

به نام خدا

پارسا محمدپور ۹۸۲۴۳۰۵۰

-۱

الف) در این قسمت از ما خواسته شده است که بدون استفاده از دستور checkerboard یک تصویر شطرنجی بسازیم.

در متلب هر تصویر به صورت یک ماتریس است که بنا بر نوع آن (سفید و سیاه و یا رنگی) نمایش داده می شود.

پس ابتدا با توجه به خواسته صورت سوال و با توجه به اینکه ماتریس سیاه و سفید است یک ماتریس دو بعدی $128 * 128$ میسازیم

```
a = zeros(128, 128);
```

سپس با توجه به اینکه میخواهیم تصویر شطرنجی تولید کنیم و در تصاویر شطرنجی، خانه ها یکی در میان در هر سطر و در هر ستون سفید و سیاه هستند، پس در داخل دو تا حلقه for تودرتو، کل خانه های ماتریس را پیمایش میکنیم.

```
for i = 1 : 128
    for j = 1 : 128
        ...
    end
end
end
```

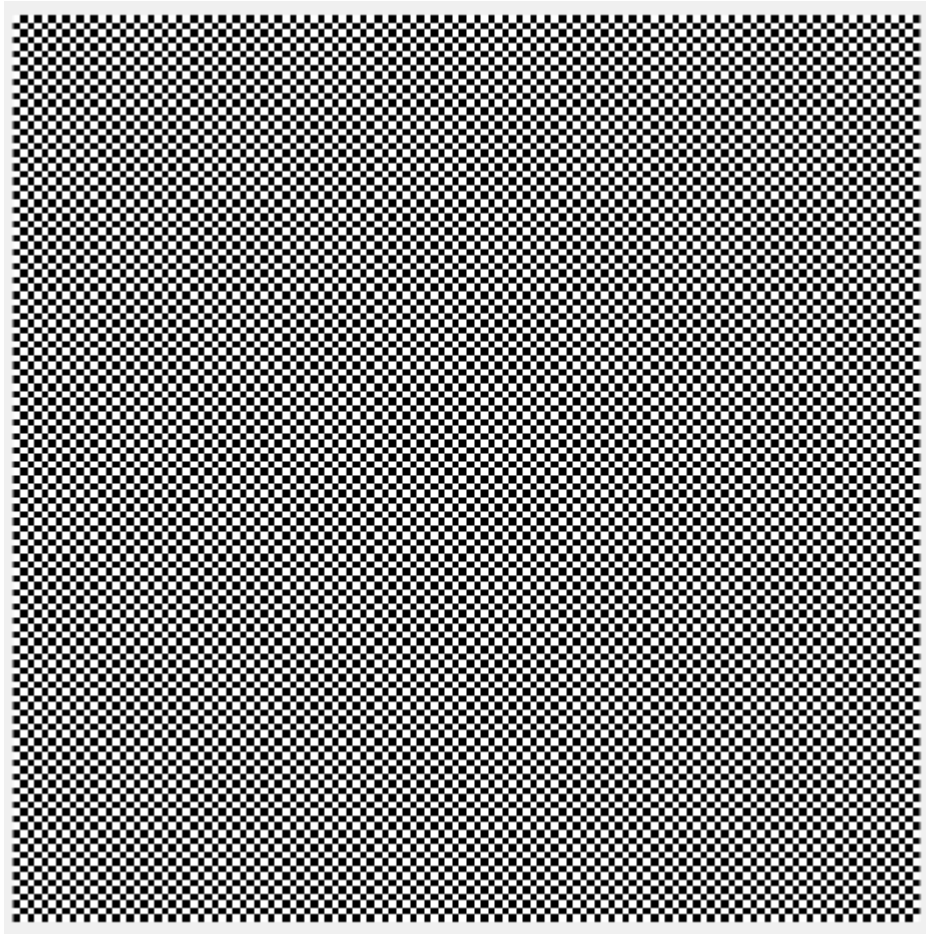
با توجه به اینکه می خواستیم خانه ها یکی در میان سیاه و سفید باشند می توان به این صورت این کار را انجام که اگر خانه ای، مجموع i و j آن که شماره سطر و ستون هر درایه در ماتریس است، زوج بود، رنگ آن خانه را سفید میگذاریم (مقدار عددی این درایه با توجه به اینکه رنگ هایمان را در بازه ۰ تا ۲۵۵ در نظر میگیریم، پس رنگ سفید برابر با ۲۵۵ می باشد و رنگ سیاه هم برابر با صفر می باشد) و اگر هم این مقدار (جمع شماره سطر و ستون آن درایه فرد بود) رنگ آن خانه را سیاه میگذاریم (عدد صفر را در آن درایه میگذاریم)

```
if (mod(i + j, 2) == 0)
    a(i,j) = 255;
else
    a(i,j) = 0;
end
```

(برای حساب کردن باقیمانده عدد a به عدد b از تابع $\text{mode}(a, b)$ استفاده می شود)
سپس برای نمایش و ذخیره عکس هم (مشابه با تمرین قبلی اینگونه عمل میکنیم) و آن بازه موجود در تابع `imshow` در اصل بازه رنگ های مورد نظر است که ما میخواهیم بین صفر تا ۲۵۵ باشند

```
figure();
imshow(a, [0,255]);
imwrite(a, 'Q1_Part1_Image.png');
```

عکس خروجی قطعه کد بالا به صورت زیر می باشد:



(ب)

در این قسمت از ما خواسته شده است که تصویری را نشان دهیم که در آن از شدت رنگ ها با حرکت به سمت راست تصاویر زیاد وند و شدت رنگ هر ستون یکسان باشد پس ما درایه های یک ماتریس 256×256 را به گونه ای پر میکنم که شدت رنگ هر ستون متناسب با عدد آن ستون باشد و به ازای تمامی سطر های آن ستون این مقدار ثابت باشد. (ساخت ماتریس)

```
a = zeros(256, 256);
```

برای این کار میشد از دو تا فور تو در تو استفاده کرد اما یک سینتکس راحت تر برای انجام این کار موجود است ، که در این سینتکس به یثوان با گذاشتن ":" گفت که این کار را به ازای تمامی مقادیر این بعد از ماتریس یک کار مشخص را انجام بده .

```
for j = 1 : 256  
    a(:,j) = j - 1;  
end
```

معنی این حلقه و این عملیات در کل بدین شکل است که برای تمامی ستون های از یک تا ۲۵۶ ، در هر تمامی سطر های اسن ستون ، عدد 1 - z را قرار بده (به این دلیل منفی یک میکنیم که ستون های ماتریس از ۱ شروع می شوند ولی بازه رنگ های ما از صفر!).
(همانطور که در کد هم کامنت شده است ، اگر میخواستیم رنگ های سمت راست ، سفید تر باشند ، باید به جای 1 - z مقدار z - 256 را قرار میدادیم)

حال از تیکه کد زیر برای نمایش و ذخیره سازی این عکس خروجی استفاده میکنیم:

```
figure();  
imshow(a, [0,255]);  
imwrite(a, 'Q1_Part2_Image.png');
```

(آن بازه موجود در تابع imshow در اصل بازه رنگ های مورد نظر است که ما میخواهیم بین صفر تا ۲۵۵ باشند)



(ج)

(یک تصویر RGB که در اصل همان تصویر رنگی میباشد ، یک ماتریس ۳ بعدی است که دو بعد اول ابعاد تصویر هستند و بعد سوم ، یک ۳ تایی است که هر کدام از اینها معادل یک شدت رنگ در حالت RGB برای هر کدام از رنگ های قرمز و آبی و سبز میباشد) برای ساختن یک تصویر RGB هم ابتدا می آییم و مقدار دهی اولیه را انجام میدهیم ب صورت زیر:

```
a = zeros(128,128,3,'uint8');
```

حال با توجه خواسته صورت سوال ، تمامی درایه های این ماتریس باید پیمایش شوند:

```
for i = 1 : 128
    for j = 1 : 128
        for k = 1 : 3
            ...
        end
    end
end
```

