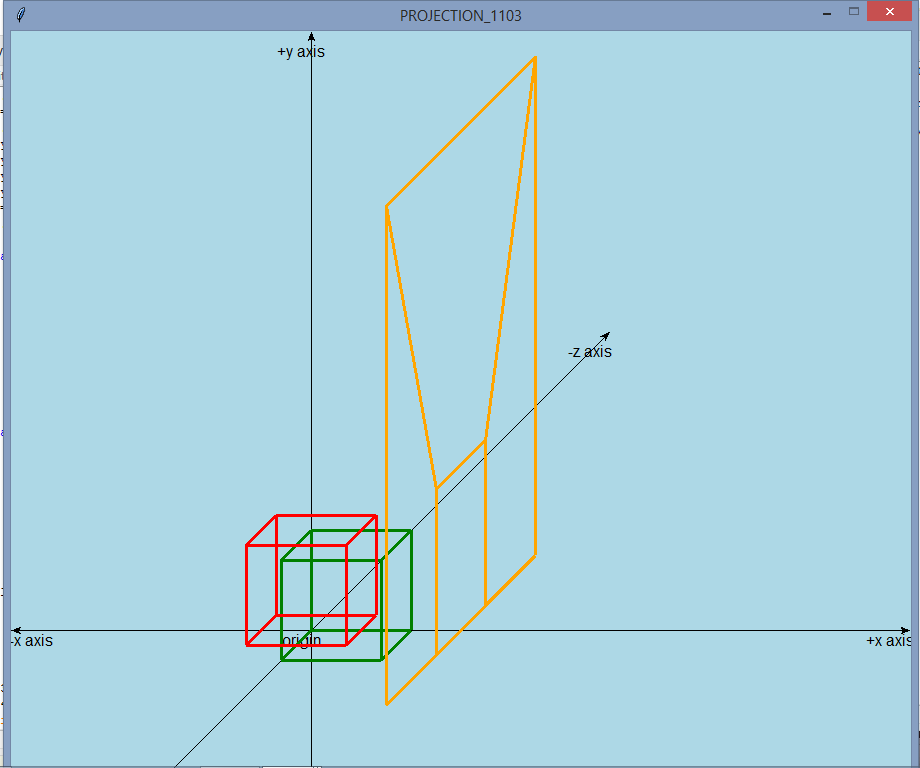
win\_obj.getMouse()

win\_obj.close()

**1)on any of the axis**

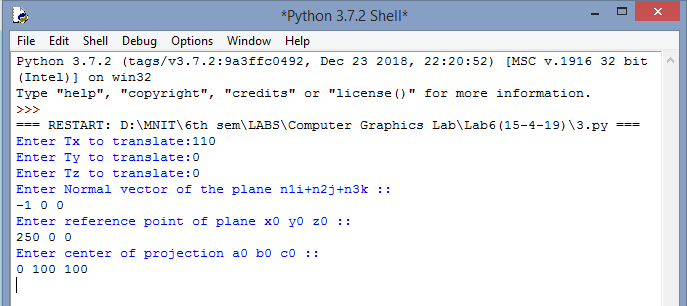
**OUTPUT:**

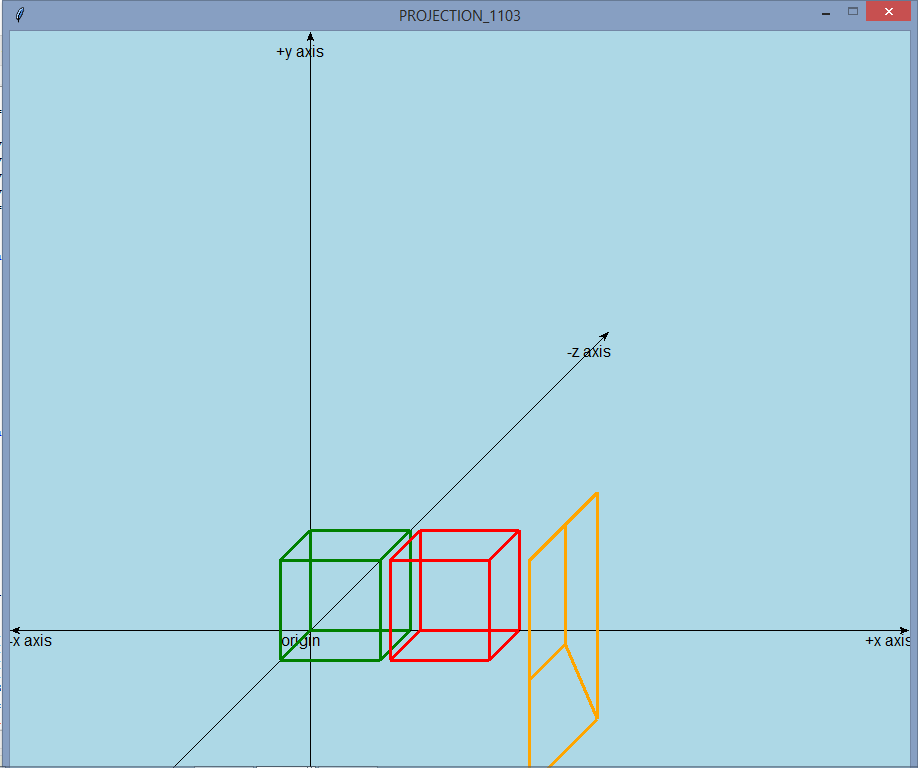
****

****

**2)or it is in a plane**

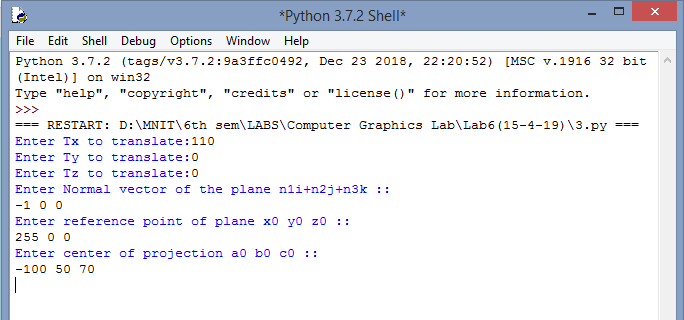
**OUTPUT:**

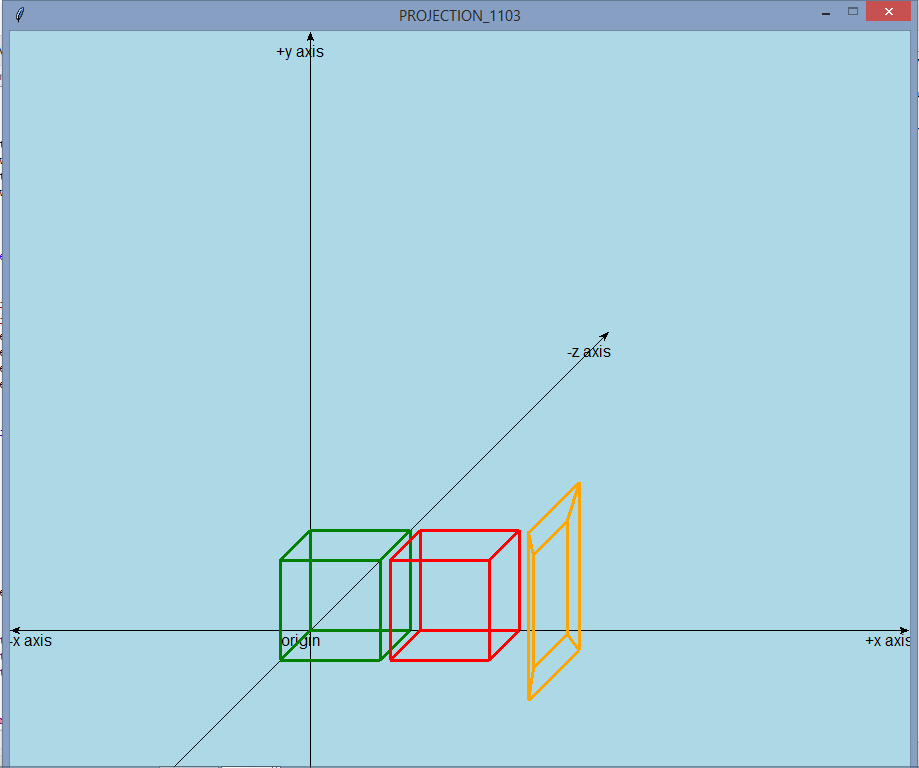
****

****

**3) it is in space.**

**OUTPUT:**

****

****