- Name: Sarvesh Surve
- Roll No. 64

Generate 10 random numbers using numpy Close

```
import numpy as np
np.random.seed(seed=0)
I = np.random.choice([0,1], 4)
W = np.random.choice([-1,1], 4)
print(f'Input vector:{I}, Weight vector:{W}')
```

```
Input vector:[0 1 1 0], Weight vector:[1 1 1 1]
```

```
dot = I @ W
print(f'Dot product: {dot}')
```

```
Dot product: 2
```

```
def linear_threshold_gate(dot: int, T: float) -> int:
    if dot >= T:
        return 1
    else:
        return 0
```

```
T = 1
activation = linear_threshold_gate(dot, T)
print(f'Activation: {activation}')
```

```
Activation: 1
```

+ Code + Text

```
T = 3
activation = linear_threshold_gate(dot, T)
print(f'Activation: {activation}')
```

```
Activation: 0
```

```
input_table = np.array([
    [0,0,0],
    [0,1,1],
    [1,0,1],
    [1,1,0]
])
```

```
print(f'input table:\n{input_table}')
```

```
input table:
[[0 0 0]
 [0 1 1]
 [1 0 1]
 [1 1 0]]
```

```
weights = np.array([1, -1, 1])
print(f'weights: {weights}')
```

```
weights: [ 1 -1  1]
```

```
dot_products = input_table @ weights
print(f'Dot products: {dot_products}')
```

```
Dot products: [0 0 2 0]
```

```
def linear_threshold_gate(dot: int, T: float) -> int:
    if dot >= T:
        return 1
    else:
        return 0
```

```
T = 1
for i in range(0,4):
    activation = linear_threshold_gate(dot_products[i], T)
    print(f'Activation: {activation}')
```

```
Activation: 0
Activation: 0
Activation: 1
Activation: 0
```