CN LAB

```
def crc(input_msg, poly, mode):
    # Prepare the output message
    output_msg = input_msg
    if mode:
        output_msg += '0' * (len(poly) - 1)

    # Perform XOR on the message with the selected polynomial
    output_msg = list(output_msg)
    poly_len = len(poly)
    for i in range(len(input_msg)):
        if output_msg[i] == '1':
            for j in range(poly_len):
                if i + j < len(output_msg):
                    output_msg[i + j] = '0' if output_msg[i + j] == poly[j] else '1'

    # Check for errors and return 0 if error detected
    return '1' not in output_msg[-(poly_len-1):]

def main():
    poly = "10001000000100001"

    # Input the original message
    input_msg = input("Enter the input message in binary: ")

    # Calculate transmitted message
    crc(input_msg, poly, 1)
    transmitted_msg = input_msg + '0' * (len(poly) - 1)

    print("The transmitted message is:", transmitted_msg)

    # Input the received message
    received_msg = input("Enter the received message in binary: ")

    # Check for errors in received message
    if crc(received_msg, poly, 0):
        print("No error in data")
    else:
        print("Error in data transmission has occurred")

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

```
Enter the input message in binary: 11111
The transmitted message is: 111110000000000000000
Enter the received message in binary: 11111
No error in data
```

```
def main():
    # Initialize variables
    storage = 0
    no_of_queries = 4
    bucket_size = 10
    input_pkt_size = 4
    output_pkt_size = 1

    for i in range(no_of_queries):
        # Calculate space left in the bucket
        size_left = bucket_size - storage

        if input_pkt_size <= size_left:
            # Update storage if space is available
            storage += input_pkt_size
        else:
            # If not enough space, print packet loss
            print(f"Packet loss = {input_pkt_size}")

        # Print current buffer status
        print(f"Buffer size = {storage} out of bucket size = {bucket_size}")

        # Update storage by removing output packets
        storage -= output_pkt_size

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

```
Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10
Packet loss = 4
Buffer size = 9 out of bucket size = 10
```