In [1]:

```
#Question1
#Use Gauss-Jordon elimination to find the solution of the following system of linear eq
uations

# x + y + z + w = 13
# 2x + 3y − w = −1
# −3x + 4y + z + 2w = 10
# x + 2y − z + w = 1


#Augmented form has been entered to matrix2.txt

#              _               _
# C= [A | b]= | 1 1 1 1    | 13  |
#             | 2 3 0 -1   | -1  |
#             | -3 4 1 2   | 10  |
#             |_ 1 2 -1 1 |   1 _|
#



#importing gauss Jordan from library
from My_Lib import Gauss_jordan



# inserting matrix C
list_C=[]
with open("matrix1.txt") as matC:
    for k in matC:
        list_C.append(list(map(float, k.split())))


#######################################
#Criterion for when exististance of solution
a,b=Gauss_jordan(list_C)
if (a,b) != (None,None):
    print("Solution of the given Linear equation is:") #output rounded upto 3 places of
decimal
    print("x = %.2f" %b[0])
    print("y = %.2f" %b[1])
    print("z = %.2f" %b[2])
    print("w = %.2f" %b[3])
else:
    print("No unique solutions exist")
```

```
Solution of the given Linear equation is:
x = 2.00
y = -0.00
z = 6.00
w = 5.00
```

In [1]:

```
#Question2
#Use Gauss-Jordon elimination to find the solution of the following system of linear eq
uations

#2y - 3z = -1
#x + z = 0
#x - y = 3


#Readable augmented matrix form:
#Entering one matrix C=[A|b]

# C=          _                 _
# [A | b] = | 0 2 -3    | -1   |
#           | 1 0  1    |  0   |
#           |_1 -1 0    |  3 _|
#
#Readable augmented matrix form:
#Entering one matrix C=[A|b]



 #importing gauss Jordan function from library
from My_Lib import Gauss_jordan


list_C=[]
with open("matrix2.txt") as matC:
    for k in matC:
        list_C.append(list(map(float, k.split())))

#######################################
#Criterion for exististance of solution
a,b=Gauss_jordan(list_C)
if (a,b) != (None,None):
    print("Solution of the given Linear equation is:")#output rounded upto 3 places of
 decimal
    print("x = %.2f"%b[0])
    print("y = %.2f"%b[1])
    print("z = %.2f"%b[2])

else:
    print("No unique solution exist for the given equation")
```

```
Solution of the given Linear equation is:
x = 1.00
y = -2.00
z = -1.00
```

In [1]:

```
# Question 3
#Find the inverse of the following invertible matrix using Gauss-Jordon elimination. Ke
ep only up to 2 places in decimal. Verify that your solution is indeed the inverse of t
he given matrix.
#
# [A|b] =
#   _                  _
# |   0 2 1   | 1 0 0   |
# |   4 0 1   | 0 1 0   |
# |_ -1 2 0   | 0 0 1 _|




#calling the matrix in readable form
#importing functions from library
from My_Lib import Gauss_jordan
from My_Lib import matrix_mul

#calling the matrix in readable form
list_C=[]
with open("matrix3.txt") as matC:
    for k in matC:
        list_C.append(list(map(float, k.split())))




#using the gauss jordan function
a,b=Gauss_jordan(list_C)
print("The Inverse of the given matrix is A[-1]:")

# Print the inverse matrix in readable form
for i in range(len(b)):
    for j in range(len(b)):
        print("%.2f"%b[i][j],end =' ') #each element of the matrix is rounded upto 2 pl
aces of decimal
    print()

#verification of A*A[-1]=I, which verifies the inverse is correct or not

print("\n Verification of A*A[-1] = I")

list_d=matrix_mul(list_C,b)
for i in range(len(list_d)):
    for j in range(len(list_d)):
        print("%.2f"%list_d[i][j],end =' ') #each element of the matrix is rounded upto
2 places of decimal
    print()
```

```
The Inverse of the given matrix is A[-1]:
-0.33 0.33 0.33
-0.17 0.17 0.67
1.33 -0.33 -1.33

 Verification of A*A[-1] = I
1.00 0.00 0.00
0.00 1.00 0.00
0.00 0.00 1.00
```

In [3]:

```
#Use Gauss-Jordon elimination to determine the determinant of the matrix,

#              1 4 2 3
#              0 1 4 4
#             -1 0 1 0
#              2 0 4 1



#importing the determinant calculator function from library
from My_Lib import determinant_calc

#Readable augmented matrix form:

list_C=[]
with open("matrix4.txt") as matC:
    for k in matC:
        list_C.append(list(map(float, k.split())))


print("The determinant is:")
determinant_calc(list_C)
```

```
The determinant is:
```

Out[3]:

```
65.0
```