Lab 6 - CRC-16 Adithya N 1BM18CS128

Aim: To write a program to demonstrate error checking with CRC

Procedure:
1. Read data word (length n) and gen. polynomial (length k)
2. Shift data word to left by k-1 bits
3. Perform modulo 2 division by dividing shifted data by polynomial
4. Append remainder to original dataword to get encoded dataword
5. Upon receiving the data if a bit is changed or misplaced, performing mod 2 division will have a remainder syndrome showing an error, else it shows 0 with no errors.

Code:

```
def encode(data, key):
    n1 = len(bin(key)) - 3
    dividend = data << n1
    rem = xordiv(dividend, key)
    n2 = len(bin(rem)) - 2
    suffix = '0'*(n1-n2) + bin(rem)[2:] if rem else '0'*n1
    return int(bin(data)[2:] + suffix, 2)
```

```
def decode(data, key):
    rem = xordiv(data, key)
    print(f"Error is {bin(rem)[2:]}" if rem
          else "No errors.")
```

```
def xordiv(data, key):
    a = data
    b = key
    while len(bin(a)) >= len(bin(key)):
        b = key << len(bin(a)) - len(bin(key))
        a = a ^ b
    return a

data = int(input(), 2)
p = int(input(), 2)
print("Encoded", bin(encode(data, p))[2:])
response = int(input(), 2)
decode(response, p)
```