1. Please answer questions 1 - 10 refer to the following arrays. If the operation cannot be done, say "not possible".

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad C = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad D = \begin{pmatrix} -1 & 2 & 1 \\ 1 & 0 & 1 \\ 2 & 1 & 4 \end{pmatrix}$$

1. A + B

In [6]:

```python
import numpy as np
A = np.array([[1,1],[1,0],[1,0]])
B = np.array([[1,1],[1,0],[0,1]])
print(A+B)
```

```
[[2 2]
 [2 0]
 [1 1]]
```

2. C + B

Can not do it. The dimension doesn't match.

3. A - B

In [7]:

```python
import numpy as np
A = np.array([[1,1],[1,0],[1,0]])
B = np.array([[1,1],[1,0],[0,1]])
print(A-B)
```

```
[[ 0  0]
 [ 0  0]
 [ 1 -1]]
```

4. C · D (matrix multiplication)

In [8]:

```python
import numpy as np
C = np.array([[0,1,1],[1,1,0],[1,0,1]])
D = np.array([[-1,2,1],[1,0,1],[2,1,4]])
print(np.matmul(C,D))
```

```
[[3 1 5]
 [0 2 2]
 [1 3 5]]
```

5. 2B

In [9]:
```
import numpy as np
B = np.array([[1,1],[1,0],[0,1]])
print(2*B)
```

```
[[2 2]
 [2 0]
 [0 2]]
```

6. B^T

In [11]:
```
import numpy as np
B = np.array([[1,1],[1,0],[0,1]])
```

In [12]:
```
B_transpose = np.transpose(B)
print(B_transpose)
```

```
[[1 1 0]
 [1 0 1]]
```

7. A ∨ B

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

8. A ∧ B

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}$$

9. $C \times A$ (Boolean matrix multiplication)

$\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}$

10. Draw a 5×5 Identify matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

```
In [18]: np.eye(5)
```

```
Out[18]: array([[1., 0., 0., 0., 0.],
                [0., 1., 0., 0., 0.],
                [0., 0., 1., 0., 0.],
                [0., 0., 0., 1., 0.],
                [0., 0., 0., 0., 1.]])
```

2. If A is a symmetric matrix, find u, v, and w:

$$A = \begin{pmatrix} 2 & w & u \\ 7 & 0 & v \\ 1 & -3 & 4 \end{pmatrix}$$

w = 7
u = 1
v = -3

3. If A is a $n \times n$ invertible matrix, please use gaussian elimination method to find A^{-1}

$$A = \begin{pmatrix} 2 & w & u \\ 7 & 0 & v \\ 1 & -3 & 4 \end{pmatrix}$$

```
In [19]: A = np.array([[2,7,1],[7,0,-3],[1,-3,4]])
         print(A)
```

```
[[ 2  7  1]
 [ 7  0 -3]
 [ 1 -3  4]]
```

```
In [20]: I = np.eye(3)
         print(I)
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

In [21]:

```
aut = np.hstack([A,I])
print(aut)
```

```
[[ 2.  7.  1.  1.  0.  0.]
 [ 7.  0. -3.  0.  1.  0.]
 [ 1. -3.  4.  0.  0.  1.]]
```

In [22]:

```
def gauss_method(aug_mat):
    # num of rows/unknown variables
    n_row,n_col = aug_mat.shape
    for i in range(n_row):

        divisor = aug_mat[i,i]
        for j in range(n_col):
            aug_mat[i,j] /= divisor

        for j in range(n_row):
            if j != i:
                multiplier = -aug_mat[j,i]
                aug_mat[j,:] += multiplier*aug_mat[i,:]
    return aug_mat
```

In [23]:

```
print(gauss_method(aut))
```

```
[[ 1.          0.          0.          0.03515625  0.12109375  0.0820312
5]
 [-0.          1.          0.          0.12109375 -0.02734375 -0.0507812
5]
 [ 0.          0.          1.          0.08203125 -0.05078125  0.1914062
5]]
```

In [ ]: