

```
In [1]: import numpy as np
n = 7
k = 4
P = np.array([[0,0,1],[1,1,1],[1,1,0],[1,0,1]])
print("Parity: ", P)
```

```
Parity: [[0 0 1]
 [1 1 1]
 [1 1 0]
 [1 0 1]]
```

```
In [2]: I = np.eye(k)
print(I)
```

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

```
In [3]: G = np.hstack((I,P))
print(G)
```

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

```
In [4]: m = np.arange(2 ** k)
m
```

```
Out[4]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15])
```

```
In [5]: D = np.zeros((2 ** k, k))
D
```

```
Out[5]: array([[0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.],
 [0., 0., 0., 0.]])
```

```
In [6]: for i in m:
    z = '{0:04b}'.format(i)
    zs = [int(i) for i in z]
    D[i,:] = zs
```

D

```
Out[6]: array([[0., 0., 0., 0.],
               [0., 0., 0., 1.],
               [0., 0., 1., 0.],
               [0., 0., 1., 1.],
               [0., 1., 0., 0.],
               [0., 1., 0., 1.],
               [0., 1., 1., 0.],
               [0., 1., 1., 1.],
               [1., 0., 0., 0.],
               [1., 0., 0., 1.],
               [1., 0., 1., 0.],
               [1., 0., 1., 1.],
               [1., 1., 0., 0.],
               [1., 1., 0., 1.],
               [1., 1., 1., 0.],
               [1., 1., 1., 1.]])
```

```
In [7]: C = np.dot(D,G)
        print("Codeword: ",C)
```

```
Codeword: [[0. 0. 0. 0. 0. 0. 0.]
           [0. 0. 0. 1. 1. 0. 1.]
           [0. 0. 1. 0. 1. 1. 0.]
           [0. 0. 1. 1. 2. 1. 1.]
           [0. 1. 0. 0. 1. 1. 1.]
           [0. 1. 0. 1. 2. 1. 2.]
           [0. 1. 1. 0. 2. 2. 1.]
           [0. 1. 1. 1. 3. 2. 2.]
           [1. 0. 0. 0. 0. 0. 1.]
           [1. 0. 0. 1. 1. 0. 2.]
           [1. 0. 1. 0. 1. 1. 1.]
           [1. 0. 1. 1. 2. 1. 2.]
           [1. 1. 0. 0. 1. 1. 2.]
           [1. 1. 0. 1. 2. 1. 3.]
           [1. 1. 1. 0. 2. 2. 2.]
           [1. 1. 1. 1. 3. 2. 3.]])
```

```
In [8]: C = C % 2
        C
```

```
Out[8]: array([[0., 0., 0., 0., 0., 0., 0.],
               [0., 0., 0., 1., 1., 0., 1.],
               [0., 0., 1., 0., 1., 1., 0.],
               [0., 0., 1., 1., 0., 1., 1.],
               [0., 1., 0., 0., 1., 1., 1.],
               [0., 1., 0., 1., 0., 1., 0.],
               [0., 1., 1., 0., 0., 0., 1.],
               [0., 1., 1., 1., 1., 0., 0.],
               [1., 0., 0., 0., 0., 0., 1.],
               [1., 0., 0., 1., 1., 0., 0.],
               [1., 0., 1., 0., 1., 1., 1.],
               [1., 0., 1., 1., 0., 1., 0.],
               [1., 1., 0., 0., 1., 1., 0.],
               [1., 1., 0., 1., 0., 1., 1.],
               [1., 1., 1., 0., 0., 0., 0.],
               [1., 1., 1., 1., 1., 0., 1.]])
```

```
In [9]: C = C.astype(int)
        C
```

```
Out[9]: array([[0, 0, 0, 0, 0, 0, 0],
               [0, 0, 0, 1, 1, 0, 1],
               [0, 0, 1, 0, 1, 1, 0],
               [0, 0, 1, 1, 0, 1, 1],
               [0, 1, 0, 0, 1, 1, 1],
               [0, 1, 0, 1, 0, 1, 0],
               [0, 1, 1, 0, 0, 0, 1],
               [0, 1, 1, 1, 1, 0, 0],
               [1, 0, 0, 0, 0, 0, 1],
               [1, 0, 0, 1, 1, 0, 0],
               [1, 0, 1, 0, 1, 1, 1],
               [1, 0, 1, 1, 0, 1, 0],
               [1, 1, 0, 0, 1, 1, 0],
               [1, 1, 0, 1, 0, 1, 1],
               [1, 1, 1, 0, 0, 0, 0],
               [1, 1, 1, 1, 1, 0, 1]])
```

## DECODING

```
In [10]: index = 3
         loc = 1
         R = C[index, :]
         R[loc] = int(not R[loc])
         R
```

```
Out[10]: array([0, 1, 1, 1, 0, 1, 1])
```

```
In [11]: I = np.eye(n-k)
         I
```

```
Out[11]: array([[1., 0., 0.],
               [0., 1., 0.],
               [0., 0., 1.]])
```

```
In [12]: H = np.hstack((P.T, I))
```

```
H
```

```
Out[12]: array([[0., 1., 1., 1., 1., 0., 0.],
                [0., 1., 1., 0., 0., 1., 0.],
                [1., 1., 0., 1., 0., 0., 1.]])
```

```
In [13]: S = np.dot(R,H.T).astype(int) % 2
S
```

```
Out[13]: array([1, 1, 1])
```

```
In [14]: X = H.T
for i in range(X.shape[1]):
    if np.array_equal(S,X[i,:]):
        R[i] = int(not R[i])
        print("Error in bit: ", i)
        break

print("Corrected C: ",R)
```

```
Error in bit: 1
Corrected C:  [0 0 1 1 0 1 1]
```