در هر درایه عددی تصادفی از باز صفر تا ۲۵۵ قرار گیرد.
برای تولید عدد تصادفی از تابع randi() استفاده میکنیم و اگر آرگومان آن را عددی مثل i بدهیم خروجی حاصل یک عدد تصادفی در بازه 1 تا i خواهد بود و از آنجایی که بایه اعداد ما ، از صفر تا ۲۵۵ است پس تابع randi را با آرگومان ۲۵۶ صدا میکنیم و از مقدار حاصل یکی کم میکنیم :

```
a(i,j,k)= randi(256) - 1;
```



سپس همانند قسمت های قبل با استفاده از دستورا زیر خروجی را چاپ و ذخیره میکنیم :

```
figure();  
imshow(a);  
imwrite(a, 'Q1_Part3_Image.png');
```

(د)

در این قسمت از ما خواسته شده است که یک تصویر RGB تولید کنیم که در آن اشکال خواسته شده باشند ، برای این کار ابتدا یک ماتریس ۳ بعدی تولید میکنیم(و سپس مقدار تمام درایه های آن را ۲۵۵ میگذاریم ، با اینکار تصویر سفیدی تولید میشود) :

```
a = zeros(256,256,3,'uint8');  
for i = 1 : 256  
    for j = 1 : 256  
        for k = 1 : 3  
            a(i,j,k) = 255;  
        end  
    end  
end
```

حال ابتدا سراغ مربع سیاه و توخالی خواسته شده میرویم و آن را میکشیم.
برای اینکار ه صورت دلخواه ، گوشه راست بالا مربع را در نقطه (با در نظر گرفتن اینکه گوشه بالا تصویر نقطه (۱و۱) است) در نقطه (۱و۱۲۰) قرار میدهیم ، سپس برای رسم این مربع کافی است در طی یک حلقه ۴ ضلع آن را سیاه کنیم و برای این کار میدانیم که فاصله اضلاع عمودی از یکدیگر ۲۰ میباشد و فاصله اضلاع افقی نیز از یکدیگر ۲۰ میباشد ، پس :

```
for i = 1:20  
    for k = 1:3  
        a(120+i,1,k) = 0;  
        a(120,i,k) = 0;  
        a(120+i,20,k) = 0;  
        a(140,i,k) = 0;  
    end  
end
```

در خط اول ضلع سمت چپی مربع و در خط دوم ، ضلع بالایی ، در خط سوم ضلع سمت راستی و در خط آخر هم ضلع پایین بمقدار دهی میشود.

سپس باید دایره تو خالی زرد را رسم کنیم ، برای این کار از sin و cos کمک میگیریم. در این کار ابتدا به صورت دلخواه نقطه مرکز دایره را نقطه (۷۰و۷۰) در نظر میگیریم و بنا بر خواسته صورت سوال ، شعاع این دایره نیز ۱۰ میباشد:

```
r = 10;  
ox = 70;  
oy = 70;
```

حال با توجه به زوایای دایره ، یک متغیر $teta$ تعریف میکنیم که مقدار آن در یک حلقه فور از صفر تا ۳۶۰ افزایش میابد و در هر بار اجرای حلقه میاییم و با توجه به زاویه ای که در آن هستیم حالت بندی میکنیم که بین صفر تا ۹۰ هستیم یا بین ۹۰ تا ۱۸۰ یا بین ۱۸۰ تا ۲۷۰ ی بین ۲۷۰ تا ۳۶۰ ، floor و یا ceil سینوس و کسینوس زاویه را (که محاسبه کردیم) محاسبه میکنیم و سپس درایه ای از ماتریس (سطر و ستون ماتریس را) را با جمع کردن مقدار بدست آمده با مختصات مرکز دایره ، (هر سه خانه مربوط به رنگ را) به شکلی مقدار دهی میکنیم که رنگ زرد در سطر و ستون مورد نظر قرار گیرد (خانه اول و دوم بعد سوم را ۲۵۵ میگذاریم و خانه سوم بعد سوم را هم صفر میگذاریم تا رنگ زرد تولید شود) اما ه دلیل اینکه سینوس و کسینوس دو زاویه که با هم یک درجه فقط اختلاف دارن خیلی زیاد نیست ، در پایان هر حلقه $teta$ را بعلاوه یک میکنیم تا با در نظر گرفتن وقتی که به خاطر خود حلقه فور ، بعلاوه یک میشود ، باعث شود در مرحله بدی در اصل ، بعلاوه دو شده باشد

```
for teta = 0 : 360
    y = cos(teta) .* r + oy;
    x = sin(teta) .* r + ox;
    if teta >= 270
        a(floor(x),floor(y),1) = 255;
        a(floor(x),floor(y),2) = 255;
        a(floor(x),floor(y),3) = 0;
    end
    if teta>=180 && teta <= 270
        a(floor(x),ceil(y),1) = 255;
        a(floor(x),ceil(y),2) = 255;
        a(floor(x),ceil(y),3) = 0;
    end
    if teta>=90 && teta <= 180
        a(ceil(x),ceil(y),1) = 255;
        a(ceil(x),ceil(y),2) = 255;
        a(ceil(x),ceil(y),3) = 0;
    end
    if teta <= 90
        a(ceil(x),floor(y),1) = 255;
        a(ceil(x),floor(y),2) = 255;
        a(ceil(x),floor(y),3) = 0;
    end
    teta = teta + 1;
end
```

حال باید دایره تو پر قرمز را تولید کنیم ، برای این کار مابه قسمت قبل عمل میکنیم و ابتدا یک دایره تو خالی قرمز رسم میکنیم با استفاده از کد زیر (با توجه به صورت سوال ، شعاع دایره ۵ است و به صورت دلخواه مختصات مرکز دایره را (۱۴۰ و ۱۴۰) در نظر میگیریم) :

```
for teta = 0 : 360
    y = cos(teta) .* r + oy;
    x = sin(teta) .* r + ox;
    if teta >= 270
        a(floor(x),floor(y),1) = 255;
        a(floor(x),floor(y),2) = 0;
        a(floor(x),floor(y),3) = 0;
    end
    if teta>=180 && teta <= 270
        a(floor(x),ceil(y),1) = 255;
        a(floor(x),ceil(y),2) = 0;
        a(floor(x),ceil(y),3) = 0;
    end
    if teta>=90 && teta <= 180
        a(ceil(x),ceil(y),1) = 255;
        a(ceil(x),ceil(y),2) = 0;
        a(ceil(x),ceil(y),3) = 0;
    end
    if teta <= 90
        a(ceil(x),floor(y),1) = 255;
        a(ceil(x),floor(y),2) = 0;
        a(ceil(x),floor(y),3) = 0;
    end
end
end
```

(برای رنگ قرمز ، فقط اولین درایه بعد سوم را ۲۵۵ میگذاریم و بقیه را مان صفر میگذاریم تا رنگ قرمز تولید شود)

سپس با استفاده از یک حلقه میاییم و هر سری با توجه به اینکه در هندسه میدانیم معادله دایه به صورت

$$(x - x_0)^2 + (y - y_0)^2 = R^2$$

معادله یک دایره است و اگر آن را به صورت زیر تغییر دهیم ، شکا حاصل یک دایره توپر خواهد بود:

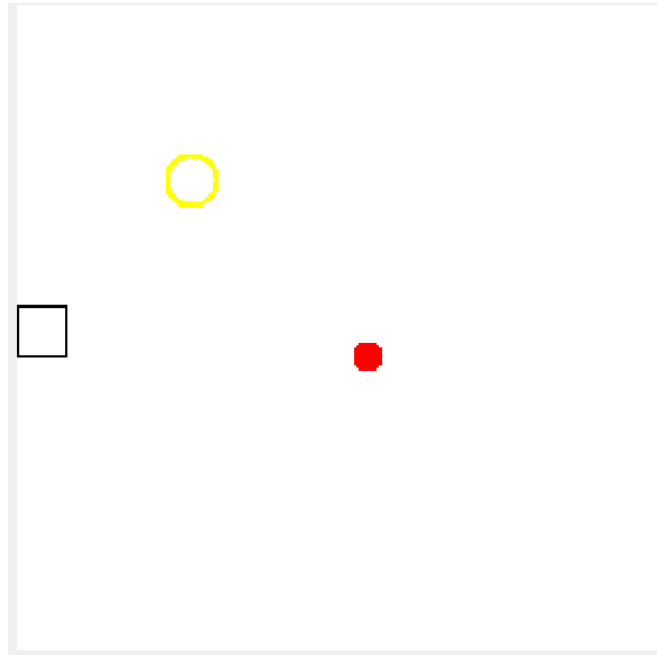
$$(x - x_0)^2 + (y - y_0)^2 \leq R^2$$

حال با توجه به اینکه طول مربعی که دایره ممکن است در قسمت هایی از آن ، توپر باشد ، ۵ است ، پس در دو تا حلقه فور تودرتو که از منفی ۵ شروع میشوند و تا مثبت ۵ میروند ، میاییم و حساب میکنیم که آیا این فاصله x و y با نقطه صفر و صفر (که در آخر برای رنگ کردن ، اینها را بعلاوه مختصات مرکز دایره میکنیم) آیا کوچیکتر مساوی شعاع به توان دو است ا نه ، اگر بود آن را رنگی میکنیم وگرنه میگذاریم سفید بماند.

```

for x=-5:5
    for y=-5:5
        if (x.*x + y.*y <= r.*r)
            a(ox + x, oy + y, 1) = 255;
            a(ox + x, oy + y, 2) = 0;
            a(ox + x, oy + y, 3) = 0;
        end
    end
end
end

```



برای نمایش و ذخیره هم از از کد زیر استفاده میکنیم :

```

figure();
imshow(a);
imwrite(a, 'Q1_Part4_Image.png');

```

(۵)

ابتدا یک متریس را مقدار دهی اولیه میکنیم و با توجه به اینکه از ما خواسته شده که تصویر را با مقیاس ۱۰ برابر بزرگ کنیم ، ابتدا تصویر را دستی مقدار دهی میکنیم :

```

a = zeros(11,24);
for i = 1 : 11
    for j = 1 : 24
        a(i,j) = 255;
    end
end

for i = 1:4
    a(i,3) = 0;
end

```



```

end
for i=3:5
    a(4,i) = 0;
end
for i = 7:8
    a(6,i) = 0;
    a(11,i) = 0;
end
for i=6:9
    a(7,i) = 0;
    a(10,i) = 0;
end
for i =5: 6
    a(8,i) = 0;
    a(8,i+4) = 0;
    a(9,i) = 0;
    a(9,i+4) = 0;
end
for i=1:10
    a(i,13) = 0;
    a(i,16) = 0;
end
for i=13:22
    a(7,i) =0;
    a(8,i) =0;
    a(9,i) = 0;
    a(10,i) = 0;
end
for i=7:8
    a(i,14) = 255;
    a(i,15) = 255;
end
for i =1:3
    a(3.*i,24) = 0;
end
for i=2:4
    for j =19:21
        if mod(i+j,2) == 1
            a(i,j) = 0;
        end
    end
end
end

```

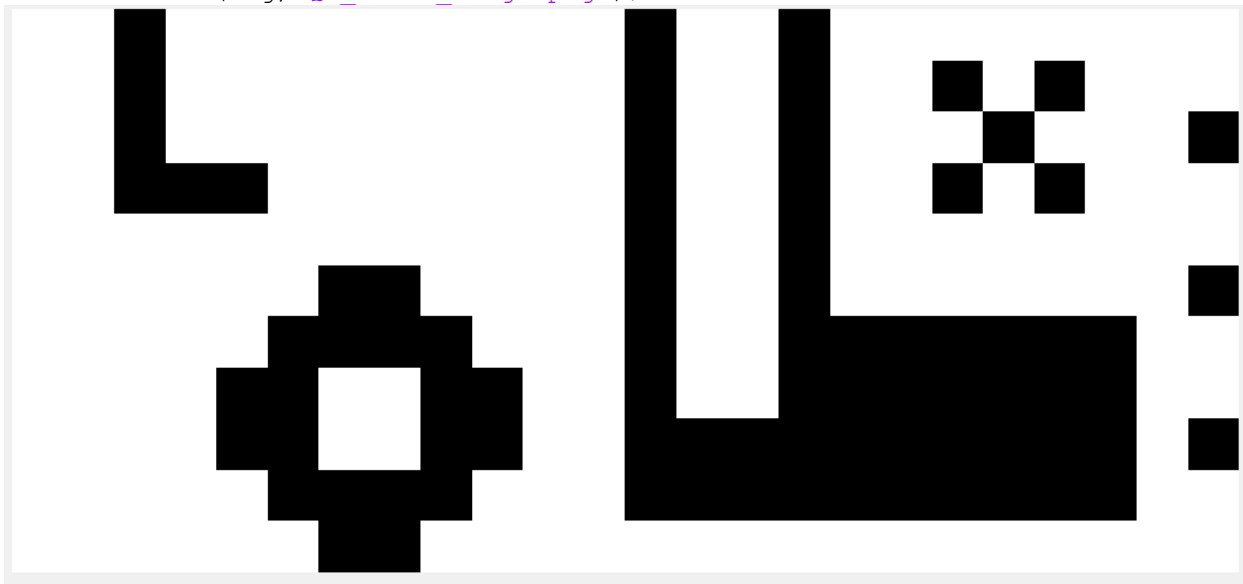
برای اینکه تصویر را ۱۰ برابر بزرگ کنیم باید به ازای هر درایه در ماتریس اولیه ، یک مربع 10×10 را به همان رنگ درآوریم.

سپس یک ماتریس میسازیم (مقدار دهی) میکنیم که تعداد سطر ها و تعداد ستون های آن هرکدام به ترتیبی ۱۰ برابر تعداد سطر ه و تعداد ستون های ماتریس اولیه است ، سپس در ۴ تا حلقه تودرتو ، دو تا برای پیمایش درایه های ماتریس اولیه مان که بازه هایشان هم با توجه به همان تعداد سطر ها و ستون های ماتریس اولیه تعیین میشود ، و دو تا ماتریس دیگر برای رنگ کردن مربع 10×10 متناظر با آن.

```
big = zeros(110,240);
for i=1:11
    for j =1:24
        for k =1:10
            for l=1:10
                big(10.*(i-1) + k,10.*(j-1)+l) = a(i,j);
            end
        end
    end
end
end
```

سپس با استفاده از کد زیر ، عکس را نمایش میدهیم و ذخیره میکنیم.

```
figure();
imshow(big,[0,255]);
imwrite(big,'Q1_Part5_Image.png');
```



(با استفاده از سرچ کردن ، اطلاعات بدست آمده است)

-۲

برای این سوال ابتدا یک تابع `imconvolution` میسازیم و سپس از آن استفاده میکنیم (که شباهت و در اصل استفاده شده از کد تیم حل تمرین است)
 یک سری تغییرات در این کد اعمال شده است ، مثلاً متغیر `stride` چون همواره یک بود ، آن را حذف کردیم و از خود عدد یک استفاده کردیم (و یا بعضاً جایی از کد که همه درایه های ماتریس تقسیم بر یک میشدن را هم حذف کردیم)
 استرینگ `status` رو اضافه کردیم که مشخص میکند آیا این حالت ، فیلتر میانه گیری است یا خیر.

به جای آدرس عکس ، خود عکس رو به این تابع میدهیم.
 خروجی همچنان یک ماتریس است که در هر دو حالت فیلتر میانه بودن یا نبودن ، رنگی است ، پس به صورت زیر مقدار دهی میشود.

```
out = uint8(zeros(0,0));
```

چون تصویر ما اینجا مستطیلی است پس از به جای استفاده از تابع `length` از تابع `size` استفاده میکنیم که مختصات را در قالب یک ماتریس یک بعدی برمیگرداند و در هنگام پیمایش حلقه ها ، به جای یکسان در نظر گرفتن آنها ، از درایه هایاین ماتریس استفاده میشود(درایه خونه اول ، تعداد سطر و درایه دوم ، تعداد ستون هاست) پس در حلقه های فور آنها را جایگزین `n` موجود در کد کردیم.

سپس با استفاده از سرچ کردن ، توانستیم تابعی که فیلتر های مناسب را برای ما میسازد با توجه به نوع فیلتری که میخواهیم انجام دهیم ، که در این تابع توانایی های تولید فیلتر های گوسین و فیلتر میانگین موجود است.

