

# Báo Cáo Xử Lý Ảnh BT1

## Bài 2.2

1. Tạo một ảnh 8-bits X1 có kích cỡ  $200 \times 300$  chỉ gồm các điểm ảnh màu đen. Đưa code MATLAB và ảnh vào báo cáo.

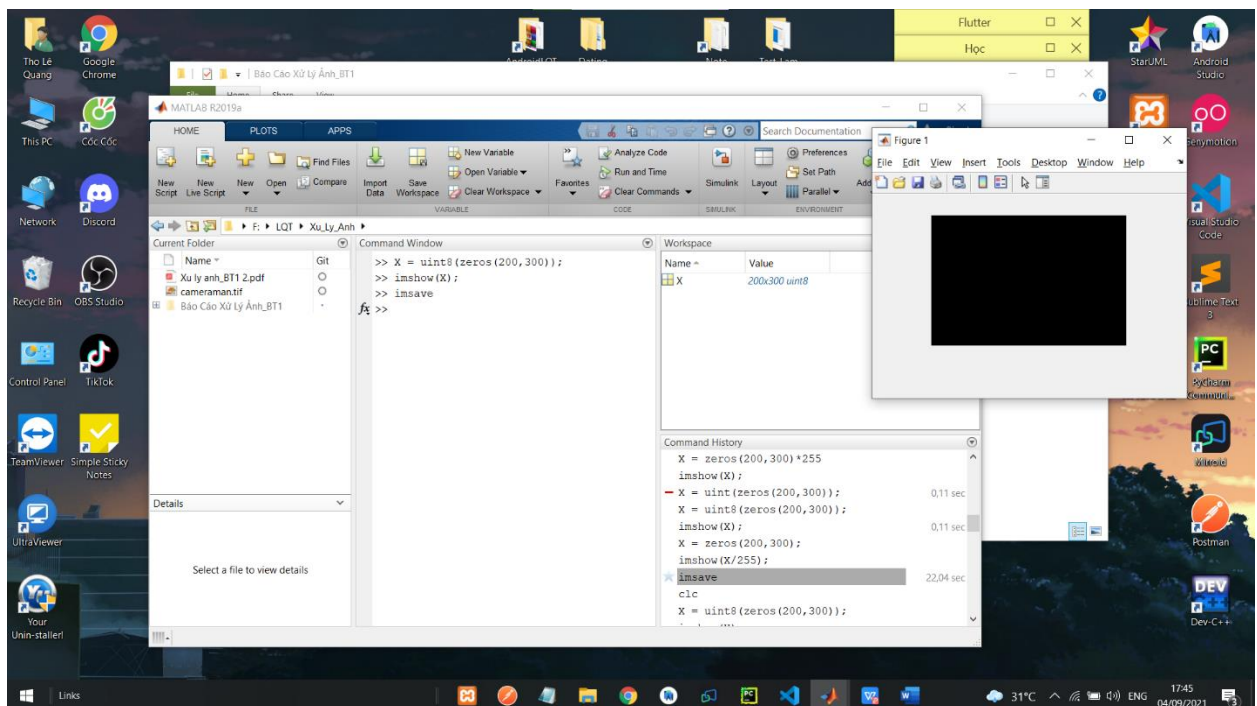
- Code:

```
>> X = uint8(zeros(200,300));
```

```
>> imshow(X);
```

```
>> imsave
```

- Ảnh



Ảnh 2.2.1

2. Tạo một ảnh 8-bits có kích cỡ  $200 \times 300$  chứa toàn các điểm ảnh màu trắng.

Đưa code MATLAB và ảnh vào báo cáo.

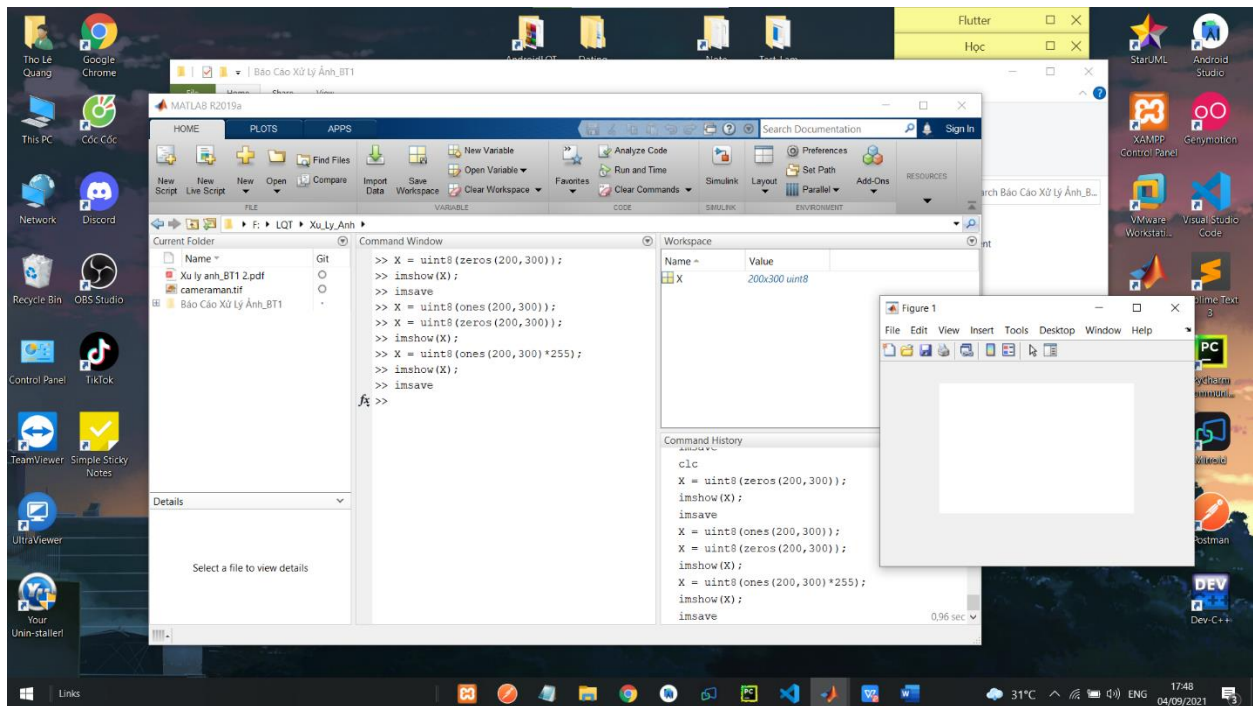
- Code:

```
>> X = uint8(ones(200,300)*255);
```

```
>> imshow(X);
```

```
>> imsave
```

