```python
import copy
def swap(sArray, i , j) -> None:
    sArray[i], sArray[j] = sArray[j], sArray[i]

def keyScheduling(sArray: list, tArray: list) -> list:
    j: int = 0
    size: int = len(sArray)
    for i in range(size):
        j = (j + sArray[i] + tArray[i])
        while j < 0:
            j += size
        j = j % size
        swap(sArray, i, j)

    return sArray

def Encryption(plainText: list, tArray: list) -> None:
    encrypt: list[int] = list()
    for p, k in zip(plainText, tArray):
        encrypt.append(p ^ k)
    return encrypt

def streamGeneration(sArray: list, plainText: list) -> list:
    print(f'{sArray=}')
    slen: int = len(sArray)
    size: int = len(plainText)
    i: int = 0
    j: int = 0
    key: list = list()
    for i in range(1, size + 1):
        j = (j + sArray[i]) % slen
        swap(sArray, i , j)
        t = (sArray[i] + sArray[j]) % slen
        print(f'{i=}, {j=}, {t=}, {sArray[t]=}, {sArray=}')
        key.append(sArray[t])
    return key



def main() -> None:
    try:
        sArray = list(map(int, input("Enter S-Array (in DEC format X X ...): ").split(" ")))
        plainText = list(map(int, input("Enter PlainText (in DEC format X X ...): ").split(" ")))
        key = list(map(int, input("Enter Key Array (in DEC format X X ...): ").split(" ")))
        tArray: list[int] = list()
        i: int = 0
        keysize: int = len(key)
        while(len(sArray) != len(tArray)):
            tArray.append(key[i])
            i = (i+1) % keysize

        sArray = keyScheduling(sArray, tArray)
        key = streamGeneration(sArray, plainText)
        encrypt = Encryption(plainText, key)

        print(f'Encrypted PlainText: {encrypt}')

    except Exception as e:
        print("Error in Handling ", e)


if __name__ == '__main__':
    main()
```

OUTPUT:

```
Enter S - Array(in DEC format X X ...): 0 1 2 3 4 5 6 7
Enter PlainText (in DEC format X X ...): 1 2 2 2
Enter Key Array (in DEC format X X ...): 5 1 0 1
sArray=[1, 2, 3, 0, 6, 7, 4, 5]
i=1, j=2, t=5, sArray[t]=7, sArray=[1, 3, 2, 0, 6, 7, 4, 5]
i=2, j=4, t=0, sArray[t]=1, sArray=[1, 3, 6, 0, 2, 7, 4, 5]
i=3, j=4, t=2, sArray[t]=6, sArray=[1, 3, 6, 2, 0, 7, 4, 5]
i=4, j=4, t=0, sArray[t]=1, sArray=[1, 3, 6, 2, 0, 7, 4, 5]
Encrypted PlainText: [6, 3, 4, 3]
```