ولی توانایی تولید فیلتر میانه را ندارد .(یک راه استفاده از تابع آماده `metfield3` بود کهالته فقط عکس را دریافت میکرد و فیلتر های متفاوت را نداشت ، مثلا نمیتوانستیم بگوییم که فیلتر `۳*۳` را اعمال کن یا مثلا `۵*۵` را اعمال کن) پس یک متغیر استرینگ `status` به عنوان ورودی تابع در نظر گرفتیم که اگر این متغیر `Median` باشد (که با استفاده از تابع

`strcmp(status,'Median')` مشخص میشود که فیلتر میانه است یا خیر ، روش فیلتر میانه گرفتن را پیاده سازی کردیم) که هرسری تهمانند کد موجود می آید و یک تیکه از ماتریس را میگیرد (تیکه ای به طول فیلر را) و سپس با استفاده از تابع `median` و آرگومان `'all'` میانه کا اعداد موجود در این درماتریس را به ما میدهد و سپس ما این عدد را با عدد فعلی موجود در این درایه جایگزین میکنیم.

(ماتریس `window` در اصل یک زیر ماتریس حاوی اندیس های `i` تا `i + f - 1` و حاوی ستون های `j` تا `j + f - 1` است).

هر سری این عمل میانه گرفتن یا میانگین گیری برای تمام درایه های ماتریس انجام میشود

```
function out = imconvolution(image, filter, status)
n = size(image);
f = length(filter);
if strcmp(status, 'Median')
    f_power = f .* f;
    a = ones(f_power);
    out = uint8(zeros(0,0));
    for i = 1 : n(1) - f + 1
        for j = 1 : n(2) - f + 1
            for channel = 1 : 3
                window = image(i : i + f - 1, j : j + f - 1, channel);
                out(i, j, channel) = median(window, 'all');
            end
        end
    end
else
    p = floor((f - 1) / 2);
    o = floor(n + 2.*p - f + 1);
    image = padarray(image, [p,p], 0, 'both');
    out = uint8(zeros(0,0));
    for channel = 1 : 3
        for i = 1 : n(1) - f + 1
            for j = 1 : n(2) - f + 1
                window = image(i : i + f - 1, j : j + f - 1, channel);
```

```

        out(i, j, channel) = uint8(sum(sum(double(window) .*
double(filter))));
    end
end
end
end
end

```

(الف)

در این قسمت از ما خواسته شده که فیلتر های میانگین 3×3 و 5×5 و 7×7 را اعمال کنیم . ابتدا یک عکس را به دلخواه ، میخوانیم و مطابق با آرگومان های تابع می دهیم ، از جایی که این وضعیت میانگین گیری است پس به جای status به آن 'average' را پاس می دهیم و برای ساخت فیلتر هم از تابع fspecial('average', filter Length) استفاده میکنیم که طول فیلتر یکبار ۳ و یک بار ۵ و یک بار ۷ می باشد.

و سپس پس از گرفتن خروجی تابع imconvolution ، عکس را ذخیره میکنیم و نشن می دهیم.

```

image = imread('image.png');
out = imconvolution(image, fspecial('average',3), 'average');
figure();
imshow(out);
imwrite(out, 'Q2_Part1_Filter3_Image.png');
out = imconvolution(image, fspecial('average',5), 'average');
figure();
imshow(out);
imwrite(out, 'Q2_Part1_Filter5_Image.png');
out = imconvolution(image, fspecial('average',7), 'average');
figure();
imshow(out);
imwrite(out, 'Q2_Part1_Filter7_Image.png');
imshow(out);

```

خروجی این تابع به ترتیب :





(ب)

از ما خواسته شده تا فیلتر های میانه خواسته شده را اعمال کنیم . با توجه به توضیحات ارائه شده ، status را در حالت 'Median' قرار میدهیم و سپس با توجه به اعداد فیلتر ها در هر سری (چون با خود ماتریس فیلتر در این بخش کاری نداریم ، یک ماتریس هم اندازه با سائز فیلتر میسازیم با استفاده از تابع ones و سپس به جای ماتریس فیلتر این ماتریس را به تابع پاس میدهیم) نیازی به این ماتریس در فیلتر میتنه نیست چون در فیلتر میانه ، بدون استفاده از فیلتر و مستقیم با اتسفاذه از تابع آماده median و آرگومان 'all' میانه بدست میاید)).