- Ảnh



Ảnh 2.2.2

**3. Tạo một ảnh 8-bits có kích cỡ 200x300 chứa toàn các điểm ảnh màu xám sáng (sinh viên tự chọn giá trị màu). Đưa code MATLAB và ảnh vào báo cáo.**

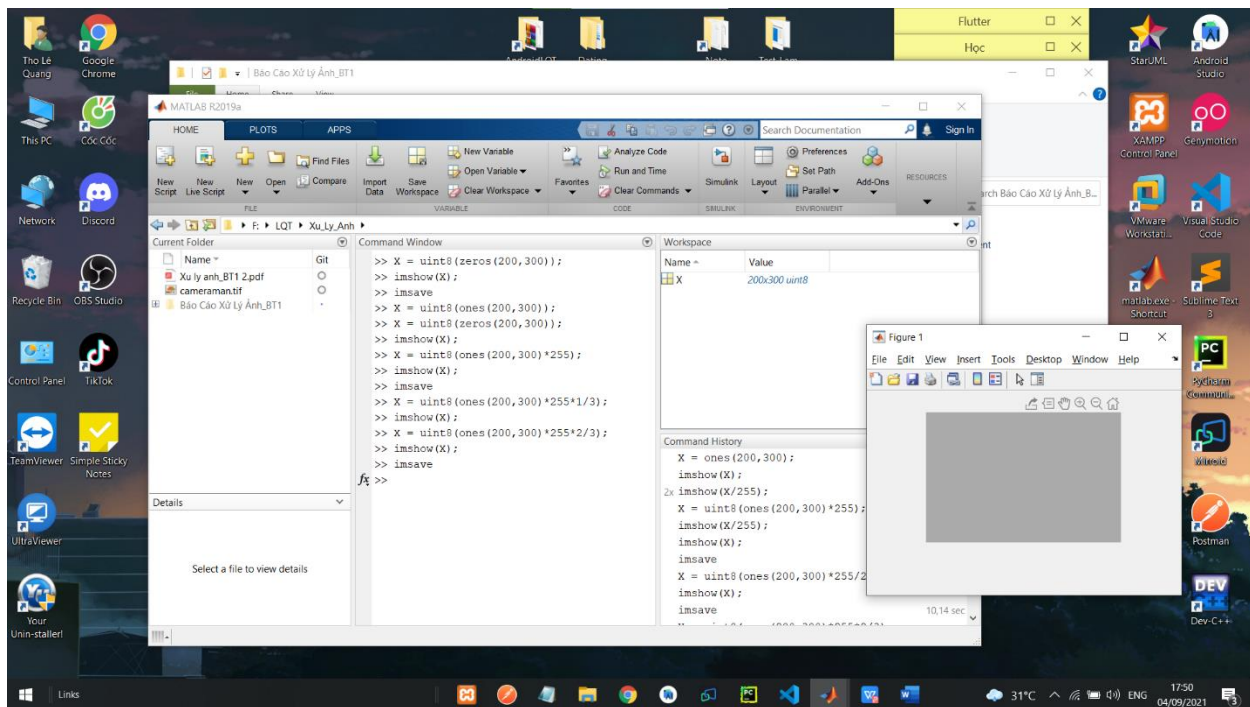
- Code:

```
>> X = uint8(ones(200,300)*255*2/3);
```

```
>> imshow(X);
```

```
>> imsave
```

- Ảnh:



Ảnh 2.2.3

\* do ảnh xám trắng nên thiên về màu trắng hơn

**4. Tạo một ảnh 8-bits có kích cỡ 200x300 chứa toàn các điểm ảnh màu xám đậm (sinh viên tự chọn giá trị màu). Đưa code MATLAB và ảnh vào báo cáo.**

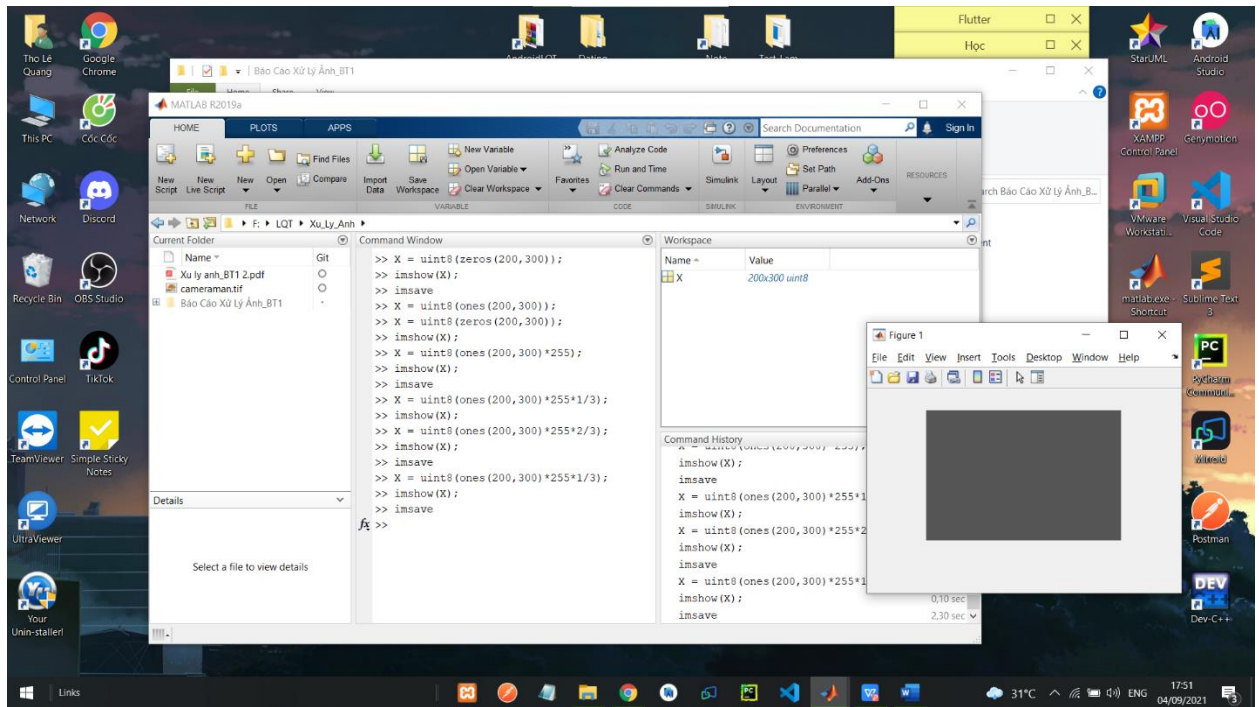
- Code

```
>> X = uint8(ones(200,300)*255*1/3);
```

```
>> imshow(X);
```

```
>> imsave
```

- Ảnh:



Ảnh 2.2.4

\* ảnh xám đậm nên thiên về màu đen hơn

**5. Tạo một ảnh 8-bits có kích cỡ 200x300 có nền màu xám sáng và có một đường màu đen nằm ngang có độ dày 10 pixels (Vị trí của đường màu đen do sinh viên tự chọn). Đưa code MATLAB và ảnh vào báo cáo.**

- Code:

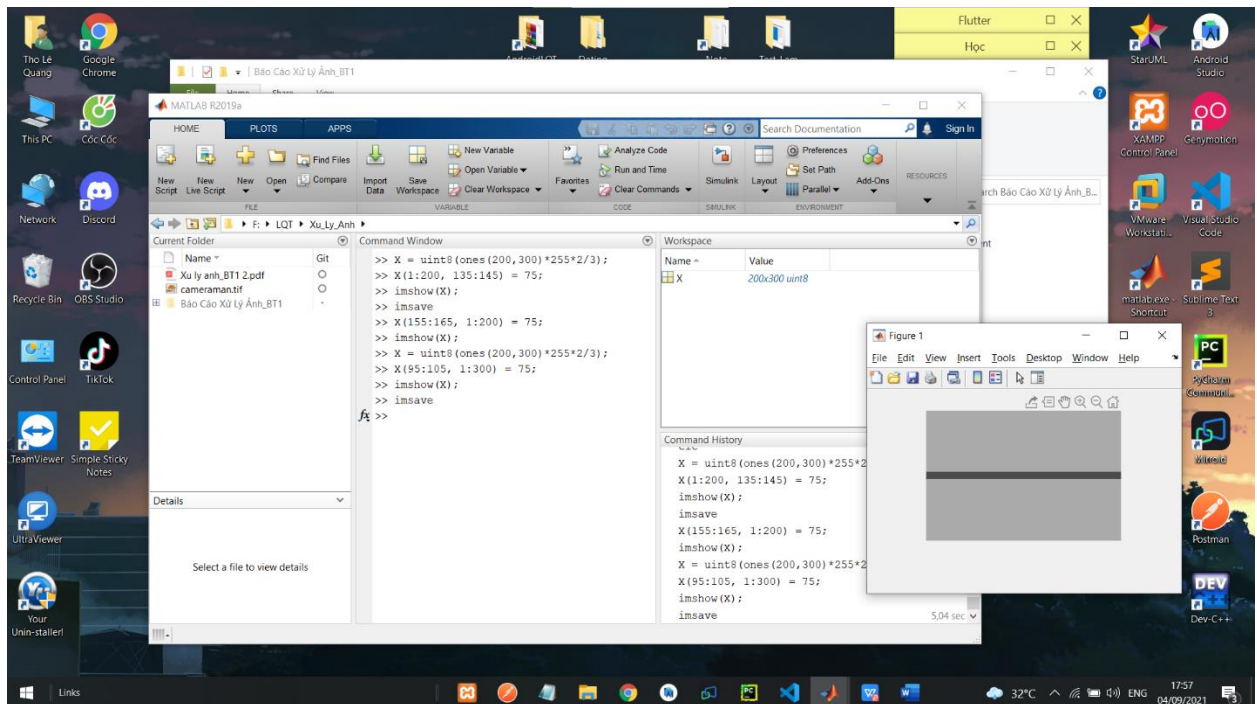
```
>> X = uint8(ones(200,300)*255*2/3);
```

```
>> X(95:205, 1:300) = 75;
```

```
>> imshow(X);
```

```
>> imsave
```

- Ảnh



Ảnh 2.2.5

**6. Tạo một ảnh 8-bits có kích cỡ 200x300 có nền màu xám đậm, có một đường màu đen nằm ngang có độ dày 10 pixels và một đường màu trắng nằm dọc có độ dày 10 pixels. Các điểm nằm trong vùng giao nhau giữa đường đen và trắng thì giữ nguyên màu xám đậm như màu nền. Đưa code MATLAB và ảnh vào báo cáo.**

- Code:

```
>> X = uint8(ones(200,300)*255*1/3);
```

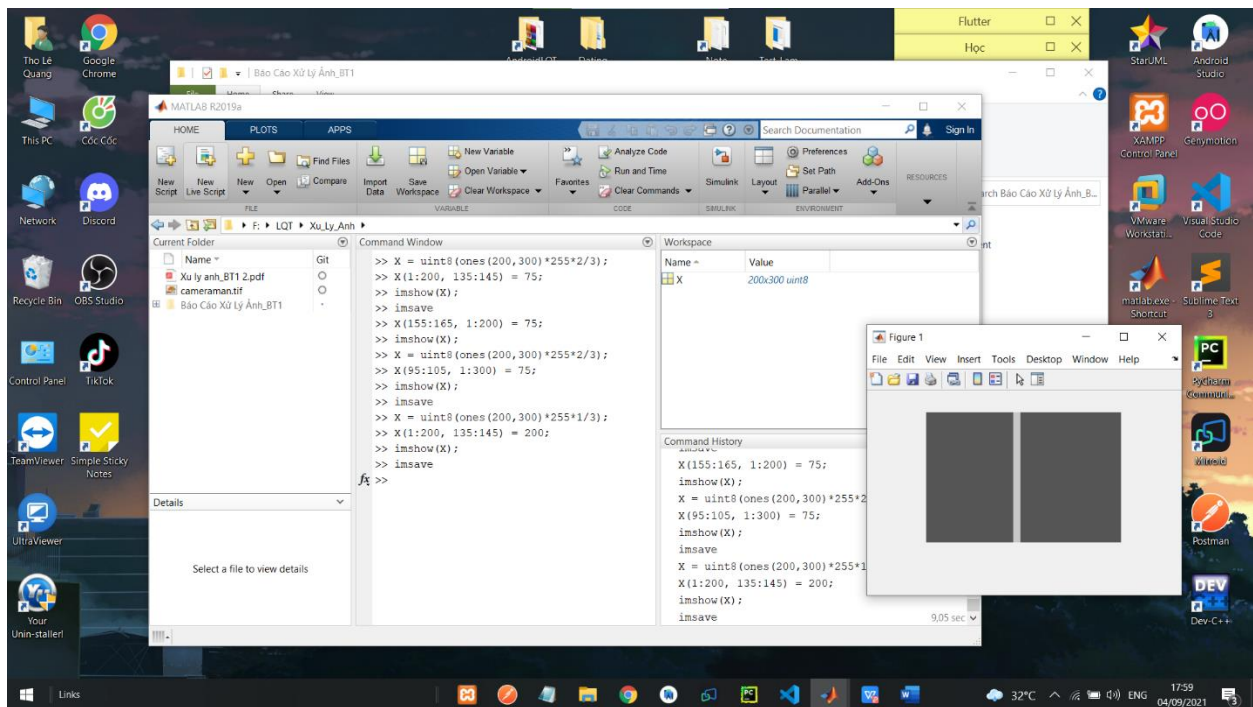
```
>> X(1:200, 135:145) = 200;
```

```
>> imshow(X);
```

```
>> imsave
```

- Ảnh:





Ảnh 2.2.6

**7. Tạo một ảnh 8-bits có kích cỡ 200x300 có nền màu xám sáng và có một ô màu đen nằm trong hình có kích cỡ 30x30. Đặt ô màu đen đó vào chính giữa hình. Đưa code MATLAB và ảnh vào báo cáo.**

- Code:

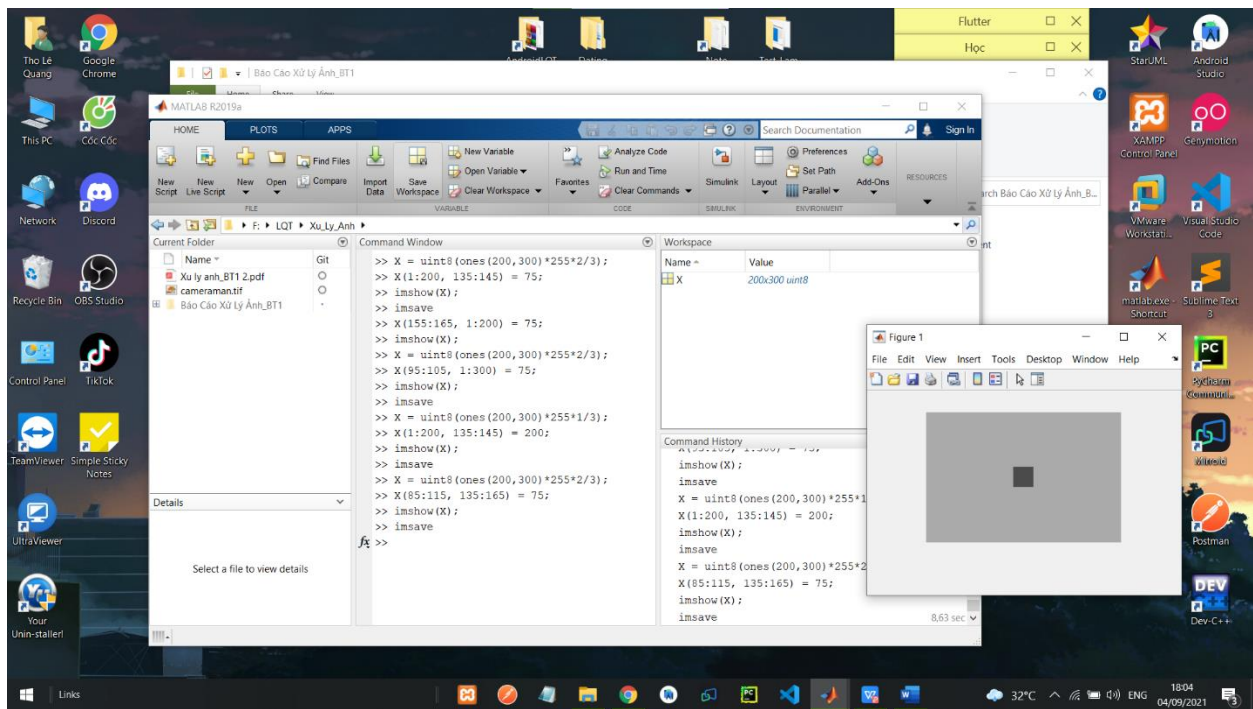
```
>> X = uint8(ones(200,300)*255*2/3);
```

```
>> X(85:115, 135:165) = 75;
```

```
>> imshow(X);
```

```
>> imsave
```

- Ảnh:



Ảnh 2.2.7

## Bài 3.2

### 1. Mở file ảnh cameraman.tiff, lưu vào ma trận X1. Chuẩn hoá ma trận

X1. Đưa code MATLAB vào báo cáo.

Để đọc ảnh trong MATLAB chúng ta dùng các câu lệnh sau:

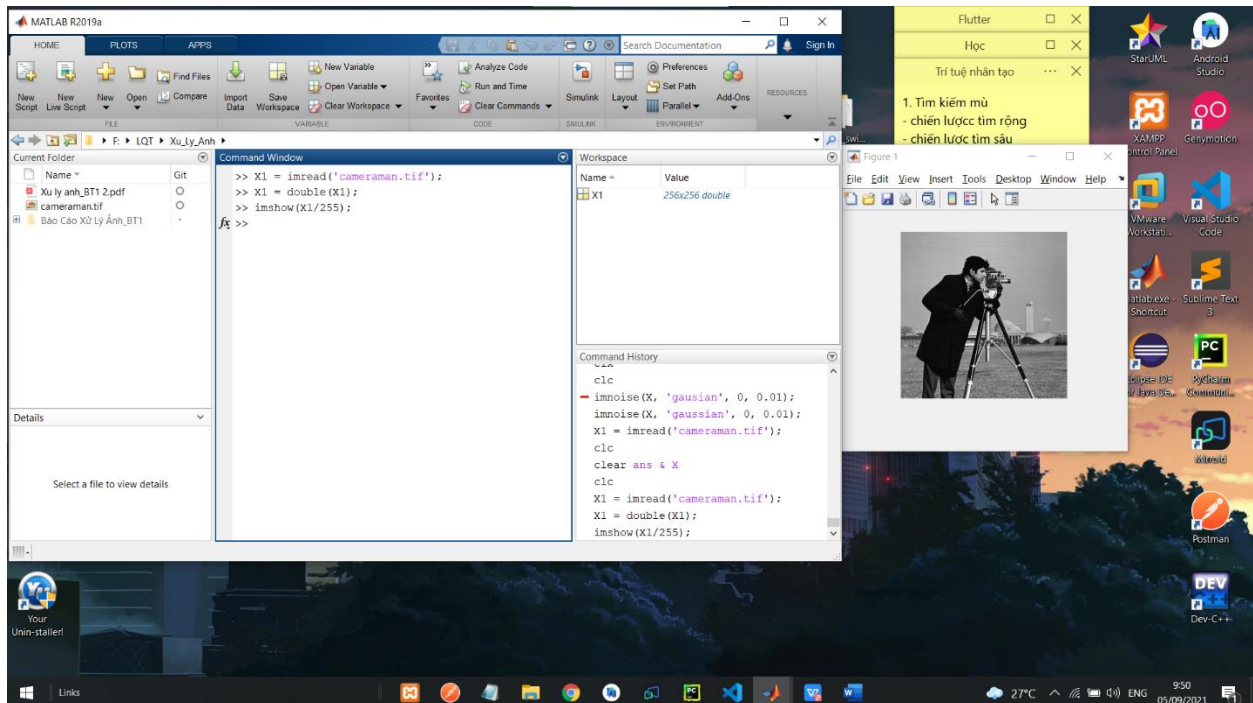
- Code:

```
>> X1 = imread('cameraman.tif');
```

```
>> X1 = double(X1);
```

```
>> imshow(X1/255);
```

- Ảnh:



Ảnh 3.2.1



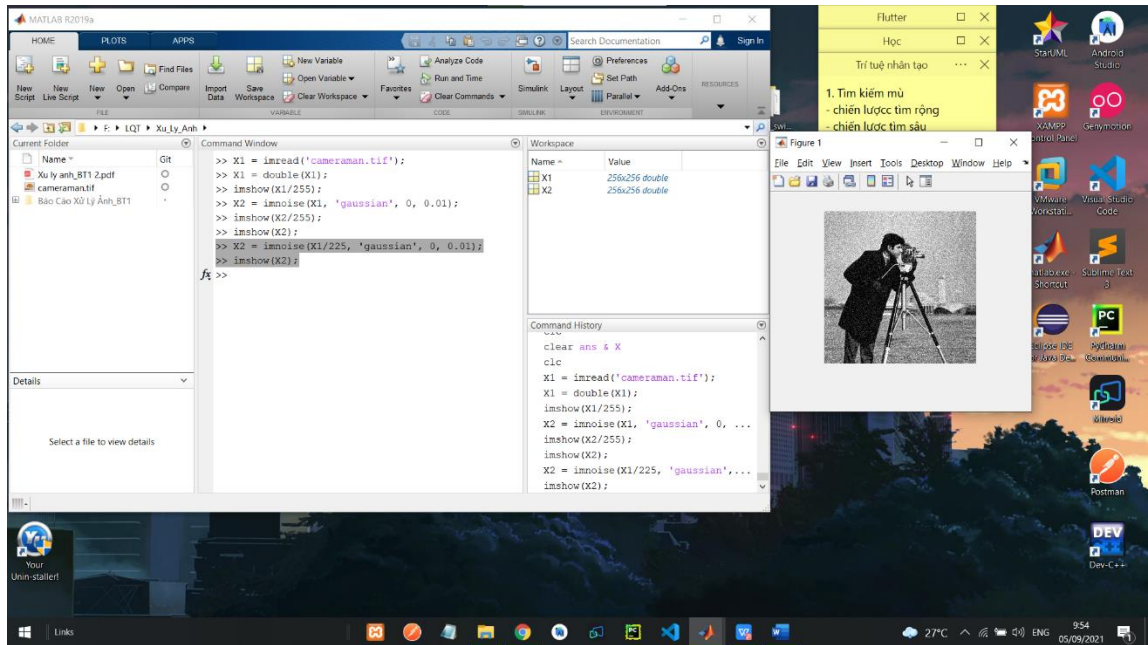
2. Thêm vào X1 một nhiễu Gausse với trung bình 0 và phương sai 0.01 và lưu ảnh mới vào ma trận X2. Đưa code và ảnh X2 vào báo cáo. Thử thay đổi phương sai xem có ảnh hưởng gì đến chất lượng ảnh?

