```
clc
clear
close all
img = imread('lena.tif');
img_double = im2double(img);
img_uint8 = uint8(img_double * 255);
img_vector = reshape(img_uint8, 1, []);
img_bits = dec2bin(img_vector, 8);
img_bitstream = reshape(img_bits.', 1, []);
input = str2num(img_bitstream')'
```

```
input = 1×524288
     1     0     0     0     1     0     0     1     1     0     0     0     1···
```

```
max_rep = find_max_consecutive_repetition(input); %max_repetition = 23
bit_len = ceil(log2(max_rep)); %bit_len = 5
rle = rle_encode_bitstream(input,bit_len)
```

```
rle = 
'0001100110000110010000101001100001100100001010011000011000100001100010000110100000001100010001010110000101
```

```
compression_ratio = size(input)/size(rle)
```

```
compression_ratio = 0.3890
```

```
function max_rep = find_max_consecutive_repetition(bitstream)
    maxCount = 0;
    currentCount = 0;
    previousBit = NaN; % initialize with NaN to handle the case where the 
bitstream starts with a NaN value

    for i = 1:length(bitstream)
        bit = bitstream(i);
        if bit == previousBit
            currentCount = currentCount + 1;
        else
            maxCount = max([maxCount, currentCount]);
            currentCount = 1;
            previousBit = bit;
        end
    end

    max_rep = max([maxCount, currentCount]);
end

function rle = rle_encode_bitstream(bitstream,bit_len)
    % Initialize variables
    rle = [];
    current = bitstream(1);
    count = 1;
```

```
    % Loop through the bitstream
    for i = 2:length(bitstream)
        if bitstream(i) == current
            % If the current bit is the same as the previous one, increment
count
            count = count + 1;
        else
            % If the current bit is different, add the previous bit and its
count to the output
            rle = [rle dec2bin(count,bit_len) num2str(current)];
            current = bitstream(i);
            count = 1;
        end
    end

    % Add the last bit and its count to the output
    rle = [rle dec2bin(count,bit_len)];
    rle = [rle num2str(current)];
end
```