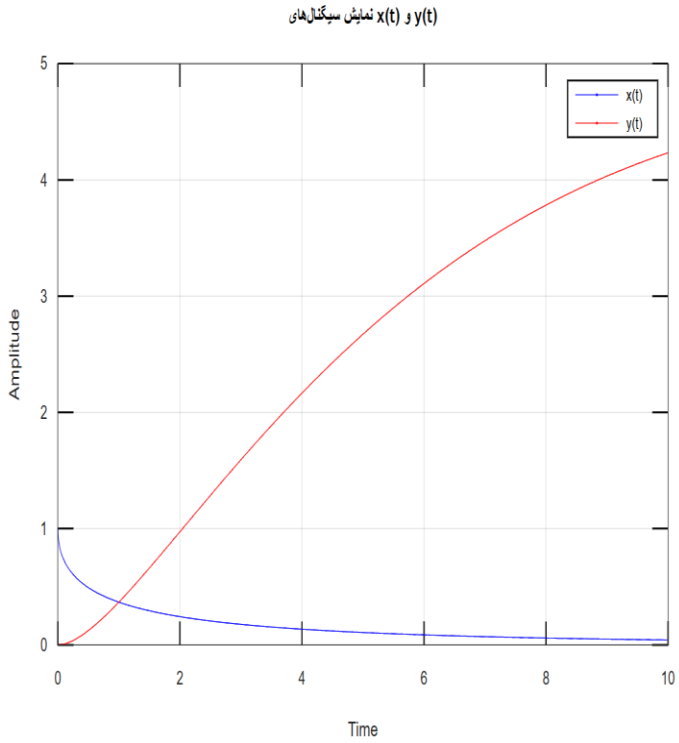
به نام خدا

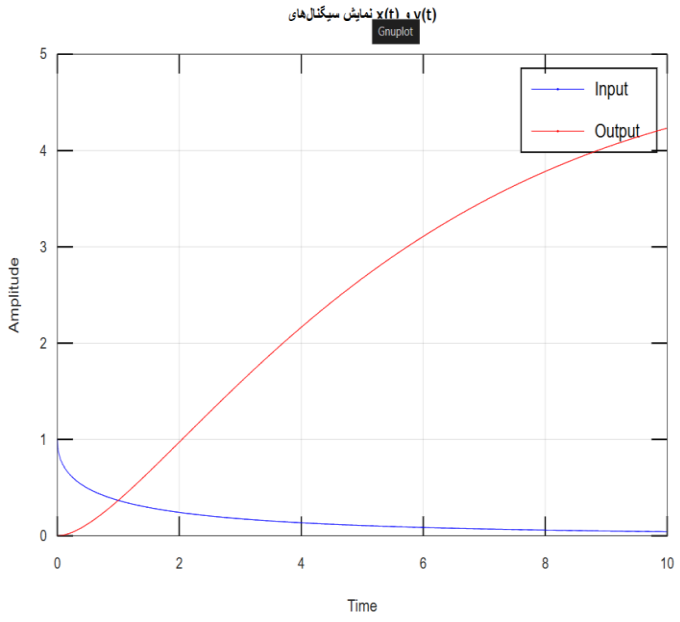
محمد امین طهماسبی نیا چناری

سوال ۱ :



ب)

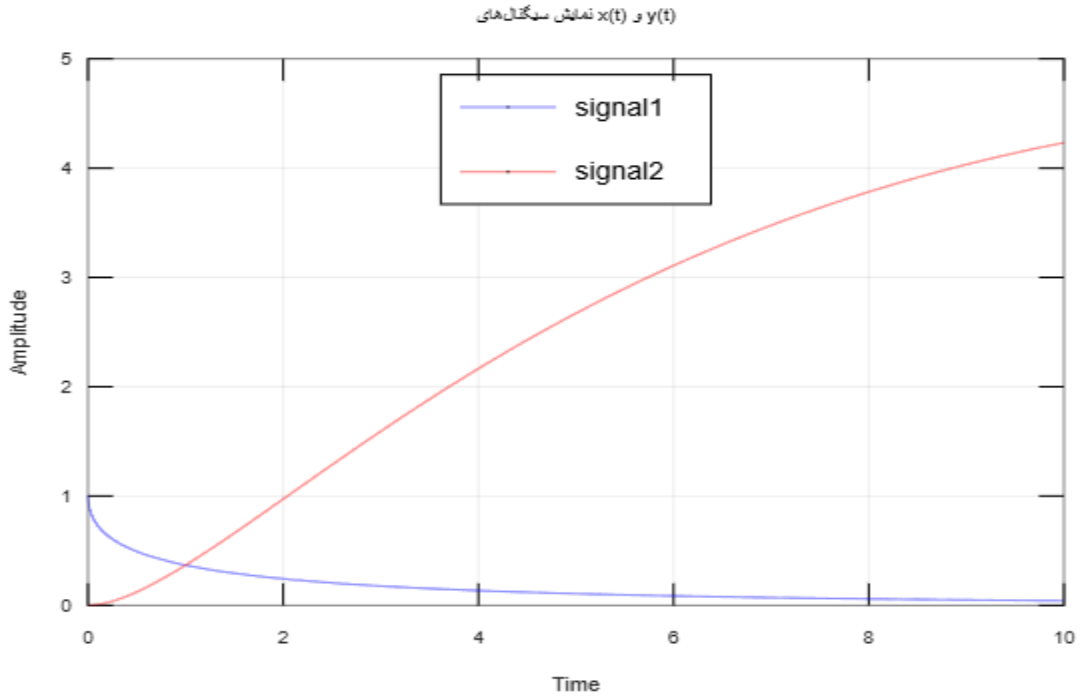
این دستور باعث تغیر موقعیت و اندازه بلاک بالا شده و موقعیت آن نیز تغیر میدهد و اسم x(t) و y(t) به input و output تبدیل میشود ضخامت فونت ها نیز ۱٫۵ برابر خواهد شد و اندازه فونت های را نیز مدیریت کرده و نوع فونت را نیز فونت legend میگذارد.



ج)

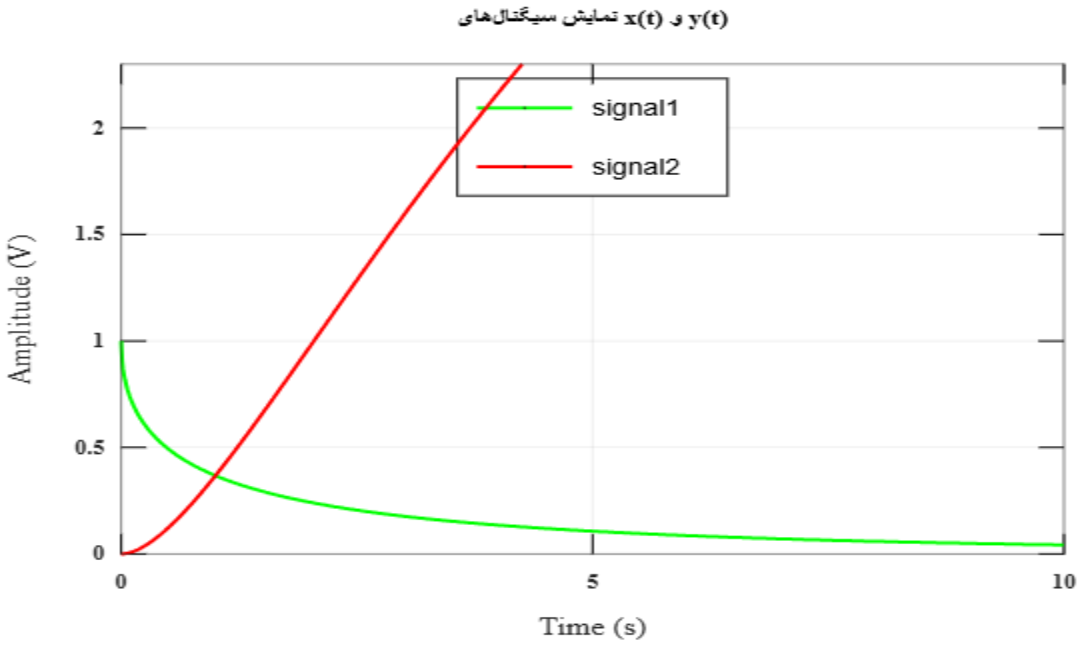
```
t = 0:0.01:10;
x = exp(-t.^0.5) .* (t >= 0);
y = (t.^2) .* exp(-t.^0.5) .* (t >= 0);

figure;
plot(t, x, 'b', t, y, 'r');
xlabel('Time');
ylabel('Amplitude');
title('نمایش سیگنال‌های x(t) و y(t)');
legend('x(t)', 'y(t)');
grid on;
% جواب بخش الف
[~, hobj, ~, ~]=legend({'signal1','signal2'},'Fontsize',14
    ,'Location'
,'North');
hl = findobj(hobj,'type','line');
set(hl,'LineWidth',2);
ht = findobj(hobj,'type','text')
set(ht,'FontSize', 14, 'fontname', 'times');

```

همان طور که دیده میشود فونت سایز تغییر داده شده و نوع فونت نیز تغییر کرده است. نام دو سیگنال را هم به سیگنال ۱ و سیگنال ۲ تغییر دادیم.



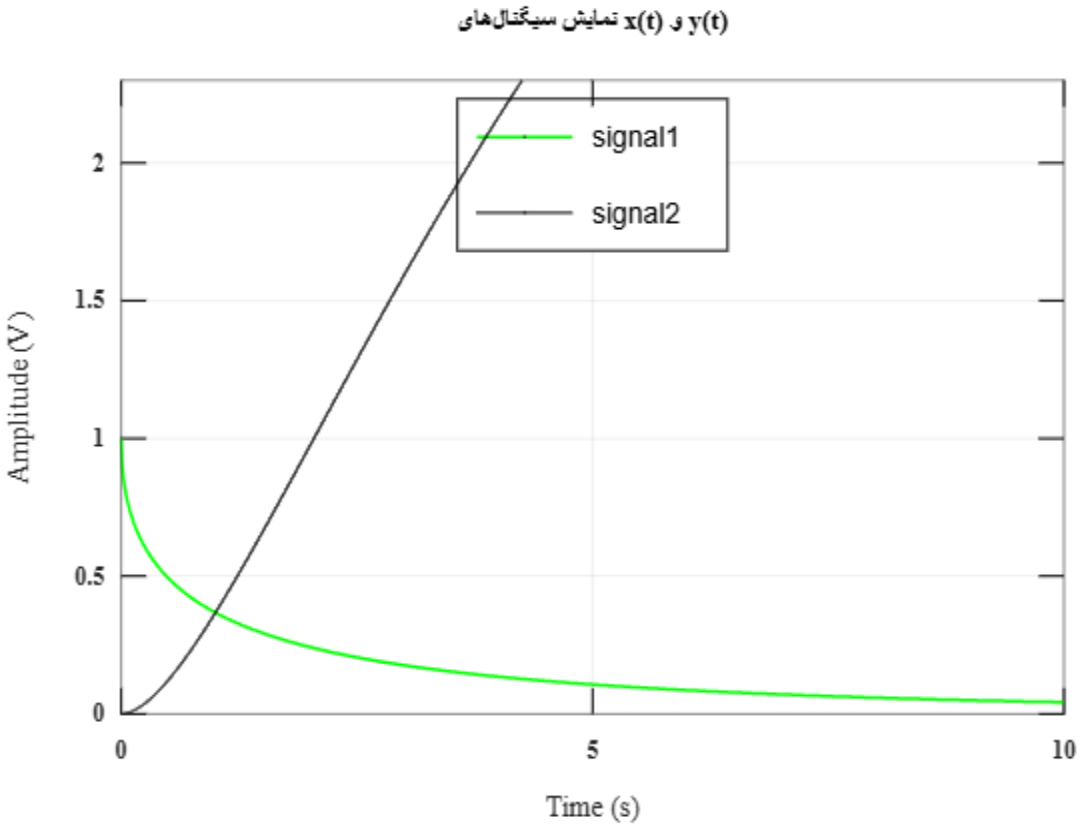
د)

باعث تغییر رنگ نمودار ها و محدود کردن نمایش نمودار ها درواقع زوم روی نمودار تغیر وزن فونت ها و نوع آنها و تغیر فونت لیبل ها و تغییر شماره فونت اعداد روی نمودار شده است.

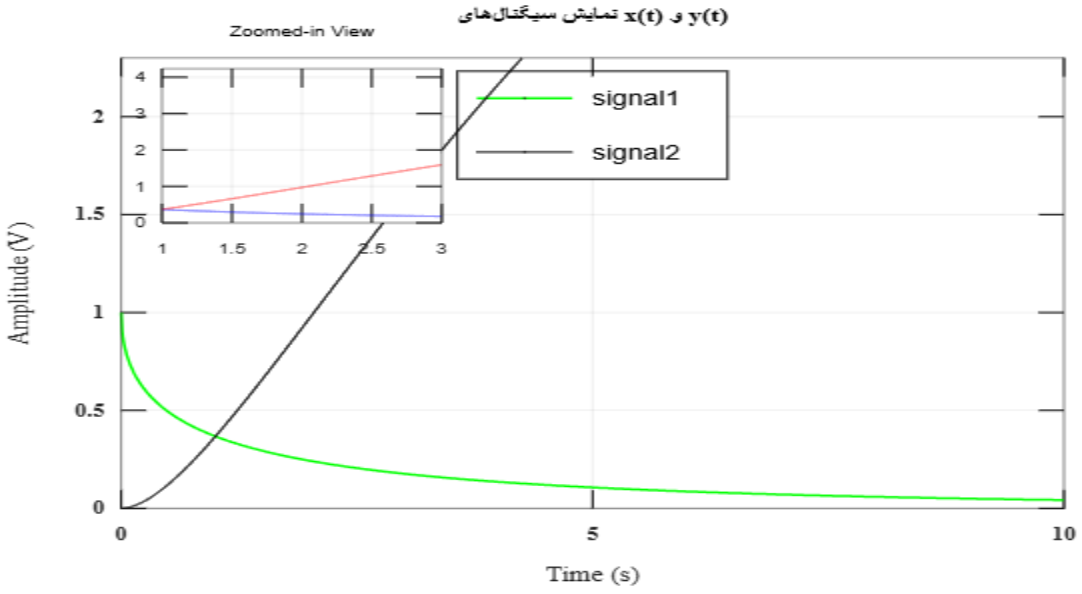


۵)

```
h0 = gca;
set(h0,'xtick',[0 5 10],'ytick',.5*(0:6))
set(h0,'ylim',[0, 2.3])
set(h0,'fontsize',12,'fontname','times','fontweight','bold')

set(get(h0,'xlabel'),'string','Time (s)','fontsize',14
    ,'fontname','helivetka','fontweight','normal')
set(get(h0,'ylabel'),'string','Amplitude (V)','fontsize',14
    ,'fontname','helivetka','fontweight','normal')
h1 = findobj(h0,'type','line');
col = [0 0 0; 0 1 0]; % First row is black [0 0 0], second row
    is green [0 1 0]

set(h1(1),'color',col(1,:),'linewidth',1) % Black Line
set(h1(2),'color',col(2,:),'linewidth',1.5) % Green Line
```