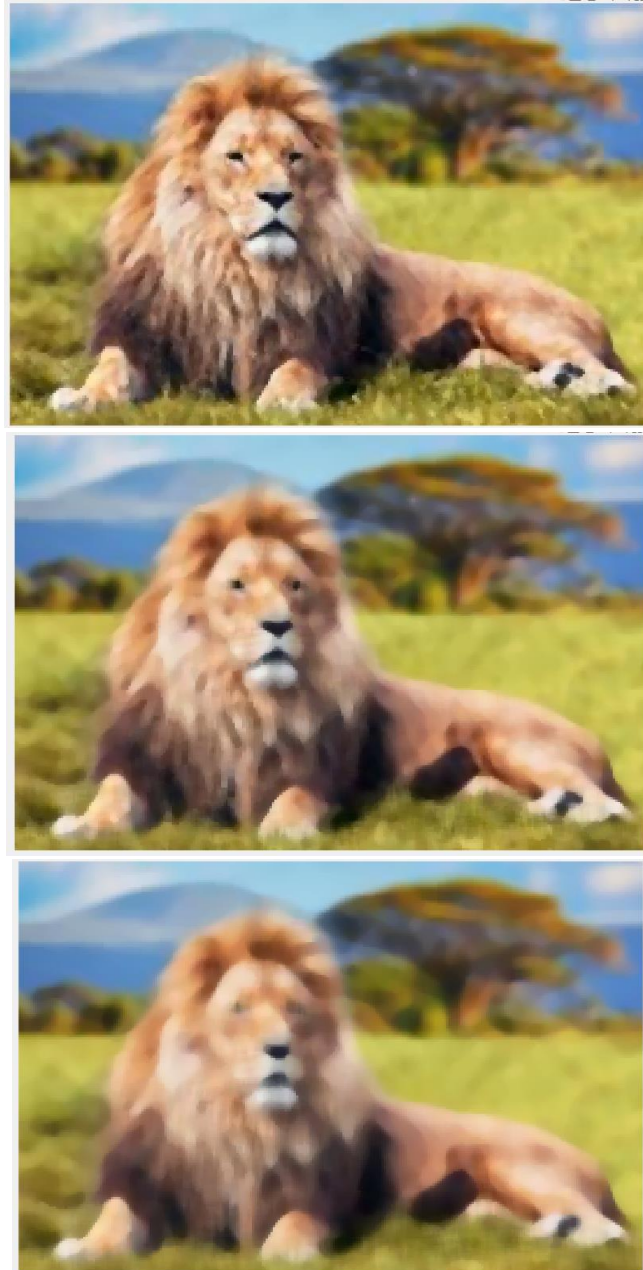
```
image = imread('image.png');
a = ones(3,3);
out = imconvolution(image, a, 'Median');
figure();
imshow(out);
imwrite(out, 'Q2_Part2_Filter3_Image.png');
b = ones(5,5);
out = imconvolution(image, b, 'Median');
figure();
imshow(out);
```

```

imwrite(out, 'Q2_Part2_Filter5_Image.png');
c = ones(7,7);
out = imconvolution(image, c, 'Median');
figure();
imshow(out);
imwrite(out, 'Q2_Part2_Filter7_Image.png');
imshow(out);

```

خروجی این تابع به ترتیب به صورت زیر است:



(پ)

در این قسمت همانند قسمت اول عمل میکنیم با ایت تفتوت که در قسمت اول status را بر روی حالت 'average' گذاشتیم ، اما انجا status را بر روی حالت 'guassin' قرار میدهم و در

آرگومان تابع fspecial به جای 'average' باز هم 'gaussian' را قرار میدهیم و عدد ۱/۵ را هم به عنوان انحراف معیار میگذاریم.

```
image = imread('image.png');
out = imconvolution(image, fspecial('gaussian',3, 1.5), 'gaussian');
figure();
imshow(out);
imwrite(out, 'Q2_Part3_Filter3_Image.png');
out = imconvolution(image, fspecial('gaussian',5, 1.5), 'gaussian');
figure();
imshow(out);
imwrite(out, 'Q2_Part3_Filter5_Image.png');
out = imconvolution(image, fspecial('gaussian',7, 1.5), 'gaussian');
figure();
imshow(out);
imwrite(out, 'Q2_Part3_Filter7_Image.png');
imshow(out);
```

خروجی کد فوق به ترتیب:

