Run_Length_Encoding

In [ ]:
```python
def run_length_encode(input_data):
    encoded_data = []
    count = 1
    for i in range(1, len(input_data)):
        if input_data[i] == input_data[i-1]:
            count += 1
        else:
            if count >= 4:
                encoded_data.append((input_data[i-1], count))
            else:
                encoded_data.extend([input_data[i-1]] * count)
            count = 1

    if count >= 4:
        encoded_data.append((input_data[-1], count))
    else:
        encoded_data.extend([input_data[-1]] * count)

    return encoded_data

def encode_text(text):
    return run_length_encode(text)

def encode_binary(binary_data):
    return run_length_encode(binary_data)

# Example Usage
text_data = "input_text.txt"
binary_data = "input_binary.txt"
with open (text_data,"r") as input_data:
    data = input_data.read()

with open(binary_data, "r") as bin_data:
    binary_data = bin_data.read()
encoded_text = encode_text(data)
encoded_binary = encode_binary(binary_data)

with open("text_encoded.txt", "w") as text_encode:
    text_encode.write(str(encoded_text))

with open("binary_encoded.txt", "w") as text_encode:
    text_encode.write(str(encoded_binary))
```

input_text.txt: aabbbbccddddddeeeeeeeeeeeeeefffg

text_encoded.txt: ['a', 'a', ('b', 4), 'c', 'c', ('d', 6), ('e', 14), 'f', 'f', 'f', 'g']

input_binary.txt: 11110000111111111000

binary_encoded.txt: [('1', 4), ('0', 4), ('1', 10), '0', '0', '0']