In [37]:
```
#Importing the library
import my_library
Q1=my_library.Matrix(my_library.read_matrix('mat1.csv')) #Augmented matrix
Q2=my_library.Matrix(my_library.read_matrix('mat2.csv')) #Augmented matrix
Q3=my_library.Matrix(my_library.read_matrix('mat3.csv')) #Augmented matrix
Q3_matrix=my_library.Matrix(my_library.read_matrix('matrix.csv')) #not augmented
Q4=my_library.Matrix(my_library.read_matrix('mat4.csv')) #Augmented matrix
```

In [38]:
```
#Solving Question 1
Q1.show()
#Below is our augmented Matrix
```

```
1,1,1,1,13,

2,3,0,-1,-1,

-3,4,1,2,10,

1,2,-1,1,1,
```

In [39]:
```
my_library.GaussJordan(Q1)
Q1.show() #
# Solution with very small values are taken to be 0
# solutions are the last coloumn in augmented matrix
```

```
1.0,0.0,0.0,0.0,2.000000000000001,

0.0,1.0,0.0,0.0,-1.3322676295501878e-15,

0.0,0.0,1.0,0.0,5.999999999999999,

0.0,0.0,0.0,1.0,5.0,
```

In [40]:
```
print("As we can see the approximate values are x =2 , y =0 , z=6 , w =5")
```

```
As we can see the approximate values are x =2 , y =0 , z=6 , w =5
```

In [41]:
```
#Solving Question 2
Q2.show()
```

```
0,2,-3,-1,

1,0,1,0,

1,-1,0,3,
```

In [42]:
```
my_library.GaussJordan(Q2)
Q2.show()
# note -0.0 is 0
```

```
1.0,0.0,0.0,1.0,

0.0,1.0,0.0,-2.0,

-0.0,-0.0,1.0,-1.0,
```

In [43]:
```
print("As we can see the approximate values are x =1 , y =-2 , z=-1")
```

```
As we can see the approximate values are x =1 , y =-2 , z=-1
```

In [44]:
```
#Solving Question 3
Q3.show()
```

```
0,2,1,1,0,0,

4,0,1,0,1,0,

-1,2,0,0,0,1,
```

In [45]:
```
Q3_inv=my_library.inverse(Q3)

for i in range(len(Q3_inv.lol)):
    for j in range(len(Q3_inv.lol[0])):
        Q3_inv.lol[i][j]=float(format(Q3_inv.lol[i][j],'.2f')) # approximated to
```

In [46]:
```
# INVERSE TESTING
Product=my_library.Mat_product(Q3_inv,Q3_matrix,show=False)

for i in range(len(Product.lol)):
    for j in range(len(Product.lol[0])):
        Product.lol[i][j]=float(format(Product.lol[i][j],'.2f'))
Product.show()
```

```
0.99,0.0,0.0,

0.01,1.0,0.0,

0.01,0.0,1.0,
```

In [47]:
```
print("As expected we get the identity matrix")
```

```
As expected we get the identity matrix
```

In [48]:
```
#Solving Question 4
Q4.show()
```

```
1,4,2,3,

0,1,4,4,

-1,0,1,0,

2,0,4,1,
```

In [49]:
```
my_library.Determinant(Q4)
```

Out[49]:
```
65.0
```

In [ ]:

In [ ]: