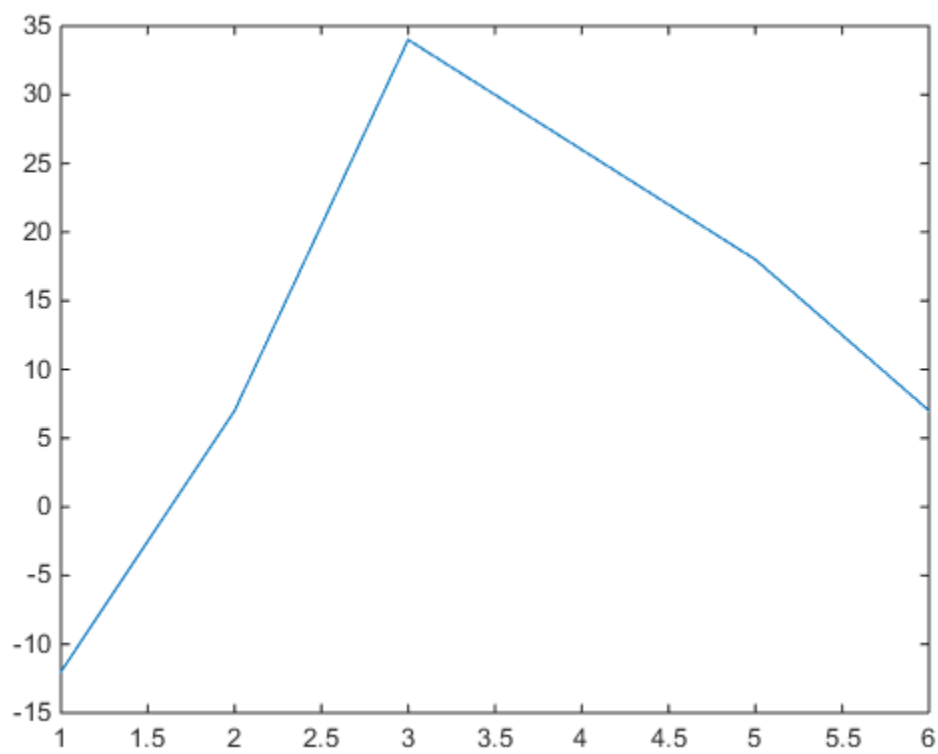
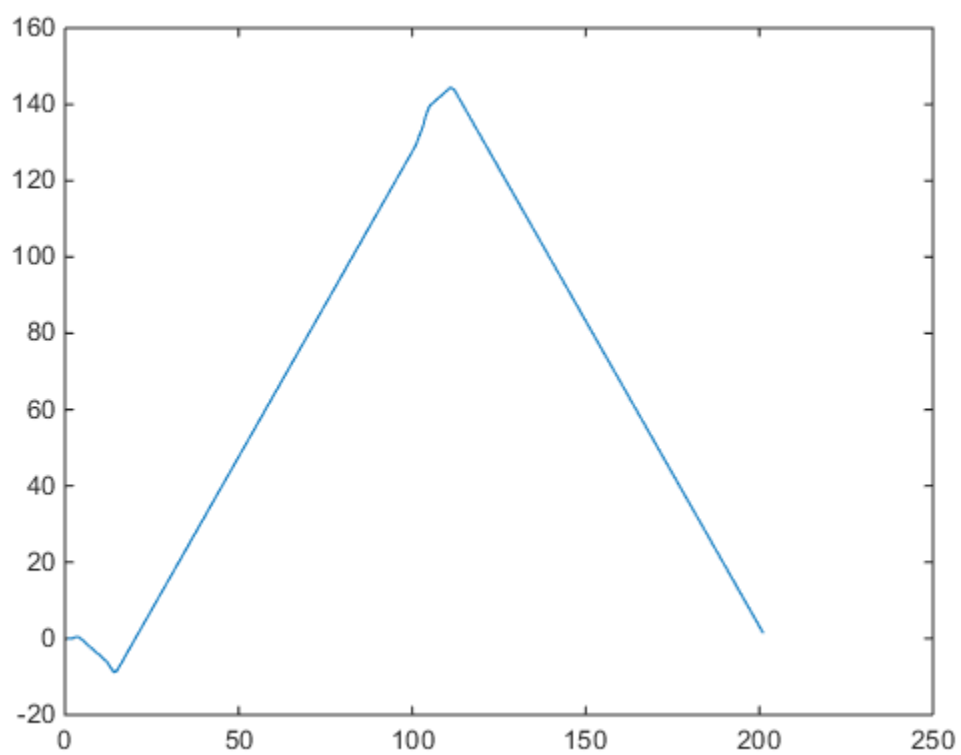


سوال اول:

تصویر اول مربوط به تابع  $a$  و تصویر دوم مربوط به تابع  $b$  است .

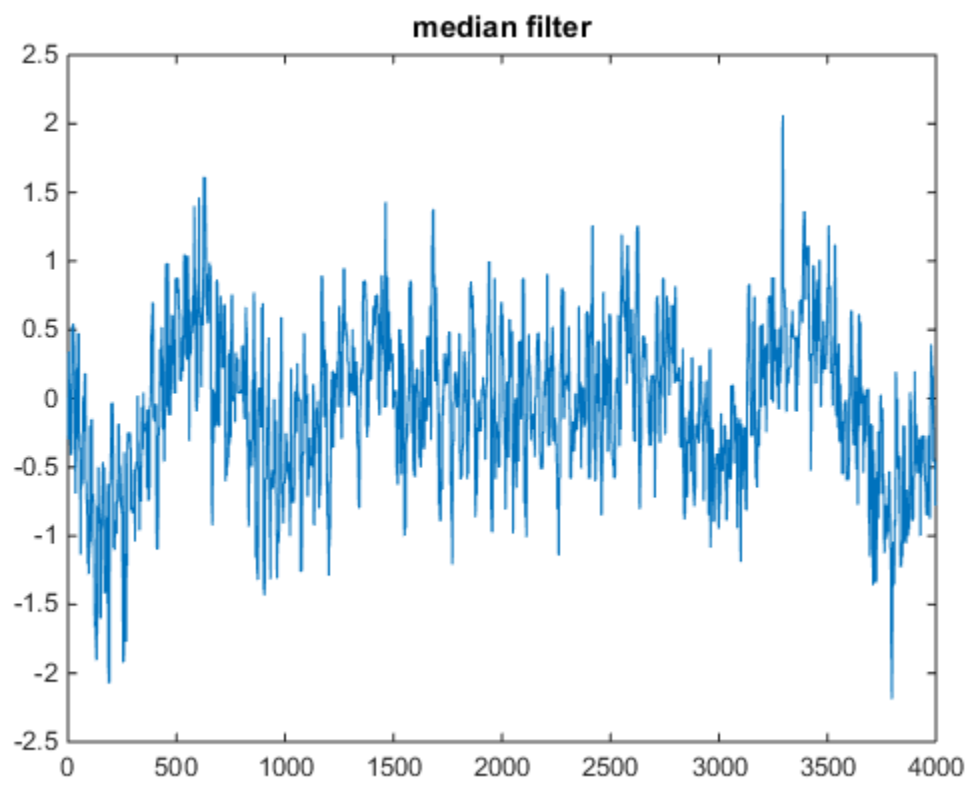
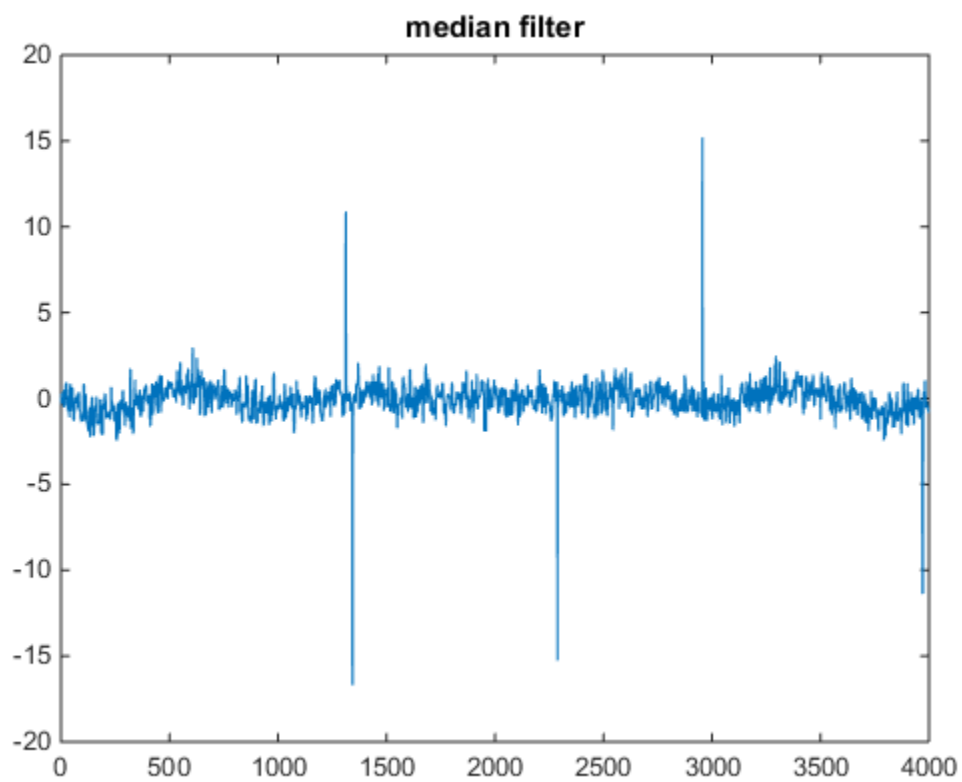
```
a = [-3,4,7];  
b = [4,3,2,1];  
w = conv(a,b);  
figure(1);  
plot(w);  
  
n = 0 : 1 : 100;  
x = heaviside(n+2) - 3*heaviside(n-10);  
h = 0.8 * ( heaviside(n-2) - 2* heaviside(n-3));  
  
z = conv(x,h);  
figure(2);  
plot(z);
```

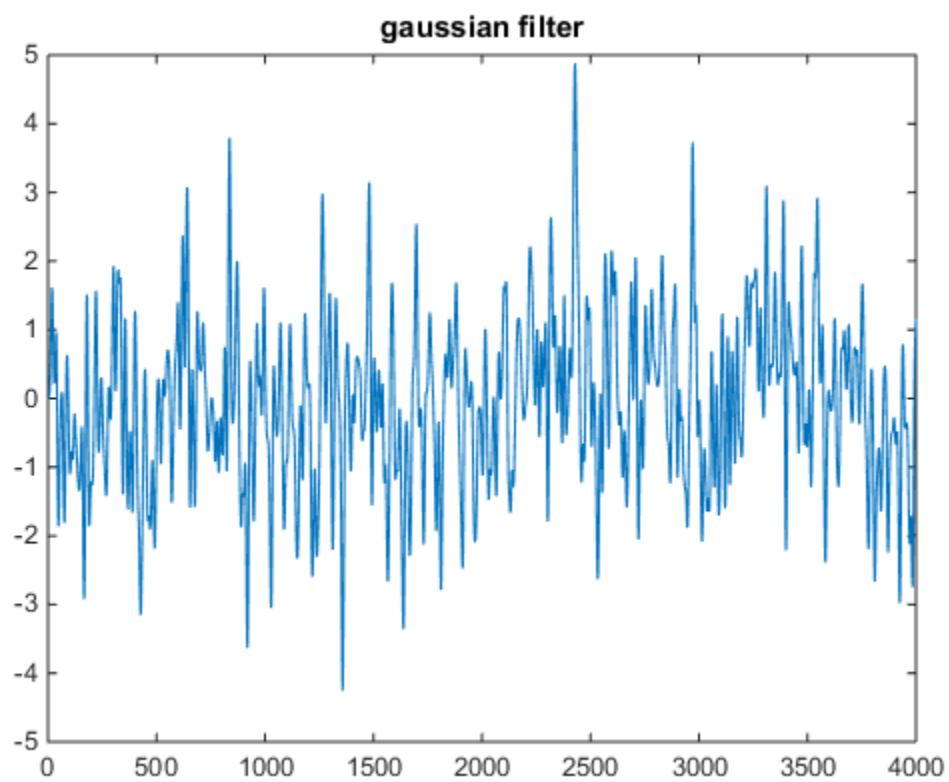
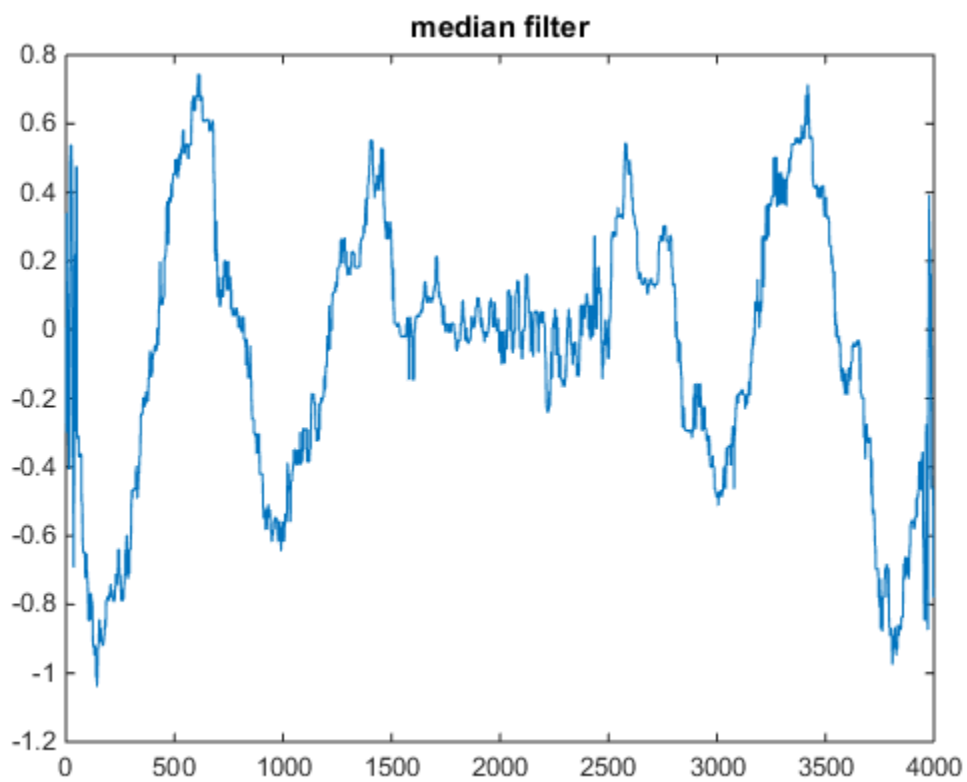


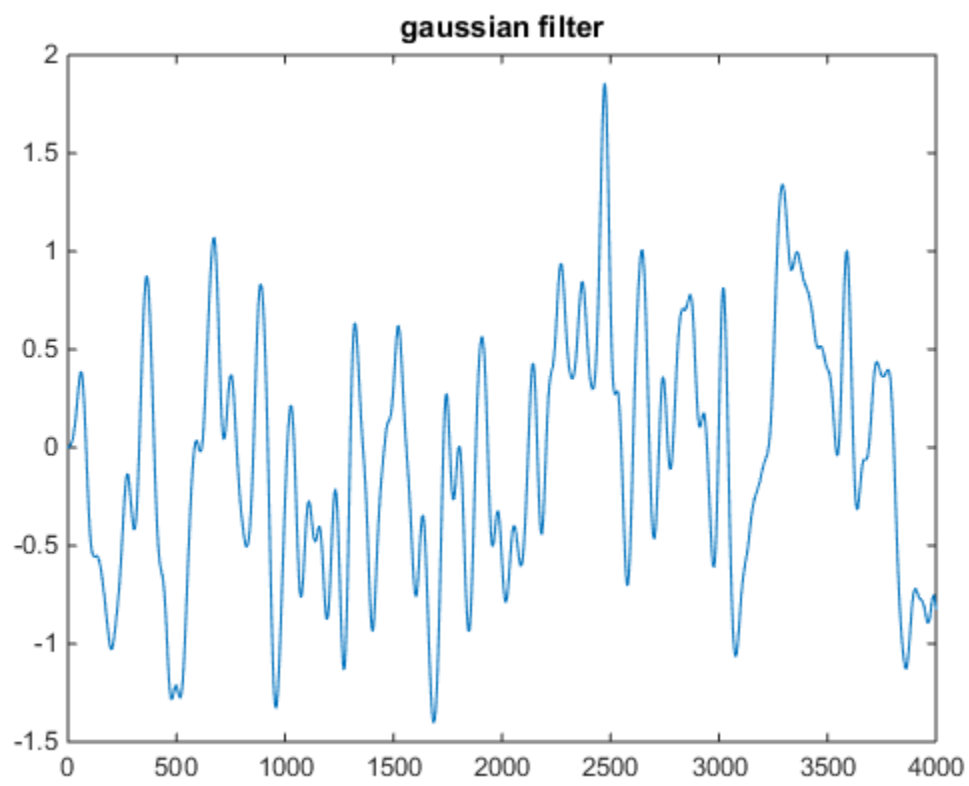
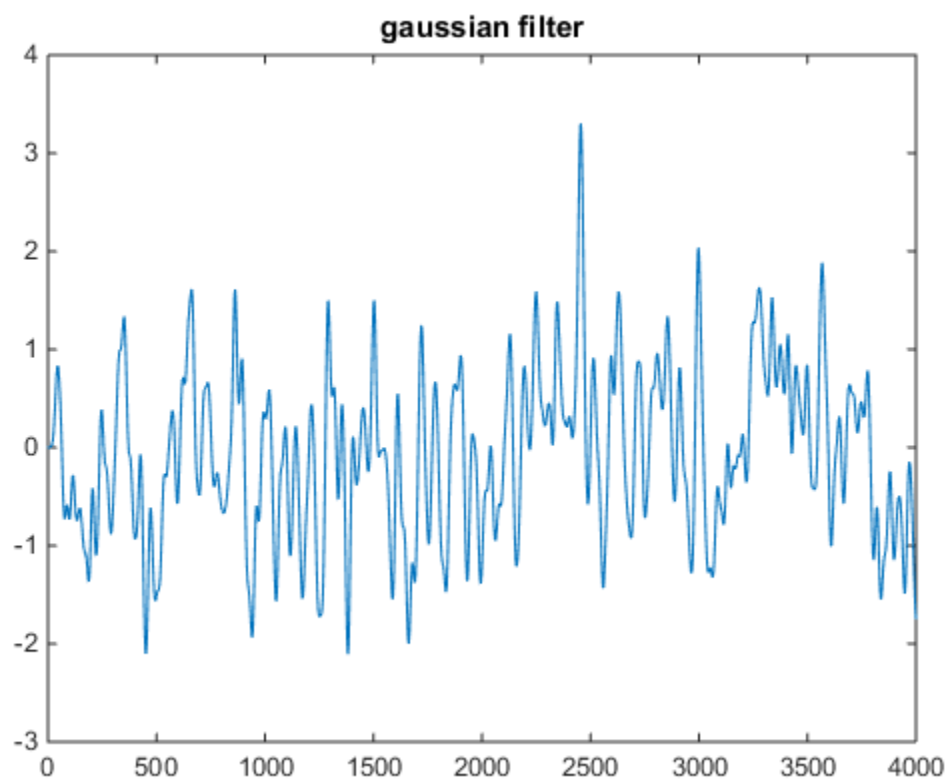


سوال دوم :

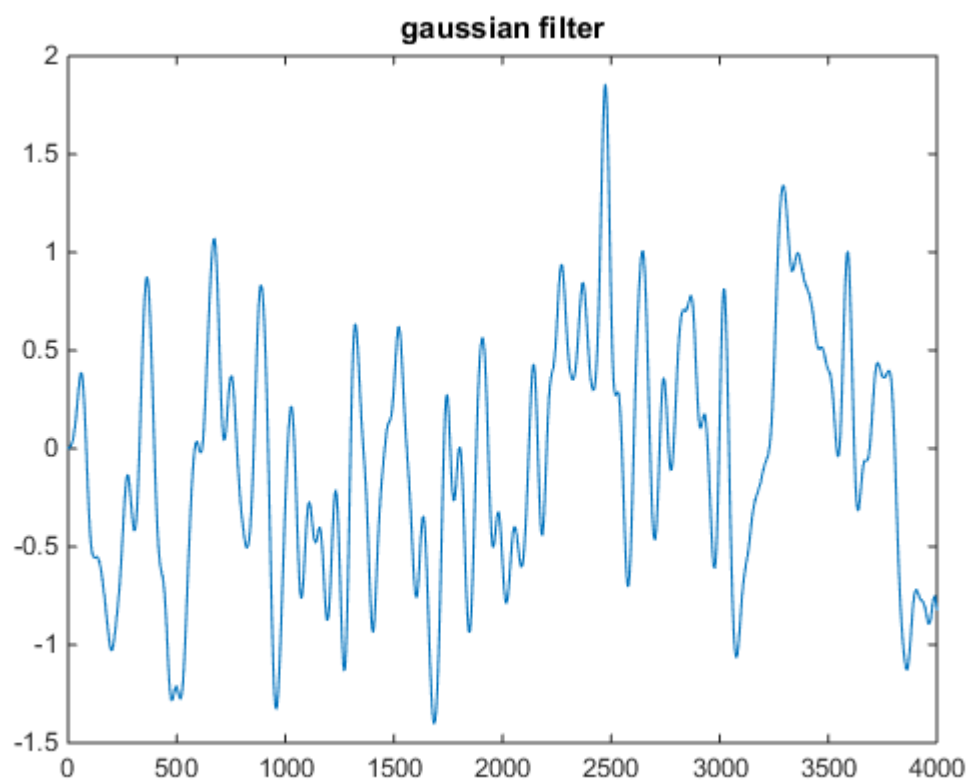
تصویرها به ترتیب برای window size 5,10,100 برای median و سپس Gaussian فیلتر هستند.

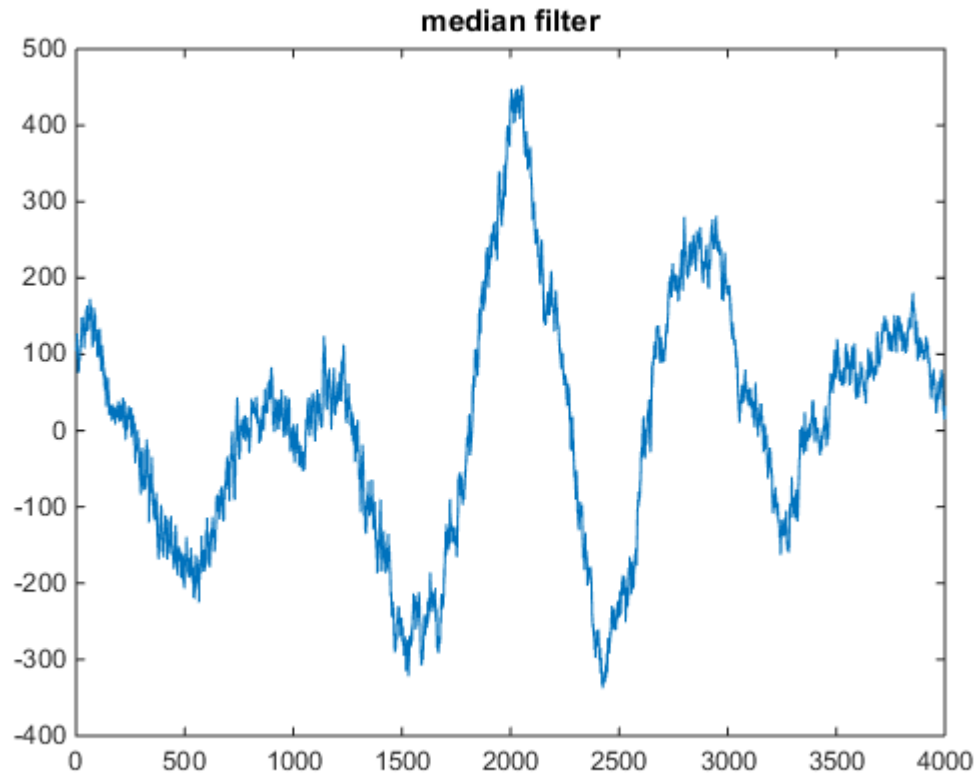






با استفاده از کانولوشن نیز میتوان این فیلتر ها را پیاده سازی کرد و در این حالت سیگنال فیلتر شده نسبت به سیگنال اولیه تاخیری نخواهد داشت.





سوال سوم :

تصویر اول حالت ابتدایی تصویر است و تصویر دوم box blur filtered و سومین تصویر با فیلتر gaussian blur می باشد.

```
p = im2double(imread('tree.png'));
bpout = p;
figure(1);
imshow(p);
bkernel=([1,1,1];[1,1,1];[1,1,1])/9; % box blur kernel
gkernel=([1,2,1];[2,4,2];[1,2,1])/16; % gaussian blur kernel
for x = drange(2:299)
    for y= drange(2:479)
        for c = drange(1:3)
            center = gkernel(3,3)*p(x-1,y-1,c);
            center = gkernel(3,2)*p(x-1,y,c) + center;
            center = gkernel(3,1)*p(x-1,y+1,c) + center;
            center = gkernel(2,3)*p(x,y-1,c) + center;
            center = gkernel(2,2)*p(x,y,c) + center;
            center = gkernel(2,1)*p(x,y+1,c) + center;
            center = gkernel(1,3)*p(x+1,y-1,c) + center;
            center = gkernel(1,2)*p(x+1,y,c) + center;
            center = gkernel(1,1)*p(x+1,y+1,c) + center;
            bpout(x,y,c) = center;
        end
    end
end
```

```

end

figure(2);
imshow(bpout);

for x = drange(2:299)
    for y= drange(2:479)
        for c = drange(1:3)
            center = bkernel(3,3)*p(x-1,y-1,c);
            center = bkernel(3,2)*p(x-1,y,c) + center;
            center = bkernel(3,1)*p(x-1,y+1,c) + center;
            center = bkernel(2,3)*p(x,y-1,c) + center;
            center = bkernel(2,2)*p(x,y,c) + center;
            center = bkernel(2,1)*p(x,y+1,c) + center;
            center = bkernel(1,3)*p(x+1,y-1,c) + center;
            center = bkernel(1,2)*p(x+1,y,c) + center;
            center = bkernel(1,1)*p(x+1,y+1,c) + center;
            gpout(x,y,c) = center;
        end
    end
end

figure(3);
imshow(gpout);

```





