

```
#      **** Rota_Mat 0.1 ****      #
# Universidad de San Carlos de Guatemala #
# Facultad de CC QQ y FF              #
# Escuela de Quimica                  #
# Matematica IV                       #
# Nombre: Rony Jose Letona Lee        #
# Carne: 200960024                    #
#
# --- Importando librerias ---      #
import fpformat
# --- Se definen funciones/metodos --- #
# Revision del dato ingresado (flotante)
def error_test2(x):
    try:
        if int(x) > 0:
            return False
        else:
            return True
    except:
        return True
# Revision del dato ingresado (entero)
def error_test3(x):
    try:
        float(x)
        return False
    except:
        return True
# Revision general
def error_test(y, z=1):
    if z == 1:
        while error_test2(y):
            print "\n\n\t\tERROR!!!\n\t\tSe han ingresado valores o caracteres no validos."
            y = raw_input("\n\nPor favor, reingrese el valor deseado: ")
            return int(y)
    elif z == 2:
        while error_test3(y):
            print "\n\n\t\tERROR!!!\n\t\tSe han ingresado valores o caracteres no validos."
            y = raw_input("\n\nPor favor, reingrese el valor deseado: ")
            return float(y)
# Generado de la matriz
def generador(n):
    mat = [[0,0,0],[0,0,0],[0,0,0]]
    den = n[0]**2 + n[1]**2 + n[2]**2
    for i in range(3):
        for j in range(3):
            if (i == j):
                mat[i][j] = 1
            mat[i][j] -= (2 * n[i] * n[j]) / den
    return mat
# Despliega matriz
def impri(m):
    print "\n"
    print "\t|" + ("\t" * (len(m[0]) + 1)) + "||"
    for i in range(len(m)):
        print "\t| |\t",
        for j in range(len(m[0])):
            print fpformat.fix(m[i][j],2),"\t",
        print "||"
```

```

        print "\t|" + ("\t" * (len(m[0]) + 1)) + "|"
    print "\n"

# Multiplicador de matriz por vector
def mult(mat, vec):
    multi = [0,0,0]
    for i in range(3):
        for j in range(3):
            multi[i] += mat[i][j] * vec[j]
    return multi

# Multiplicador de vectores (producto punto)
def vmul(vec1, vec2):
    temp = 0
    for i in range(3):
        temp += vec1[i] * vec2[i]
    if temp == 0:
        return True
    else:
        return False

# Sumador de vectores
def addi(vec1, vec2):
    vec = [0,0,0]
    for i in range(3):
        vec[i] = vec1[i] + vec2[i]
    return vec

# Vector por constante
def invert(v):
    for i in range(3):
        v[i] *= -1
    return v

# Comparador de vectores
def comp(vec1, vec2):
    try:
        temp = vec1[0] / vec2[0]
        ctrl = 0
        for i in range(3):
            if ((vec2[i] * temp) != vec1[i]):
                ctrl += 1
        if ctrl == 0:
            return True
        else:
            return False
    except:
        if (vec1[0] == vec2[0]) and (vec1[1] == vec2[1]) and (vec1[2] == vec2[2]):
            return True
        else:
            return False

# Obtencion de vectores
def extr(name,t=0):
    print
    vec = [0,0,0]
    pal = [" primer ", " segundo ", " tercer "]
    tip = [" vector ", " punto "]
    for i in range(3):
        vec[i] = error_test(raw_input("Ingrese el" + pal[i] + "componente del" + tip[t] + name + ": "),
2)
    return vec

# Creador de listas de vectores
def multiv():
    print
    lista = []

```

```
num = error_test(raw_input("Cuantos vectores/puntos desea reflejar sobre el plano? "),1)
for i in range(num):
    lista.append(extr(str(i+1)))
return lista

# Multiconversor
def converter(lis, mat, poin, n):
    resul = []
    ra = mult(mat,poin)
    for i in range(len(lis)):
        if vmul(lis[i], n):
            resul.append(addi(lis[i],ra))
        elif comp(lis[i], n):
            resul.append(addi(invert(lis[i]), ra))
        else:
            resul.append(addi(mult(mat, lis[i]),ra))
    return resul

# --- Programa Principal --- #

print "\n"*5

norm = extr("normal")
print norm
poin = extr("del plano",1)

R = generador(norm)

print "Matriz de reflexion:"
impri(R)

vectores = multiv()
resul = converter(vectores, R, poin, norm)

print "\n\nSus vectores reflejados son:\n"

for i in resul:
    for j in i:
        print "|| " + str(fpformat.fix(j,2)) + " ||"
    print "\n\n"
```