- Code:

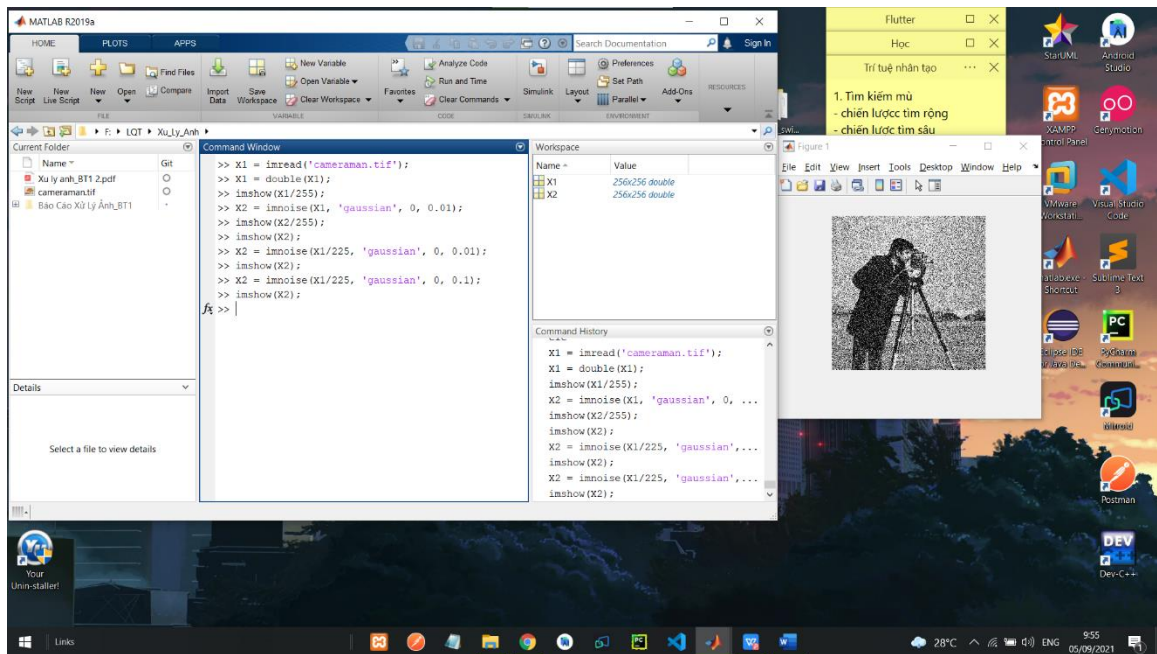
```
>> X2 = imnoise(X1/225, 'gaussian', 0, 0.01);
```

```
>> imshow(X2);
```

- Ảnh:



- Độ lớn của phương sai tỉ lệ thuận với độ noise của ảnh



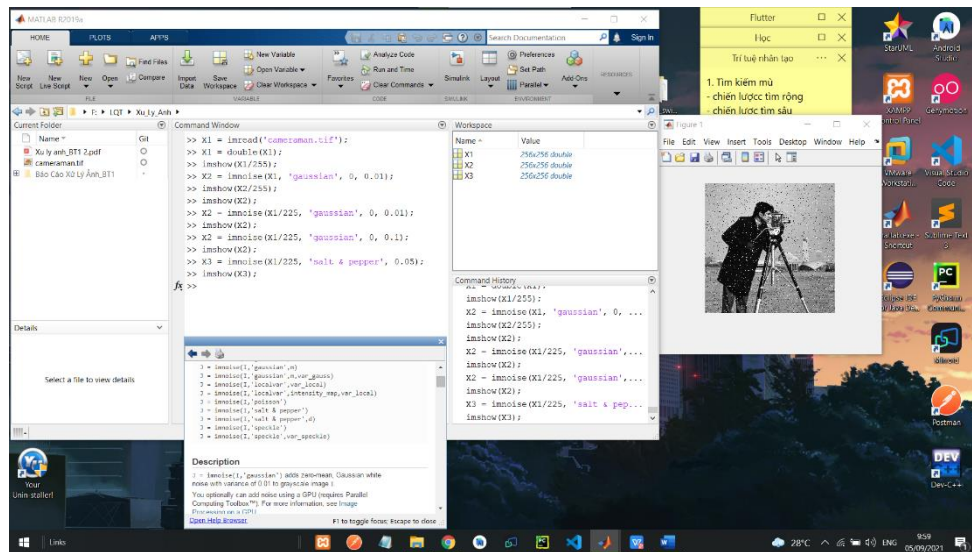
**3. Thêm vào X1 một nhiễu muối tiêu với tỉ lệ  $p = 0.05$  và lưu vào ma trận X3. Đưa code và ảnh X3 vào báo cáo. Thử thay đổi  $p$  xem có ảnh hưởng gì đến chất lượng ảnh.**

- Code:

```
>> X3 = imnoise(X1/225, 'salt & pepper', 0.05);
```

```
>> imshow(X3);
```

- Ảnh:



4. Vẽ trên một đồ thị giá trị các điểm ảnh tại hàng 128 của X1, X2, X3. Đưa code và đồ thị vào báo cáo. Nhận xét về đồ thị thu được.

