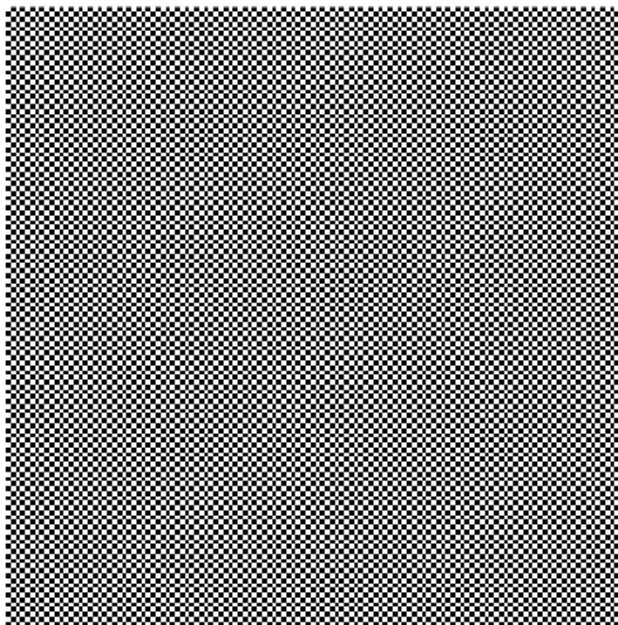


(1)

الف) برای این کار یک ماتریس با درایه های صفر تولید می کنیم. سپس برای سطر های فرد یکی در میان خانه ها را صافید می کنیم. برای سطر های زوج نیز به همین ترتیب عمل کرده تا تصویر شطرنجی حاصل شود.

```
img1 = zeros(128,128,'uint8');  
img1(1:2:end,1:2:end) = 255;  
img1(2:2:end,2:2:end) = 255;
```



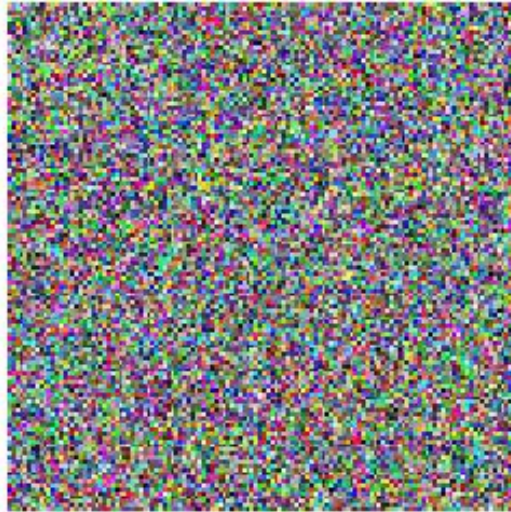
ب) برای این مورد یک آرایه یک بعدی که مقدار آن از ۰ تا ۲۵۵ زیاد می شود تولید کرده و با repmat آن را به یک ماتریس تبدیل می کنیم

```
img2 = repmat(uint8(0:255), [256,1]);
```



ج) با استفاده از randi ماتریسی سه بعدی با ابعاد ۱۲۸ در ۱۲۸ در ۳ تولید می کنیم.

```
img3 = randi([0,255],128,128,3,'uint8');
```



د) ابتدا ماتریسی ۳ بعدی با ابعاد ۲۵۶ در ۲۵۶ در ۲۵۶ تولید کرده و سپس به کمک دستور meshgrid سه ماتریس X,Y,C را تولید می کنیم که حاوی مختصات پیکسل از نظر مکانی و کانال هستند. به کمک این مختصات برای دایره تو خالی زرد به مرکزیت نقطه دلخواه رابطه دایره را نوشته و آن را کوچکتر از عددی که ضخامت لبه دایره را مشخص می کند قرار می دهیم. سپس چون برای داشتن رنگ زرد کانال سوم یعنی آبی باید صفر شود آن را نیز با مختصات مکانی ترکیب کرده تا شکل حاصل شود

```
img4 = ones(256,256,3,'uint8')*255;
[X,Y,C] = meshgrid(1:256,1:256,1:3);
img4((abs(((X-30).^2+(Y-30).^2)-(10^2)) <= 25) & (C == 3)) = 0;
```

برای دایره قرمز توپر نیز به همین ترتیب عمل کرده فقط عبارت تساوی در معادله دایره را به کوچکتر مساوی تبدیل می کنیم و برای رنگ قرمز دو کانال دوم و سوم یعنی سبز و آبی را مساوی صفر قرار می دهیم.

```
img4((((X-90).^2+(Y-90).^2) <= (5^2)) & ((C == 2) | (C == 3))) = 0;
```

برای مربع سیاه نیز ابتدا دو ضلع افقی را به صورت زیر رسم کرده

```
img4(((X<=199) & (X>=180)) & ((Y == 180) | (Y == 199))) = 0;
```

و سپس همین کار را برای دو ضلع عمودی تکرار می کنیم.

```
img4(((Y<=199) & (Y>=180)) & ((X == 180) | (X == 199))) = 0;
```

در این حالت چون رنگ خروجی سیاه است همه کانال ها باید صفر شوند.



ه) ابتدا یک ماتریس خالی تولید کرده و مختصات تک تک نقاط سیاه را در آن وارد می کنیم تا تصویر موجود در صورت سوال تولید شود. سپس برای مقیاس دهی از تابع `repelem` استفاده کرده تا هر پیکسل را ۱۰ بار تکرار کنیم

```
img5 = ones(11,24);  
img5(1:4,3) = 0;  
img5(4,4:5) = 0;  
img5(8:9,[5,10]) = 0;  
img5(7:10,[6,9]) = 0;  
img5([6:7,10:11],7:8) = 0;  
img5(1:10,[13,16]) = 0;  
img5(9:10,14:15) = 0;  
img5(7:10,17:22) = 0;  
img5([3,6,9],24) = 0;  
img5([2,4],[19,21]) = 0;  
img5(3,20) = 0;  
img5 = repelem(img5,10,10);
```



۲) تابع مورد نظر با لغزاندن یک پنجره با ابعاد فیلتر بر روی تصویر پیاده سازی شد. نتایج به صورت زیر است:



الف) فیلتر میانگیر به صورت زیر پیاده سازی شد و هرچه فیلتر بزرگتر شود تصویر محو تر ولی نویز ها کمتر می شوند

```
avg_filter = @(im) mean(im, 'all');
```

ب) فیلتر میانه به صورت زیر پیاده سازی شد و با افزایش اندازه فیلتر علاوه بر محو شدن تصویر نواحی از لحاظ شدت روشنایی یکنواخت تر می شوند.

```
med_filter = @(im) median(im, 'all');
```

ج) فیلتر گوسی ۳ در ۳ به شکل زیر تعریف و پیاده سازی شد.

```
gauss_filter3 = @(im) sum(double(im).*[1,2,1;2,4,2;1,2,1], 'all')/16;
```