DSZOB, cvičenie 4.

Zadanie:

Úloha 1 - Analýza signálu

Nasnímajte signál zvuku ladičky.

Vizualizujte nasnímaný signál v priestorovej doméne. Na základe vizualizácie vyberte vhodný úsek, vypočítajte a vizualizujte preň Fourierovu transformáciu:

- Cez funkciu **fft** na malej vzorke dát (napr. 512)
- Prepočítajte index spektrálneho koeficienta na frekvenciu pre najväčšiu hodnotu v grafe
- Cez funkciu **spectrogram**
- Zistite dominantnú frekvenciu daného signálu a vyznačte ju v grafe.

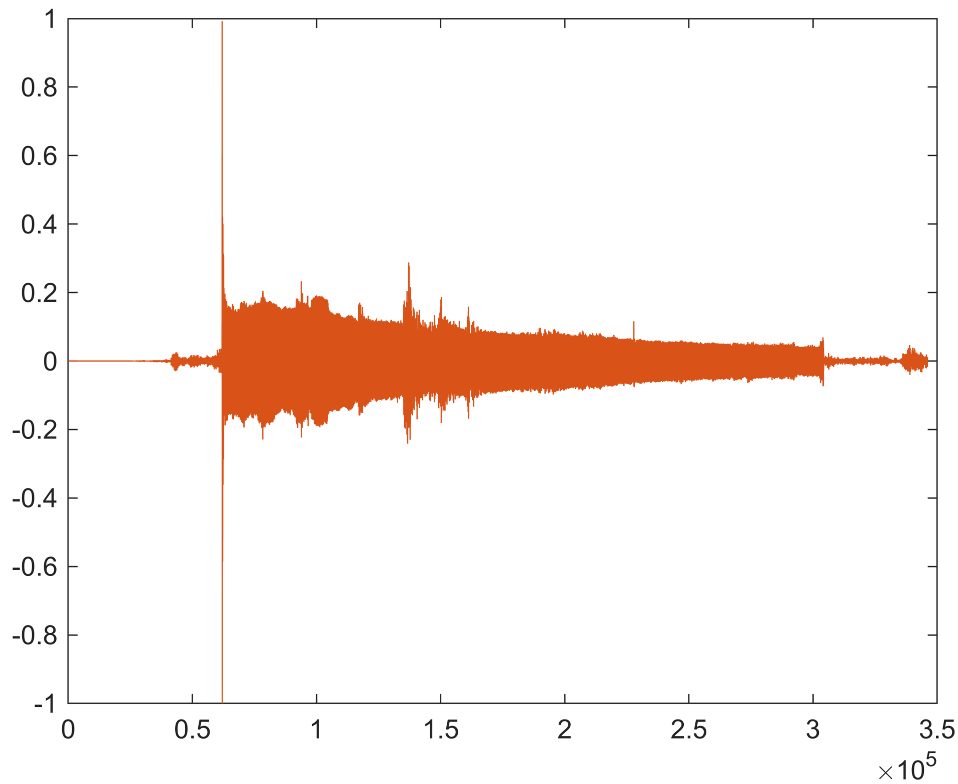
<u>Postup vhodne dokumentuje (Code/Text bloky)!</u>

Riešenie:

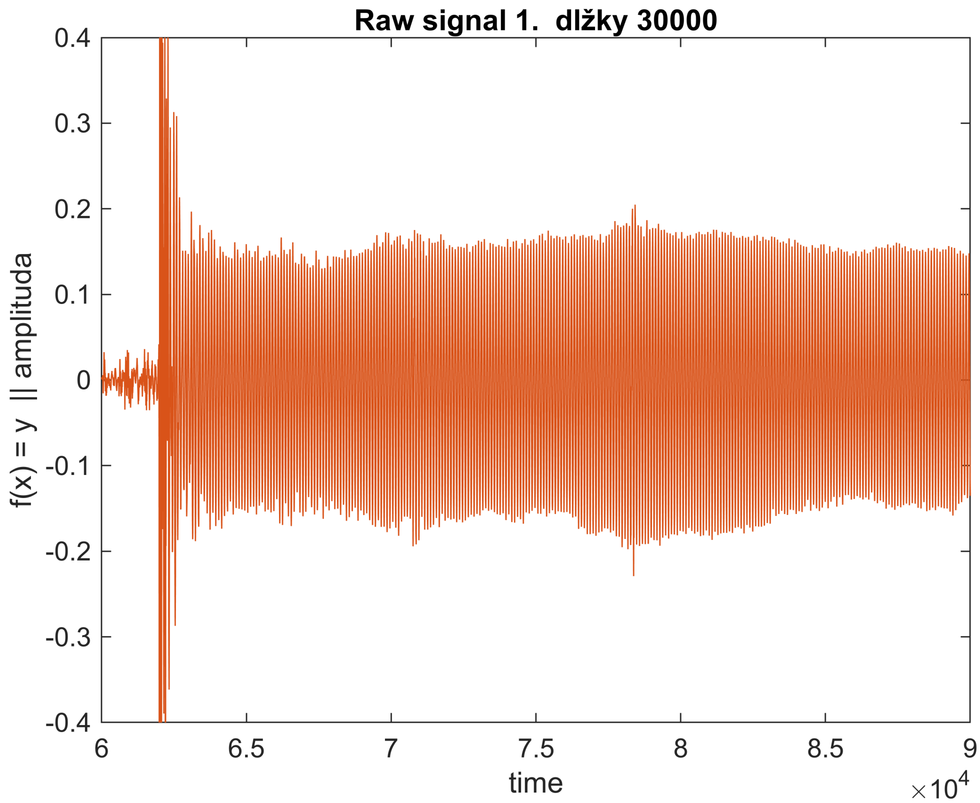
```
%1.1. Vizualizujte nasnímaný signál v priestorovej doméne.  Je ok ak FS = 4410 a
nie 44100?
ladicka = importdata("ladicka.wav")
```

```
ladicka = struct with fields:
    data: [346112×2 double]
      fs: 44100
```

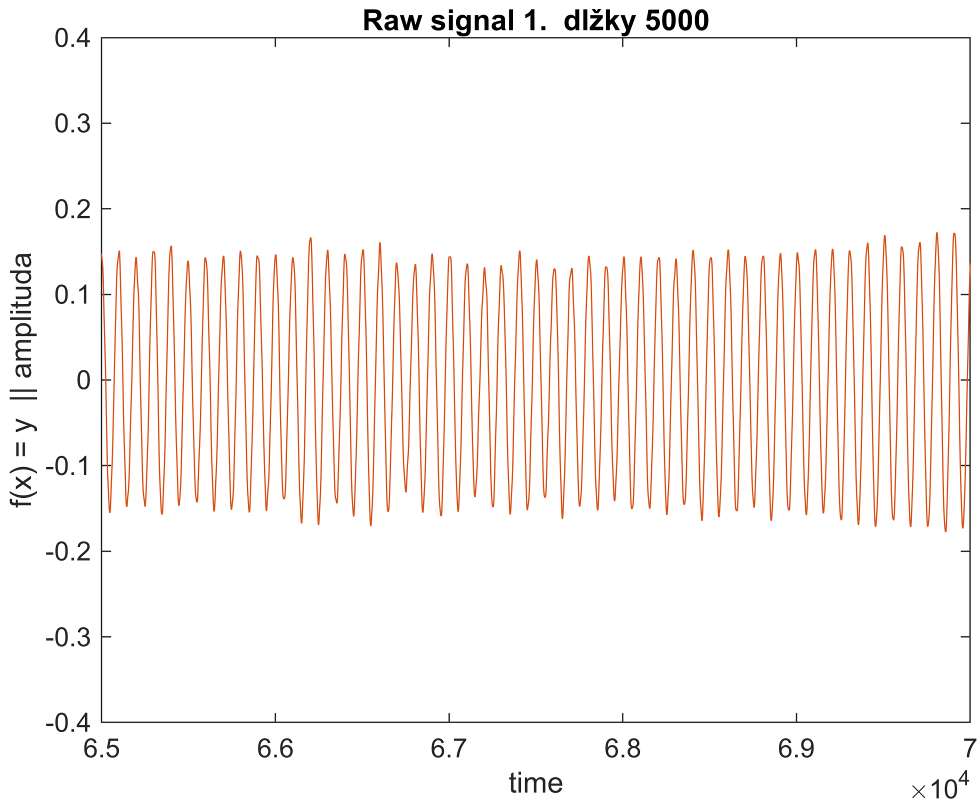
```
fado = importdata("Fado1.wav");
fs = ladicka.fs;
rawSignal = ladicka.data;
plot(rawSignal)
```