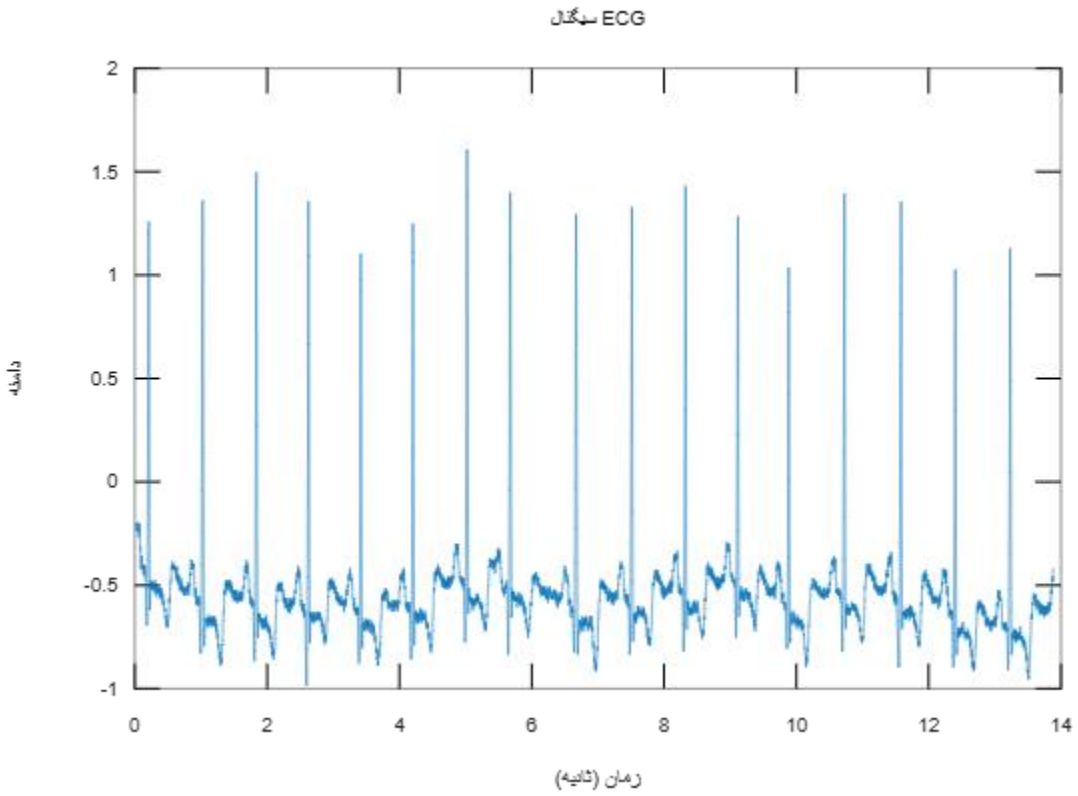


```
20  % جواب بخش ه
21  h0 = gca;
22  set(h0,'xtick',[0 5 10],'ytick',.5*(0:6))
23  set(h0,'ylim',[0, 2.3])
24  set(h0,'fontsize',12,'fontname','times','fontweight','bold')
25
26  set(get(h0,'xlabel'),'string','Time (s)','fontsize',14
        ,'fontname','helivetka','fontweight','normal')
27  set(get(h0,'ylabel'),'string','Amplitude (V)','fontsize',14
        ,'fontname','helivetka','fontweight','normal')
28  h1 = findobj(h0,'type','line');
29  col = [0 0 0; 0 1 0]; % First row is black [0 0 0], second row
        is green [0 1 0]
30
31  set(h1(1),'color',col(1,:),'linewidth',1) % Black Line
32  set(h1(2),'color',col(2,:),'linewidth',1.5) % Green Line
33
34  % zoom part
35  axes('Position',[0.15 0.6 0.25 0.25])
36  plot(t, x, 'b', t, y, 'r');
37  axis([1 3 min(min(x), min(y)) max(max(x), max(y))]); % Zoom in
        from 1 to 3 on x-axis
38
39  title('Zoomed-in View');
40  grid on;
```



