DALHOUSIE UNIVERSITY

Faculty of Computer Science

CSCI 6704 – Advanced Topics in Networks

Name: Arka Ghosh
Banner ID: B00911033
Assignment: 03 (Question 1)

Sample Input and Output for Question 1

The resulting outputs with the screenshot have been added below:

<u>Test Case 1:</u>

```
Enter the message M(x):1101011
Enter the Reference Polynomial G(x):1101
Message Transmitted P(x): 1101011010
Remainder: 010
Receiver Side: The transmitted message is not error-free
The bit string after adding random error to the P(x):  1101011011
Receiver Side: The message after random bit shifting is not error-free
```

jupyter Q1 Last Checkpoint: Last Friday at 1:21 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```
    appended_message = message + g_
    #print(appended_message)

    remainder = CRC_Division(appended_message, g)
    #Message transmitted P(x)
    trans_message = message + remainder
    #swapped_bit = int(trans_message) ^ 1
    #print(str(swapped_bit))

    print("Message Transmitted P(x):", trans_message)
    print("Remainder:", remainder)
    return trans_message


def ReceiverSide(message, g):
    #check the remainder for the transmitted message
    remainder1 = CRC_Division(message,g)
    #print(remainder)
    if remainder1 == len(g)*0:
        print("Receiver Side: The transmitted message is error-free")
    else:
        print("Receiver Side: The transmitted message is not error-free")

    #XOR the transmitted message with 1 to add random error to the last one bits
    swapped_bit = int(message) ^ 1
    swapped_bit_message = str(swapped_bit)
    print("The bit string after adding random error to the P(x): ", swapped_bit)

    remainder2 = CRC_Division(swapped_bit_message, g)
    if remainder2 == len(g)*0:
        print("Receiver Side: The message after random bit shifting is error-free")
    else:
        print("Receiver Side: The message after random bit shifting is not error-free")


message = input("Enter the message M(x):")
polynomial = input("Enter the Reference Polynomial G(x):")

tranmitted_message = SenderSide(message,polynomial)
#print(tranmitted_message)
received_message = ReceiverSide(tranmitted_message, polynomial)
```

```
Enter the message M(x):1101011
Enter the Reference Polynomial G(x):1101
Message Transmitted P(x): 1101011010
Remainder: 010
Receiver Side: The transmitted message is not error-free
The bit string after adding random error to the P(x):  1101011011
Receiver Side: The message after random bit shifting is not error-free
```

Test Case 2:

```
Enter the message M(x):10110011
Enter the Reference Polynomial G(x):11001
Message Transmitted P(x): 101100110100
Remainder: 0100
Receiver Side: The transmitted message is not error-free
The bit string after adding random error to the P(x):  101100110101
Receiver Side: The message after random bit shifting is not error-free
```

jupyter Q1 Last Checkpoint: Last Friday at 1:21 AM (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
    #Message transmitted P(x)
    trans_message = message + remainder
    #swapped_bit = int(trans_message) ^ 1
    #print(str(swapped_bit))

    print("Message Transmitted P(x):", trans_message)
    print("Remainder:", remainder)
    return trans_message


def ReceiverSide(message, g):
    #check the remainder for the transmitted message
    remainder1 = CRC_Division(message,g)
    #print(remainder)
    if remainder1 == len(g)*0:
        print("Receiver Side: The transmitted message is error-free")
    else:
        print("Receiver Side: The transmitted message is not error-free")

    #XOR the transmitted message with 1 to add random error to the last one bits
    swapped_bit = int(message) ^ 1
    swapped_bit_message = str(swapped_bit)
    print("The bit string after adding random error to the P(x): ", swapped_bit)

    remainder2 = CRC_Division(swapped_bit_message, g)
    if remainder2 == len(g)*0:
        print("Receiver Side: The message after random bit shifting is error-free")
    else:
        print("Receiver Side: The message after random bit shifting is not error-free")


message = input("Enter the message M(x):")
polynomial = input("Enter the Reference Polynomial G(x):")

tranmitted_message = SenderSide(message,polynomial)
#print(tranmitted_message)
received_message = ReceiverSide(tranmitted_message, polynomial)
```

```
Enter the message M(x):10110011
Enter the Reference Polynomial G(x):11001
Message Transmitted P(x): 101100110100
Remainder: 0100
Receiver Side: The transmitted message is not error-free
The bit string after adding random error to the P(x):  101100110101
Receiver Side: The message after random bit shifting is not error-free
```

Test Case 3:

```
Enter the message M(x):1001100
Enter the Reference Polynomial G(x):1100
Message Transmitted P(x): 1001100100
Remainder: 100
Receiver Side: The transmitted message is not error-free
The bit string after adding random error to the P(x):  1001100101
Receiver Side: The message after random bit shifting is not error-free
```

jupyter Q1 Last Checkpoint: Last Friday at 1:21 AM (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

```python
    return trans_message


def ReceiverSide(message, g):
    #check the remainder for the transmitted message
    remainder1 = CRC_Division(message,g)
    #print(remainder)
    if remainder1 == len(g)*0:
        print("Receiver Side: The transmitted message is error-free")
    else:
        print("Receiver Side: The transmitted message is not error-free")

    #XOR the transmitted message with 1 to add random error to the last one bits
    swapped_bit = int(message) ^ 1
    swapped_bit_message = str(swapped_bit)
    print("The bit string after adding random error to the P(x): ", swapped_bit)

    remainder2 = CRC_Division(swapped_bit_message, g)
    if remainder2 == len(g)*0:
        print("Receiver Side: The message after random bit shifting is error-free")
    else:
        print("Receiver Side: The message after random bit shifting is not error-free")


message = input("Enter the message M(x):")
polynomial = input("Enter the Reference Polynomial G(x):")

tranmitted_message = SenderSide(message,polynomial)
#print(tranmitted_message)
received_message = ReceiverSide(tranmitted_message, polynomial)
```

```
Enter the message M(x):1001100
Enter the Reference Polynomial G(x):1100
Message Transmitted P(x): 1001100100
Remainder: 100
Receiver Side: The transmitted message is not error-free
The bit string after adding random error to the P(x):  1001100101
Receiver Side: The message after random bit shifting is not error-free
```

In [ ]: