PROGRAM : 6 — ERROR DETECTION USING CRC (16 bits)

CODE

```
import hashlib

def xor(a,b):
    result = []
    for i in range(1, len(b)):
        if a[i] == b[i]:
            result.append('0')
        else
            result.append('1')
    result ''.join(result)

def mod2div(dividend, divisor):
    pick = len(divisor)
    tmp = dividend[0 : pick]
    while pick < len(dividend):
        if tmp[0] == '1':
            tmp = xor(divisor, tmp) + dividend[pick]
        else:
            tmp = xor('0' * pick, tmp) + dividend[pick]
        pick += 1
    if tmp[0] == '1':
        tmp = xor(divisor, tmp)
    else:
        tmp = xor('0'* pick, tmp)
    checkword = tmp
    return checkword

def encodeData(data, key):
    l_key = len(key)
    appended_data = data + '0' * (l_key - 1)
```

```
    codeword = data + remainder
    return codeword

def decodeData (code, key):
    remainder = mod2div (code, key)
    return remainder

data = input("enter data :")
print ("dataword :" + str (data))

key = "100010000001000001"
print ("generating polynomial" + key)
codeword = encodeData(data, key)
print ("Checksum :", codeword)
print (" transmitted codeword " + str(codeword))
code = input ("enter transmitted codeword: ")

recieved_data = int (decodeData (code, key))

if recieved_data == 0 :
    print ("no error")
else:
    print ("error")
    print (recieved_data)
```