- Code:

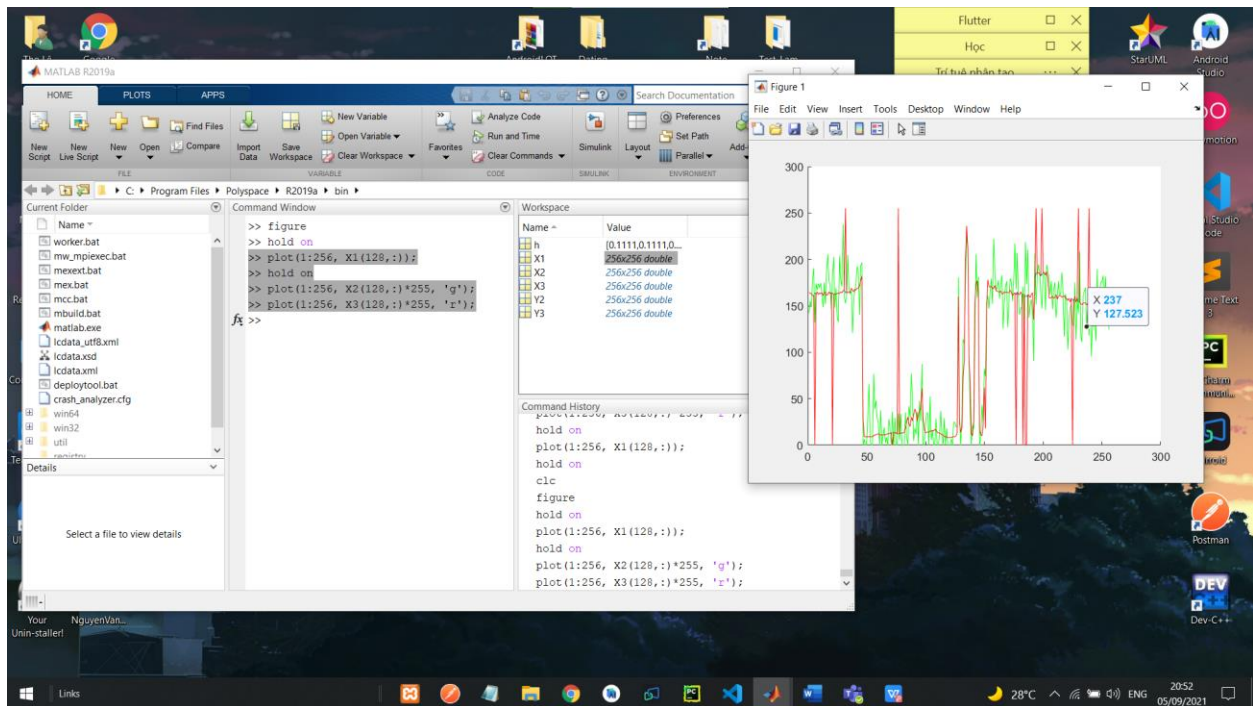
```
>> plot(1:256, X1(128,:));
```

```
>> hold on
```

```
>> plot(1:256, X2(128,:)*255, 'g');
```

```
>> plot(1:256, X3(128,:)*255, 'r');
```

- Ảnh:



Ảnh 3.3.4

\* Đồ thị của X1 và X3 gần như trùng nhau

## Bài 4.2

1. Áp dụng bộ lọc trung bình  $3 \times 3$  với ảnh X2 ở phần trước và lưu vào ma trận Y2. Đưa code và ảnh Y2 vào báo cáo. Nhiễu của ảnh X2 có được lọc hiệu quả không?

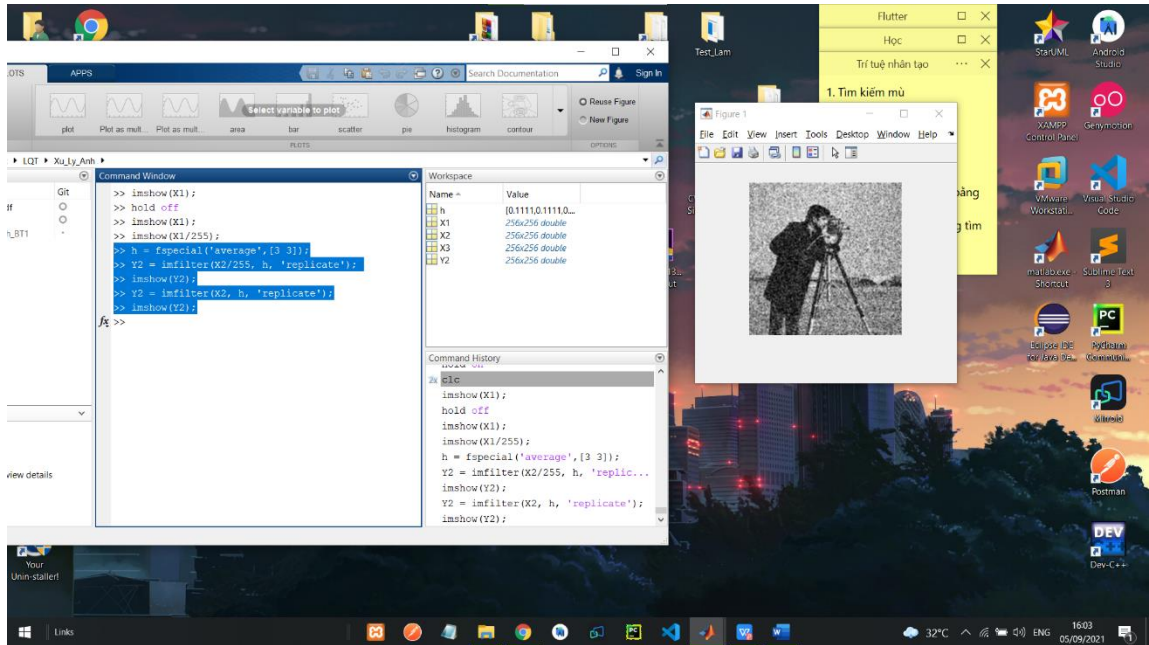
- Code:

```
>> h = fspecial('average',[3 3]);
```

```
>> Y2 = imfilter(X2, h, 'replicate');
```

```
>> imshow(Y2);
```

