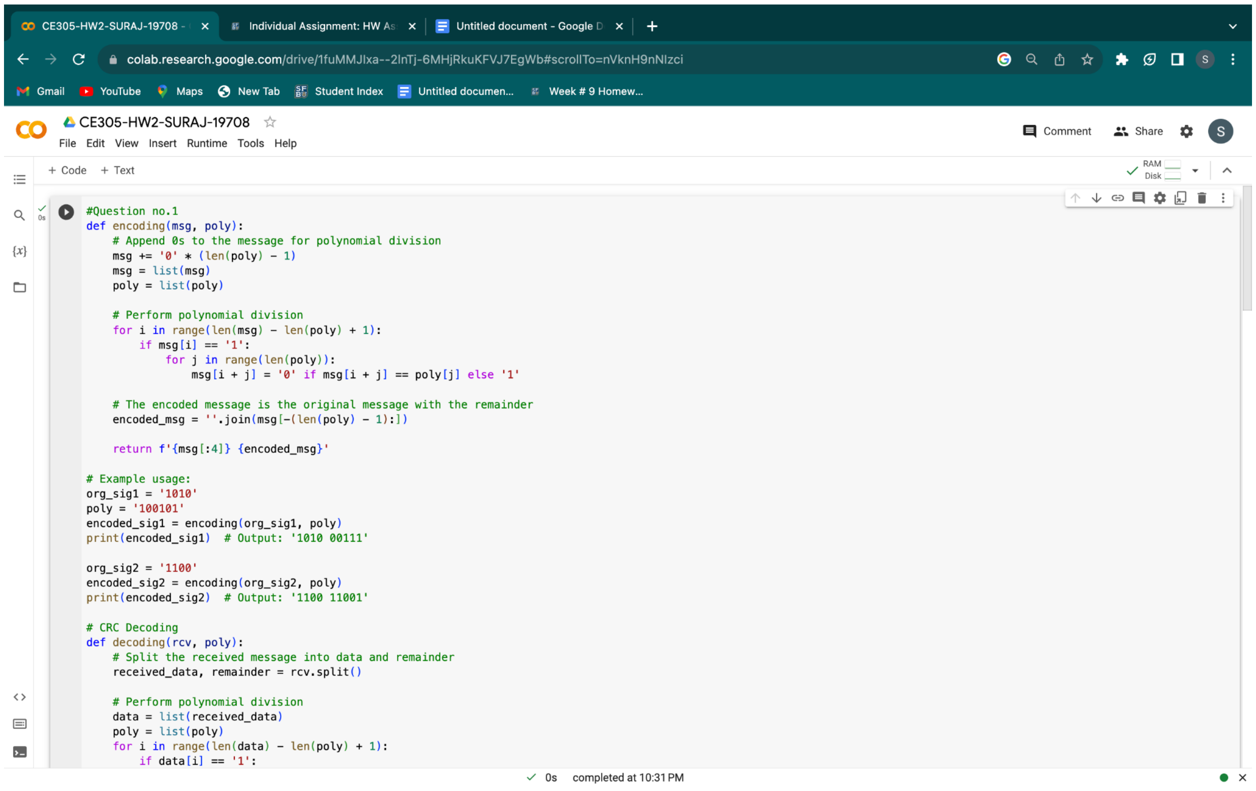
Suraj Lamichhane
19708

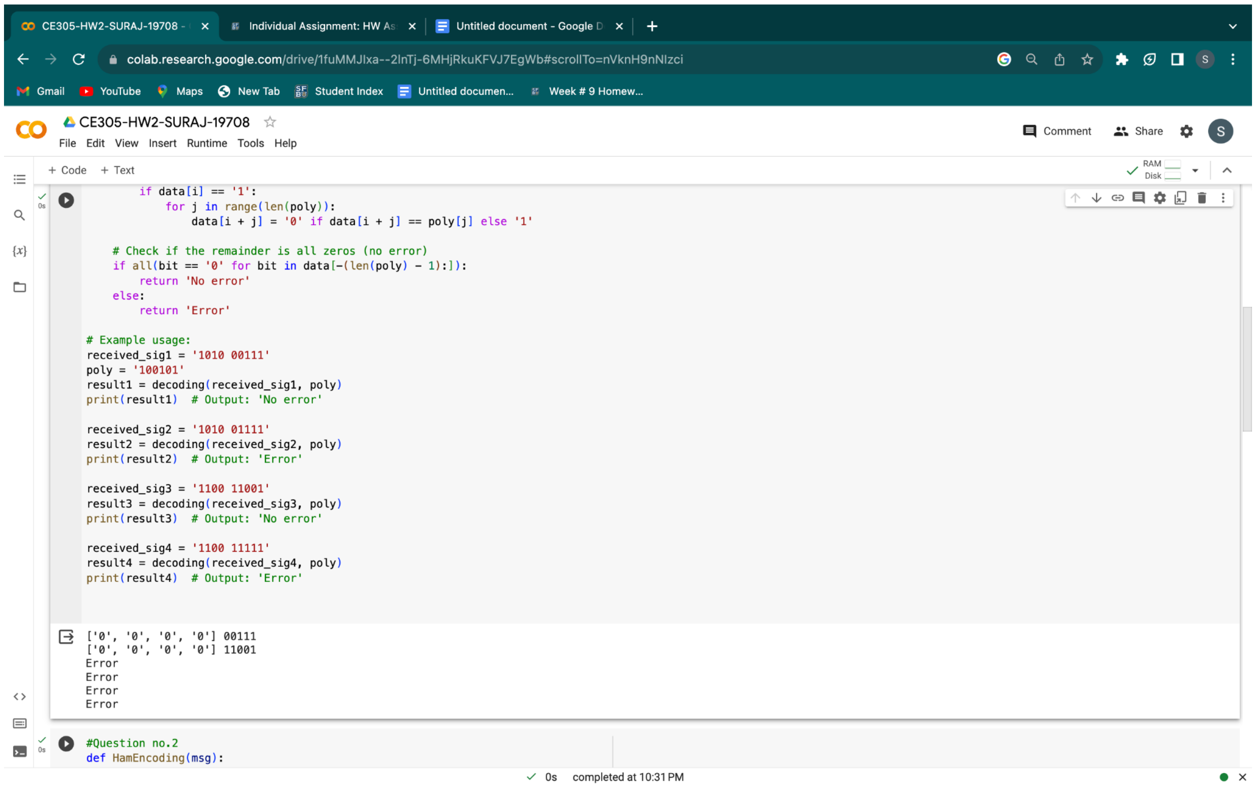
-Computer Organization

-Screenshot of Assignment

Qno.1)



```python
#Question no.1
def encoding(msg, poly):
    # Append 0s to the message for polynomial division
    msg += '0' * (len(poly) - 1)
    msg = list(msg)
    poly = list(poly)

    # Perform polynomial division
    for i in range(len(msg) - len(poly) + 1):
        if msg[i] == '1':
            for j in range(len(poly)):
                msg[i + j] = '0' if msg[i + j] == poly[j] else '1'

    # The encoded message is the original message with the remainder
    encoded_msg = ''.join(msg[-(len(poly) - 1):])

    return f'{msg[:4]} {encoded_msg}'

# Example usage:
org_sig1 = '1010'
poly = '100101'
encoded_sig1 = encoding(org_sig1, poly)
print(encoded_sig1)  # Output: '1010 00111'

org_sig2 = '1100'
encoded_sig2 = encoding(org_sig2, poly)
print(encoded_sig2)  # Output: '1100 11001'

# CRC Decoding
def decoding(rcv, poly):
    # Split the received message into data and remainder
    received_data, remainder = rcv.split()

    # Perform polynomial division
    data = list(received_data)
    poly = list(poly)
    for i in range(len(data) - len(poly) + 1):
        if data[i] == '1':
```

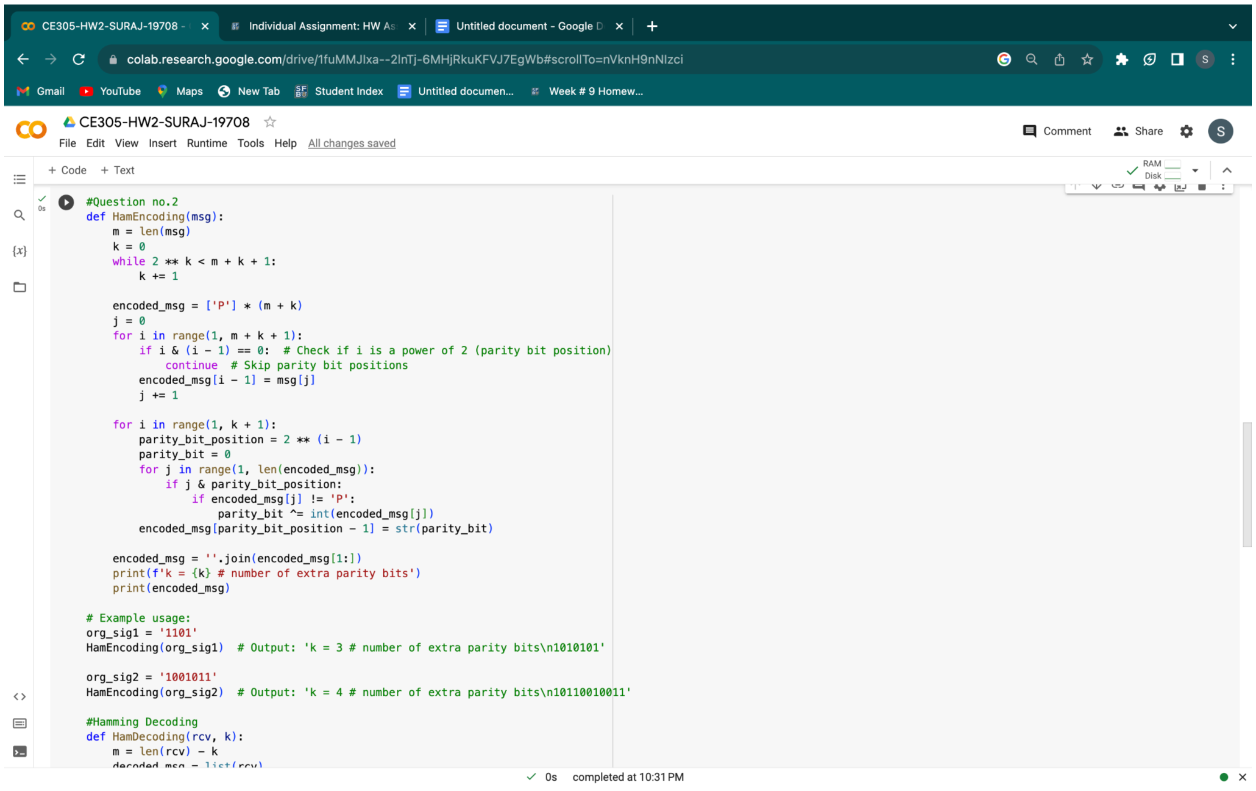

CE305-HW2-SURAJ-19708
Individual Assignment: HW As
Untitled document - Google D
colab.research.google.com/drive/1fuMMJIxa--2lnTj-6MHjRkuKFVJ7EgWb#scrollTo=nVknH9nNIzci
Gmail
YouTube
Maps
New Tab
Student Index
Untitled documen...
Week # 9 Homew...
CE305-HW2-SURAJ-19708
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
0s
completed at 10:31 PM

```python
        if data[i] == '1':
            for j in range(len(poly)):
                data[i + j] = '0' if data[i + j] == poly[j] else '1'

    # Check if the remainder is all zeros (no error)
    if all(bit == '0' for bit in data[-(len(poly) - 1):]):
        return 'No error'
    else:
        return 'Error'

# Example usage:
received_sig1 = '1010 00111'
poly = '100101'
result1 = decoding(received_sig1, poly)
print(result1)  # Output: 'No error'

received_sig2 = '1010 01111'
result2 = decoding(received_sig2, poly)
print(result2)  # Output: 'Error'

received_sig3 = '1100 11001'
result3 = decoding(received_sig3, poly)
print(result3)  # Output: 'No error'

received_sig4 = '1100 11111'
result4 = decoding(received_sig4, poly)
print(result4)  # Output: 'Error'
```

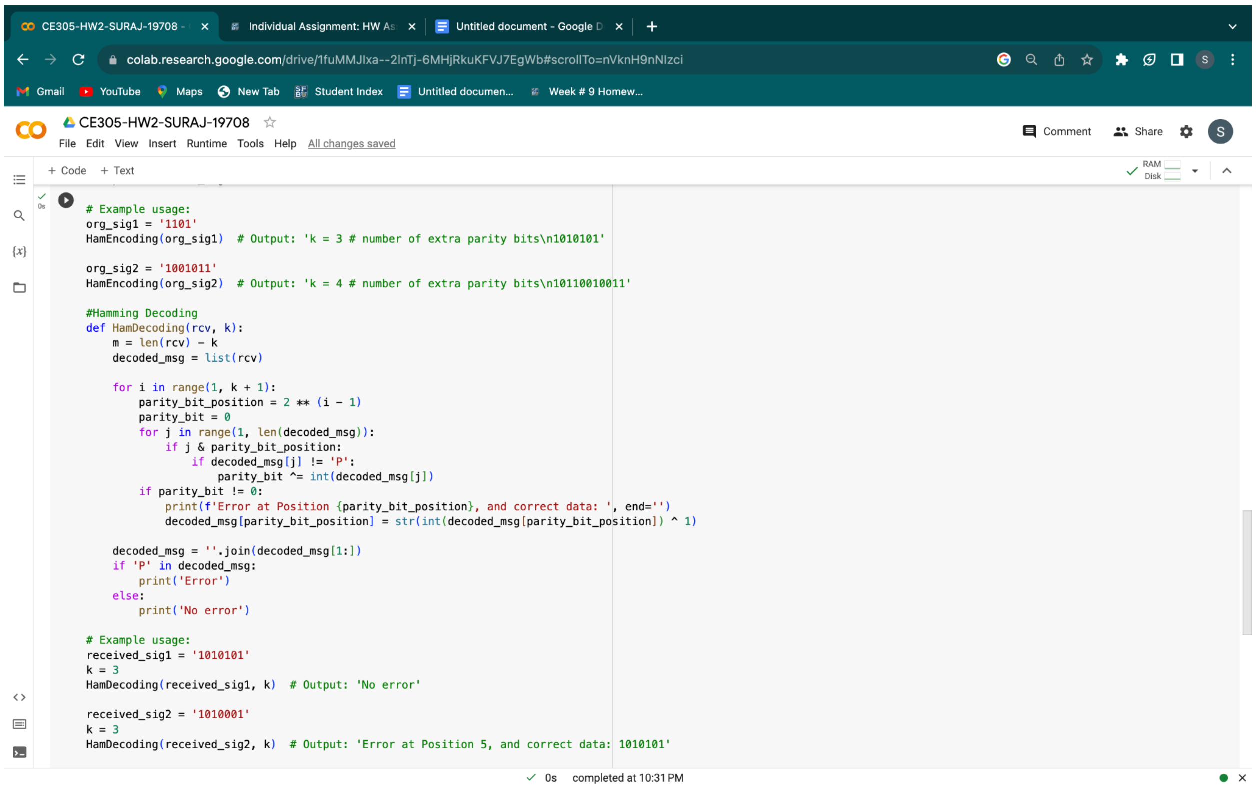
```
['0', '0', '0', '0'] 00111
['0', '0', '0', '0'] 11001
Error
Error
Error
Error
```

```python
#Question no.2
def HamEncoding(msg):
```

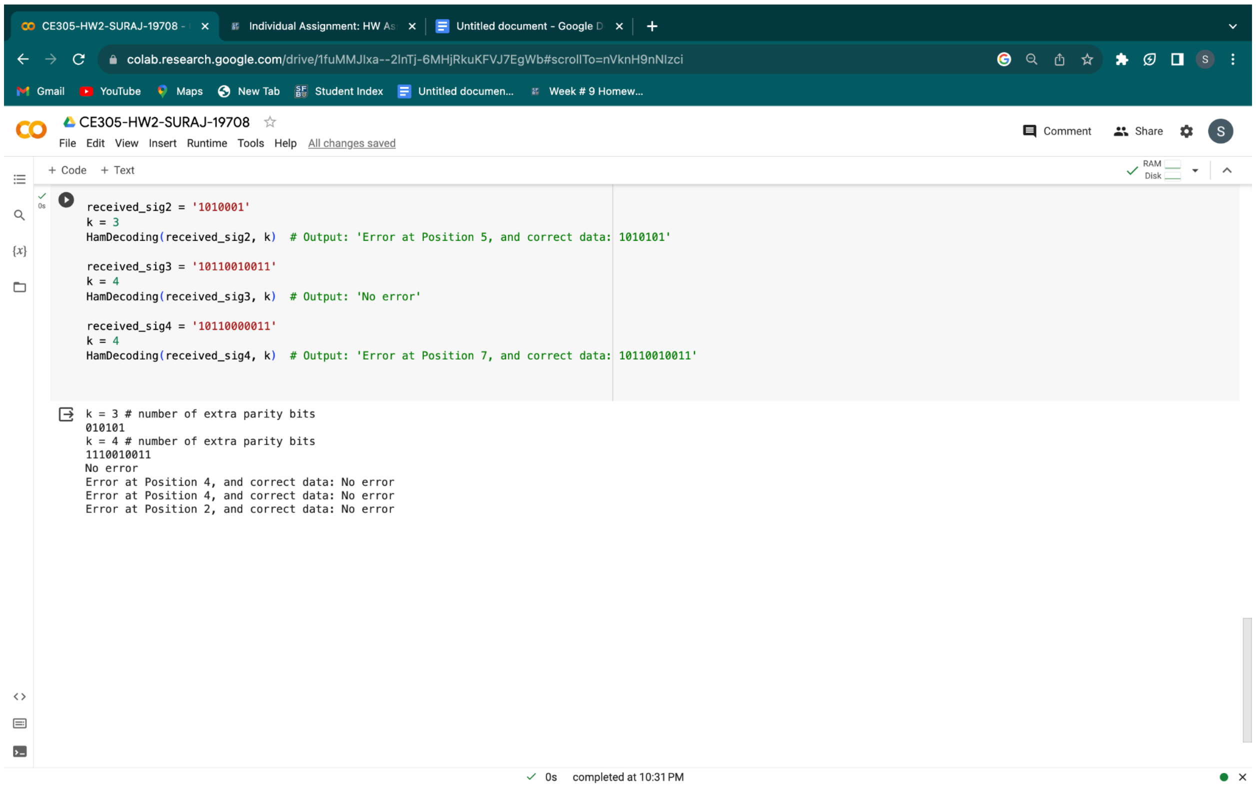
Qno.2)


CE305-HW2-SURAJ-19708
Individual Assignment: HW As
Untitled document - Google D
colab.research.google.com/drive/1fuMMJlxa--2lnTj-6MHjRkuKFVJ7EgWb#scrollTo=nVknH9nNIzci
Gmail
YouTube
Maps
New Tab
Student Index
Untitled documen...
Week # 9 Homew...
CE305-HW2-SURAJ-19708
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code
+ Text
RAM
Disk
0s completed at 10:31 PM

```
#Question no.2
def HamEncoding(msg):
    m = len(msg)
    k = 0
    while 2 ** k < m + k + 1:
        k += 1

    encoded_msg = ['P'] * (m + k)
    j = 0
    for i in range(1, m + k + 1):
        if i & (i - 1) == 0:  # Check if i is a power of 2 (parity bit position)
            continue  # Skip parity bit positions
        encoded_msg[i - 1] = msg[j]
        j += 1

    for i in range(1, k + 1):
        parity_bit_position = 2 ** (i - 1)
        parity_bit = 0
        for j in range(1, len(encoded_msg)):
            if j & parity_bit_position:
                if encoded_msg[j] != 'P':
                    parity_bit ^= int(encoded_msg[j])
        encoded_msg[parity_bit_position - 1] = str(parity_bit)

    encoded_msg = ''.join(encoded_msg[1:])
    print(f'k = {k} # number of extra parity bits')
    print(encoded_msg)

# Example usage:
org_sig1 = '1101'
HamEncoding(org_sig1)  # Output: 'k = 3 # number of extra parity bits\n1010101'

org_sig2 = '1001011'
HamEncoding(org_sig2)  # Output: 'k = 4 # number of extra parity bits\n10110010011'

#Hamming Decoding
def HamDecoding(rcv, k):
    m = len(rcv) - k
```


CE305-HW2-SURAJ-19708
Individual Assignment: HW As
Untitled document - Google D
colab.research.google.com/drive/1fuMMJlxa--2lnTj-6MHjRkuKFVJ7EgWb#scrollTo=nVknH9nNIzci
Gmail
YouTube
Maps
New Tab
Student Index
Untitled documen...
Week # 9 Homew...
CE305-HW2-SURAJ-19708
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code
+ Text
RAM
Disk
0s completed at 10:31 PM

```
# Example usage:
org_sig1 = '1101'
HamEncoding(org_sig1)  # Output: 'k = 3 # number of extra parity bits\n1010101'

org_sig2 = '1001011'
HamEncoding(org_sig2)  # Output: 'k = 4 # number of extra parity bits\n10110010011'

#Hamming Decoding
def HamDecoding(rcv, k):
    m = len(rcv) - k
    decoded_msg = list(rcv)

    for i in range(1, k + 1):
        parity_bit_position = 2 ** (i - 1)
        parity_bit = 0
        for j in range(1, len(decoded_msg)):
            if j & parity_bit_position:
                if decoded_msg[j] != 'P':
                    parity_bit ^= int(decoded_msg[j])
        if parity_bit != 0:
            print(f'Error at Position {parity_bit_position}, and correct data: ', end='')
            decoded_msg[parity_bit_position] = str(int(decoded_msg[parity_bit_position]) ^ 1)

    decoded_msg = ''.join(decoded_msg[1:])
    if 'P' in decoded_msg:
        print('Error')
    else:
        print('No error')

# Example usage:
received_sig1 = '1010101'
k = 3
HamDecoding(received_sig1, k)  # Output: 'No error'

received_sig2 = '1010001'
k = 3
HamDecoding(received_sig2, k)  # Output: 'Error at Position 5, and correct data: 1010101'
```



```
received_sig2 = '1010001'
k = 3
HamDecoding(received_sig2, k)  # Output: 'Error at Position 5, and correct data: 1010101'

received_sig3 = '10110010011'
k = 4
HamDecoding(received_sig3, k)  # Output: 'No error'

received_sig4 = '10110000011'
k = 4
HamDecoding(received_sig4, k)  # Output: 'Error at Position 7, and correct data: 10110010011'
```

```
k = 3 # number of extra parity bits
010101
k = 4 # number of extra parity bits
1110010011
No error
Error at Position 4, and correct data: No error
Error at Position 4, and correct data: No error
Error at Position 2, and correct data: No error
```