سوال ۲

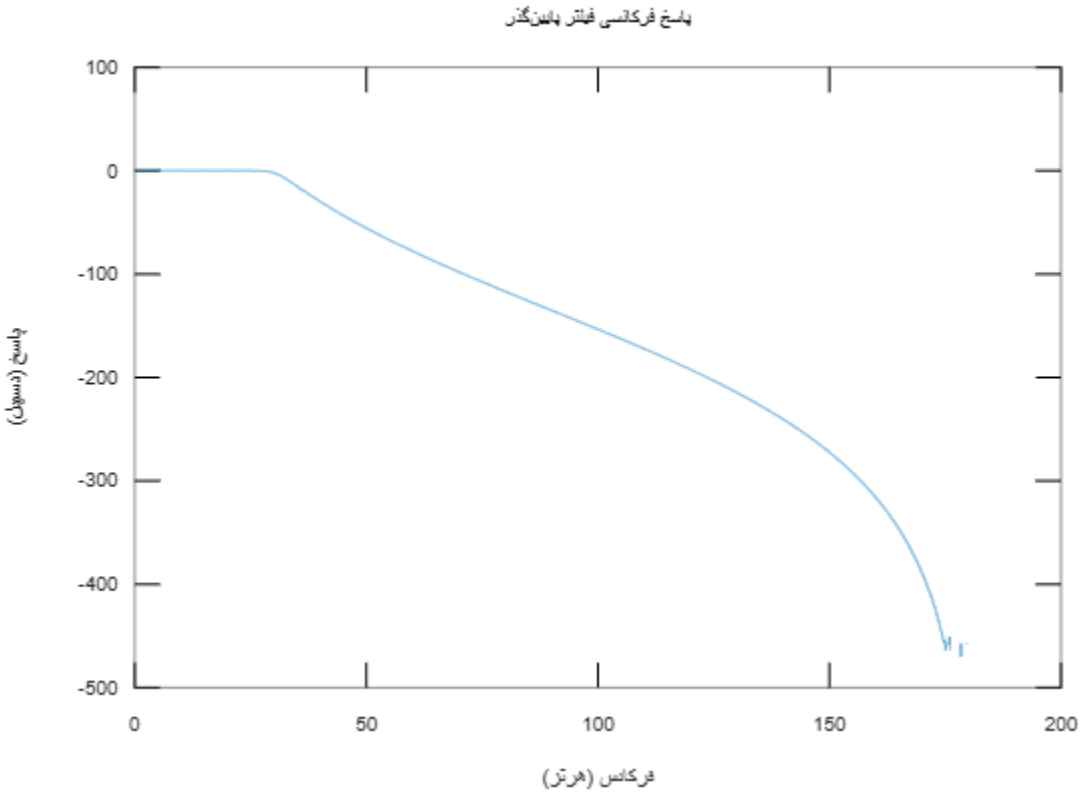
الف)




```
% خواندن سیگنال ECG
load('ecg_example (1).mat');

% تعریف محور زمان
fs = 360; % فرکانس نمونه‌برداری
t = (0:length(y)-1) / fs;

% رسم سیگنال
figure;
plot(t, y);
xlabel('زمان (ثانیه)');
ylabel('دامنه');
title('سیگنال ECG');
```

سیگنال مورد نظر از فایل ecg خوانده شده است. و توسط تابع plot رسم شده است.

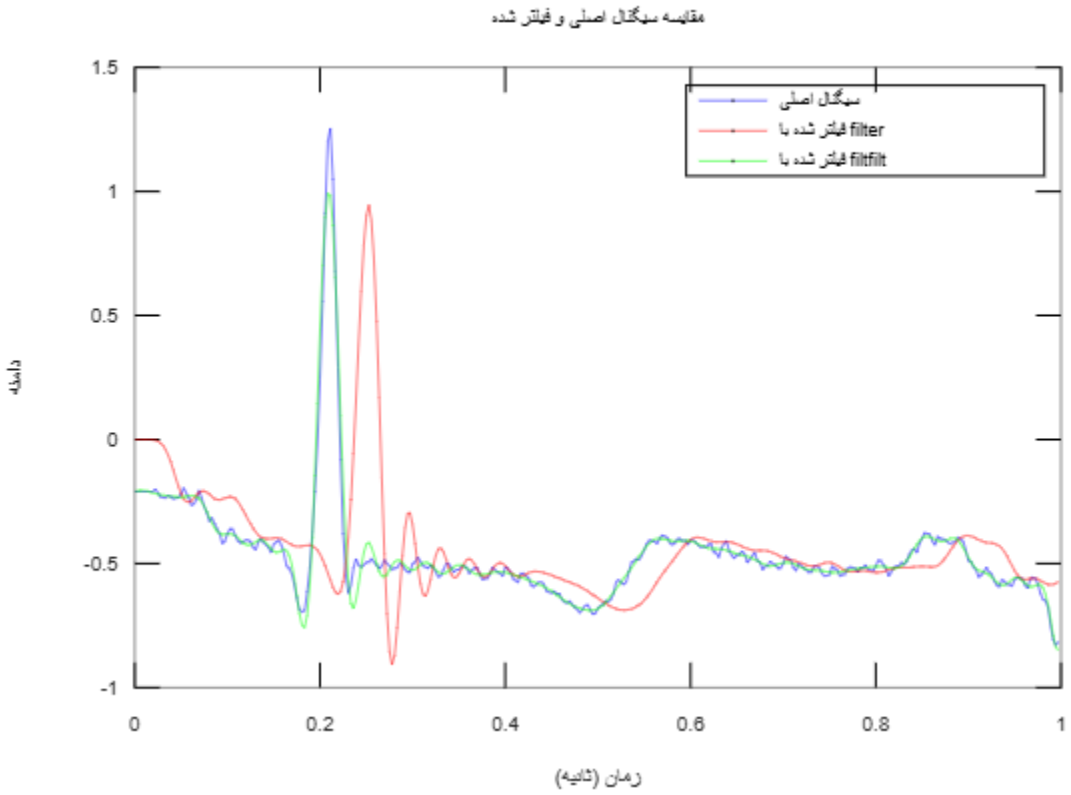
ب)




```
14  %--ب--
15
16
17  % طراحی فیلتر پایین‌گذر
18  order = 12;
19  wn = 0.17; % فرکانس نرمالیزه شده
20  [b, a] = butter(order, wn, 'low');
21
22  % رسم پاسخ فرکانسی فیلتر
23  [h, w] = freqz(b, a, 1024, fs);
24  figure;
25  plot(w, 20*log10(abs(h)));
26  xlabel('فرکانس (هرتز)');
27  ylabel('پاسخ (دسیبل)');
28  title('پاسخ فرکانسی فیلتر پایین‌گذر');
29
```

یک فیلتر پایین گذر با مرتبه ۱۲ با استفاده از butter

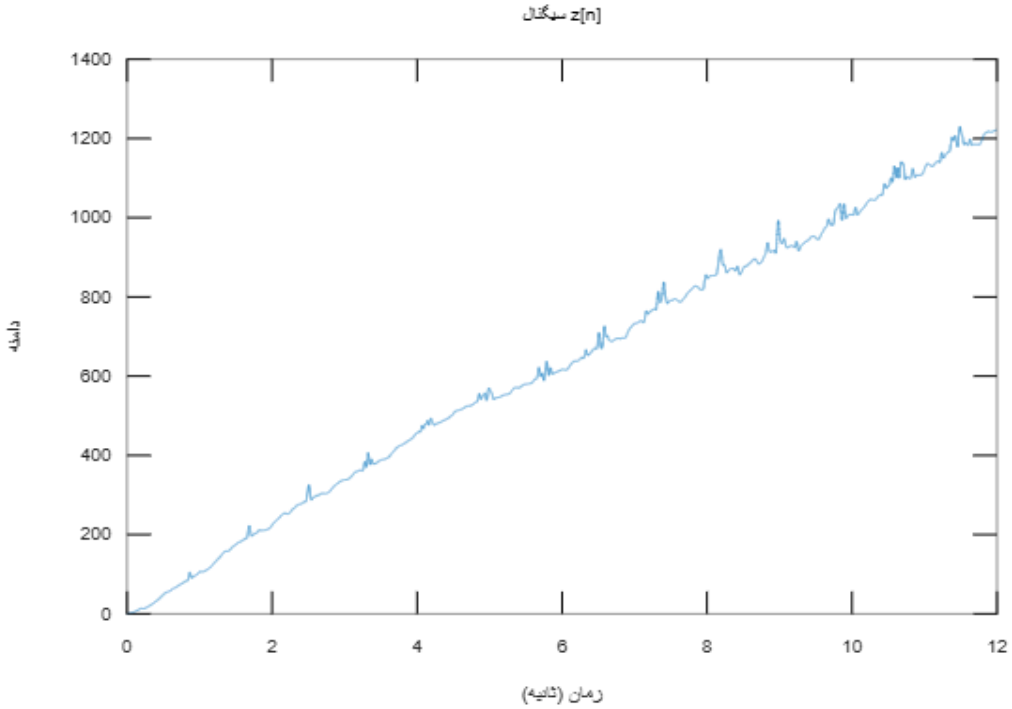
ج)




```
%--ج--
% بارگذاری سیگنال ECG

% تعریف پارامترهای فیلتر
% اعمال فیلتر با استفاده از filter
filtered_signal1 = filter(b, a, y);

% اعمال فیلتر با استفاده از filtfilt
filtered_signal2 = filtfilt(b, a, y);

% نمایش نتایج
figure;
t_window = t(1:360); % پنجره 1 ثانیه‌ای
plot(t_window, y(1:360), 'b', t_window, filtered_signal1(1:360),
    'r', t_window, filtered_signal2(1:360), 'g');
legend('سیگنال اصلی' ,'فیلتر شده با filter', 'فیلتر شده با
    filtfilt');
xlabel('زمان (ثانیه)');
ylabel('دامنه');
title('مقایسه سیگنال اصلی و فیلتر شده');
```

همان طور که در شکل مشاهده میشود این فیلتر را با دو دستور filtfilt و filter انجام دادم همان طور که در شکل دیده میشود سیگنال فیلتر شده توسط filtfilt با سیگنال سینوسی اصلی هم فاز است ولی سیگنال فیلتر شده توسط filter نسبت به سیگنال اصلی دارای تاخیر فازی است.

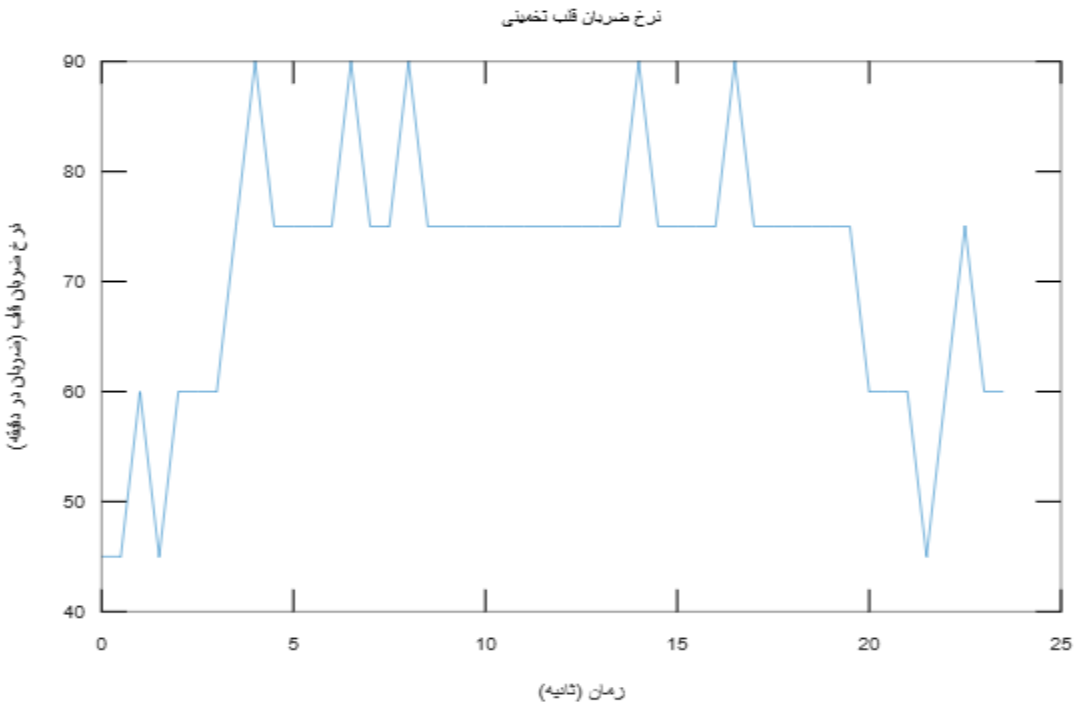
و)




```
%--و--

% محاسبه z[n]
y1 = filtered_signal2;
z = conv(y1, fliplr(y1));

% رسم z[n] در بازه (0,12)
t_z = (0:length(z)-1) / fs;
figure;
plot(t_z(1:12*fs), z(1:12*fs));
xlabel('زمان (ثانیه)');
ylabel('دامنه');
title('سیگنال z[n]');
xlim([0 12]);
```

کانکلوشن سیگنال فیلتر شده با filtfilt با معکوس زمانی خودش

ن




```
%--ز--

function heart_rate = calculate_heart_rate(z, fs)
    window_size = 4 * fs; % ای-ثانیه 4 پنجره
    overlap = 0.5 * fs; % ثانیه 0.5 همپوشانی

    heart_rate = [];
    for i = 1:overlap:(length(z)-window_size)
        window = z(i:i+window_size-1);
        [~, locs] = findpeaks(window, 'MinPeakDistance', 0.5*fs
            );
        if ~isempty(locs)
            rate = length(locs) * (60 / 4); % تبدیل به ضربان در
                دقیقه
            heart_rate = [heart_rate rate];
        end
    end
end

% محاسبه نرخ ضربان قلب
heart_rate = calculate_heart_rate(z, fs);

% نمایش نتایج
figure;
t_hr = (0:length(heart_rate)-1) * 0.5; % هر 0.5 ثانیه یک مقدار
plot(t_hr, heart_rate);
xlabel('زمان (ثانیه)');
ylabel('نرخ ضربان قلب (ضربان در دقیقه)');
title('نرخ ضربان قلب تخمینی');
```

همانطور که دیده میشود اندازه پنجره ها را ۴ قرار دادیم و همپوشانی نیز برابر ۰/۵ قرار گرفته است.

حلقه درونی به گام های fs*۰/۵ پیش می رود و تا زمانی که با انتهای سیگنال منهای اندازه پنجره نرسیده ادامه میدهد و یک پنجره ۴ ثانیه از سیگنال استخراج میکند. با استفاده از findpeak پیک های این سیگنال پیدا میشود و MinPeakDistance که برابر ۰/۵ قرار گرفته است از شناسای پیک های خیلی نزدیک جلو گیری میکند.

اگر پیکی یافت شود تعداد آن گرفته شده و این تعداد به نرخ ضربان در دقیقه تبدیل میشود.

سوال ۳

الف)

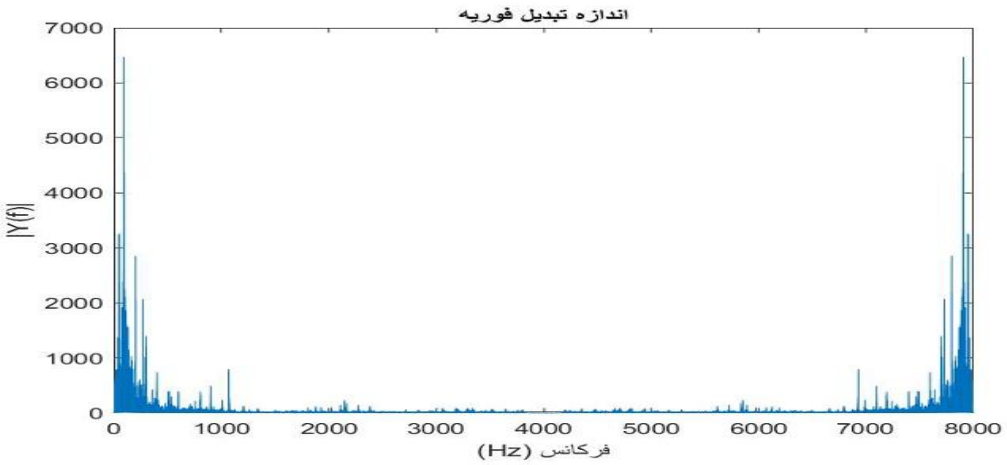
```
1  %--الف--
2  [y, Fs] = audioread('music.wav');
3  t = (0:length(y)-1) / Fs;
4  plot(t, y(:,1));
5  xlabel('(ثانیه) زمان');
6  ylabel('دامنه');
7  title('سیگنال صوتی اصلی');
8
```

ب)

```
9  %--ب--
10 player = audioplayer(y, Fs);
11 play(player);
12
```

ج)

```
%--ج--
Y = fft(y(:,1));
f = (0:length(Y)-1)*Fs/length(Y);
plot(f, abs(Y));
xlabel('فرکانس (Hz)');
ylabel('|Y(f)|');
title('اندازه تبدیل فوریه');

```



در این مورد تبدیل فوریه سیگنال رو محاسبه کرده و آن را رسم میکنیم.

د)

```
%--د--

Y_truncated = [Y(1:30000); Y(end-29999:end)];
y_recovered = real(ifft(Y_truncated));

audiowrite('recovered_30000.wav', y_recovered, Fs);
[y_play, Fs_play] = audioread('recovered_30000.wav');
player = audioplayer(y_play, Fs_play);
play(player);

```

تبدیل فوریه را به ۳۰۰۰۰ نمونه اول و آخر محدود کرده و سیگنال را با ifft بازسازی میکنیم و سیگنال بازسازی شده را پخش و ذخیره میکنیم.

ه)

```
%--ه--

Y_truncated_60000 = [Y(1:60000); Y(end-59999:end)];
y_recovered_60000 = real(ifft(Y_truncated_60000));

audiowrite('recovered_60000.wav', y_recovered_60000, Fs);
[y_play_60000, Fs_play_60000] = audioread('recovered_60000.wav'
    );
player_60000 = audioplayer(y_play_60000, Fs_play_60000);
play(player_60000);
```

و)

```
%--و--

D = dct(y(:,1));

% با 60000 نمونه
D_truncated_60000 = D(1:60000);
y_recovered_dct_60000 = idct([D_truncated_60000; zeros(length(D
    )-60000, 1)]);

audiowrite('recovered_dct_60000.wav', y_recovered_dct_60000, Fs
    );
[y_play_dct_60000, Fs_play_dct_60000] = audioread
    ('recovered_dct_60000.wav');
player_dct_60000 = audioplayer(y_play_dct_60000,
    Fs_play_dct_60000);
play(player_dct_60000);
```

```
% با 120000 نمونه
D_truncated_120000 = D(1:120000);
y_recovered_dct_120000 = idct([D_truncated_120000; zeros(length
    (D)-120000, 1)]);

audiowrite('recovered_dct_120000.wav', y_recovered_dct_120000,
    Fs);
[y_play_dct_120000, Fs_play_dct_120000] = audioread
    ('recovered_dct_120000.wav');
player_dct_120000 = audioplayer(y_play_dct_120000,
    Fs_play_dct_120000);
play(player_dct_120000);
```

در بخش و هم تبدیل کسینوسی گسسته (DCT) سیگنال را محاسبه می کند و دو نسخه برای ۶۰۰۰۰ و ۱۲۰۰۰۰ نمونه ایجاد کرده و آنها را پخش و ذخیره میکند.