Experiment 3b:

Aim: \ To find the inverse of a matrix using Gaussian elimination method.

Software packages used: \ language- Python3, packages- Numpy

Theory: \ Gaussian elimination method or row reduction method, can be used to find the inverse of a matrix using the relation $AA^{-1} = I$. For a matrix

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1n} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2n} \\ A_{31} & A_{32} & A_{33} & \dots & A_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & A_{n3} & \dots & A_{nn} \end{bmatrix}$$

by augmenting the matrix A with the identity matrix $I_{n\times n}$

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1n} & 1 & 0 & 0 & \dots & 0 \\ A_{21} & A_{22} & A_{23} & \dots & A_{2n} & 0 & 1 & 0 & \dots & 0 \\ A_{31} & A_{32} & A_{33} & \dots & A_{3n} & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & A_{n3} & \dots & A_{nn} & 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

and reducing it to a row-echelon form using elementary row operations, and then using back substitution, we can find out the inverse matrix A^{-1}.

Code:

In [ ]:
```
# importing required packages
import numpy as np
import sys
```

In [ ]:
```
# reading dimension of square matrix A
n = int(input('Enter dimension of square matrix A, n: '))
```

In [ ]:
```
# Making numpy array of n x n+1 size and initializing to zero for storing augmented matrix
A = np.zeros((n, 2*n))

# Reading augmented matrix coefficients
print('Enter Matrix Coefficients:')
for i in range(n):
    for j in range(n):
        A[i][j] = float(input( 'A'+str(i+1)+ str(j+1)+'='))

for i in range(n):
        A[i][n+i] = 1
```

In [ ]:
```
#### Applying Gauss Elimination
for i in range(n):
    # Checking for zero pivot
    if A[i][i] == 0.0:
        sys.exit('Divide by zero detected!')

    # Eliminating subdiagonal elements
    for j in range(i+1, n):
        ratio = A[j][i]/A[i][i]

        # Subtracting row i from row j
        for k in range(2*n):
            A[j][k] = A[j][k] - ratio * A[i][k]
```

In [ ]:
```
# Back Substitution

for i in range(n-1,-1,-1):
    for j in range(i+1,n):
        ratio = A[i][j]/A[j][j]
        for k in range(n+n):
            A[i][k] = (A[i][k] - ratio * A[j][k])
```

```python
for i in range(n):
    for j in range(2*n-1,-1,-1):
        A[i][j]= A[i][j]/A[i][i]
```

In [ ]:

```python
print("inverse of matrix A is: \n")
for i in range(n):
    for j in range(n,2*n):
        print(A[i][j], end="\t")
    print("\n")
```

Observation and Result:

Input =

Output =