- Ảnh:



Ảnh 4.2.1

\* Ảnh X2 sau khi qua bộ lọc trung bình thì nhiễu thậm chí không giảm mà còn lan rộng ra các điểm lân cận

**2. Áp dụng bộ lọc trung vị  $3 \times 3$  với ảnh X3 ở phần trước và lưu vào ma trận Y3. Đưa code và ảnh Y3 vào báo cáo. Nhiễu của ảnh X3 có được lọc hiệu quả không**

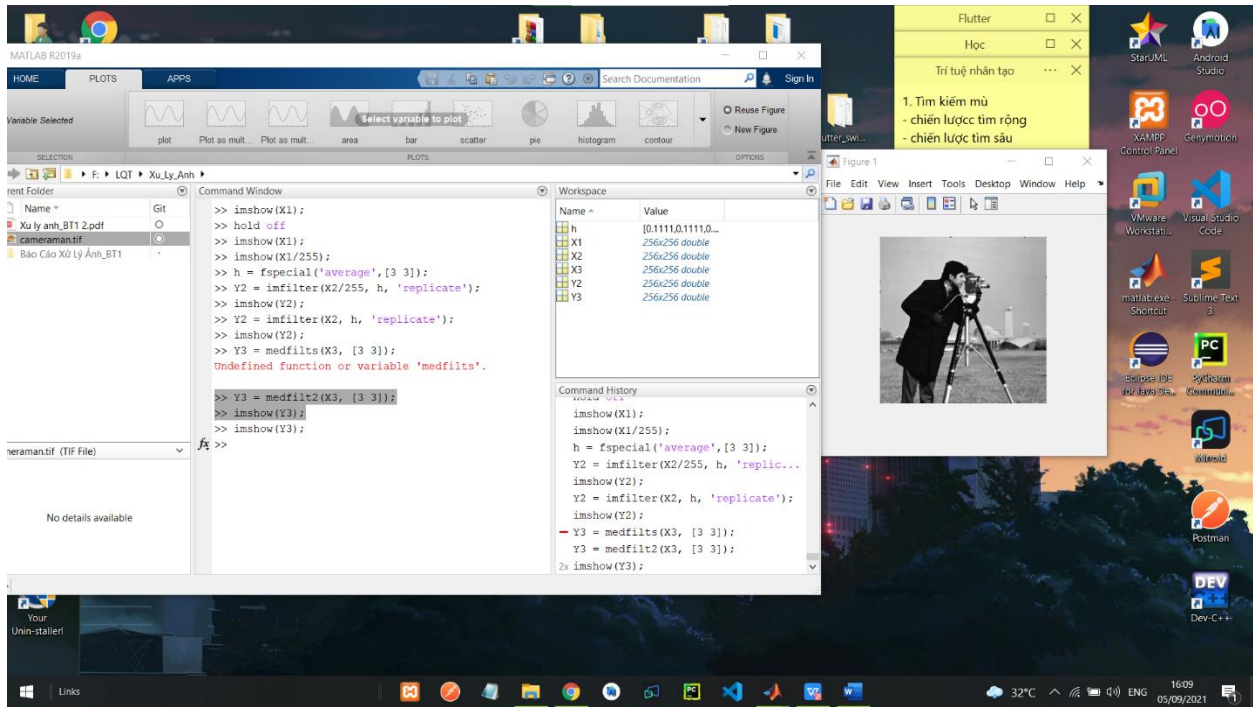
- Code:

```
>> Y3 = medfilt2(X3, [3 3]);
```

```
>> imshow(Y3);
```



- Ảnh:



Ảnh 4.2.2

\* Sau khi X3 qua lọc trung vị thì nhiễu được khử hoàn toàn

**3. Tính PSNR với hai phép khử nhiễu ở câu 2 và 3 (tương đương với việc tính PSNR của Y2, Y3 với X1). Đưa giá trị của PSNR vào báo cáo.**

- Code:

```
>> X1 = imread('cameraman.tif');
>> X2 = imnoise(X1, 'gaussian', 0, 0.01);
>> X3 = imnoise(X1, 'salt & pepper', 0.05);
>> h = fspecial('average', [3 3]);
>> Y2 = imfilter(X2, h, 'replicate');
>> Y3 = medfilt2(X3, [3 3]);
>> PSNR1 = psnr(Y2, X1);
>> PSNR2 = psnr(Y3, X1);
```

- PSNR1 = 24.3506  
- PSNR2 = 26.5186

4. Thử khử nhiễu ảnh X2 và X3 với các bộ lọc sau:

- Lọc trung bình với kích thước bộ lọc lần lượt là:  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ .
- Lọc Gausse với độ lệch chuẩn lần lượt là 2, 1.5, 1, 0.5.
- Lọc trung vị với kích thước bộ lọc lần lượt là:  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ .

Đưa giá trị PSNR của mỗi lần thử vào báo cáo dưới dạng bảng.

Kích thước	Độ lệch chuẩn	Lọc trung bình		Lọc Gaussian		Lọc trung vị	
3x3	2	24.3627	21.3241	23.2838	22.8679	24.1643	26.6604
5x5	1.5	22.6278	22.2893	23.7620	23.2524	22.9719	23.6307
7x7	1	21.4967	21.3241	24.7912	23.8973	22.2437	21.9246
	0.5			23.7588	21.3714		
X2		V		V		V	
X3			V		V		V

Bảng giá trị PSNR qua các phép thử