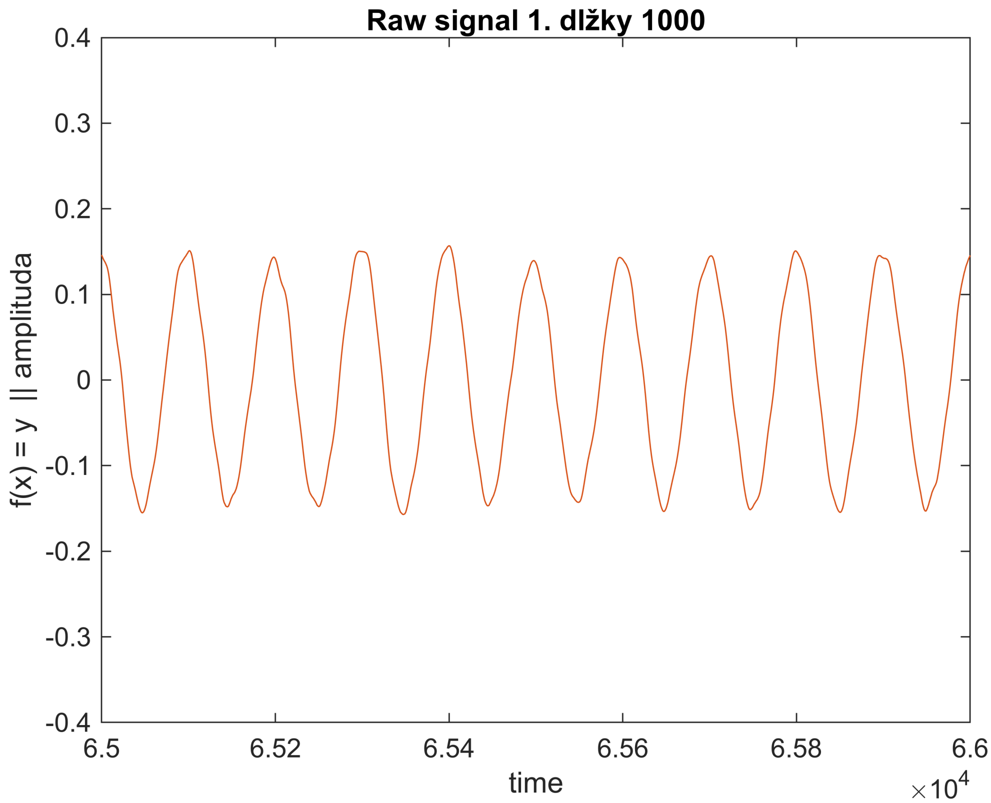

```
%1.2 Na základe vizualizácie vyberte vhodný úsek, vypočítajte a vizualizujte preň
Fourierovu transformáciu:
plot(rawSignal)
xlim([60000 90000])
ylim([-0.4 0.4])
title("Raw signal 1.  dlžky 30000")
xlabel("time")
ylabel("f(x) = y  || amplituda")
```



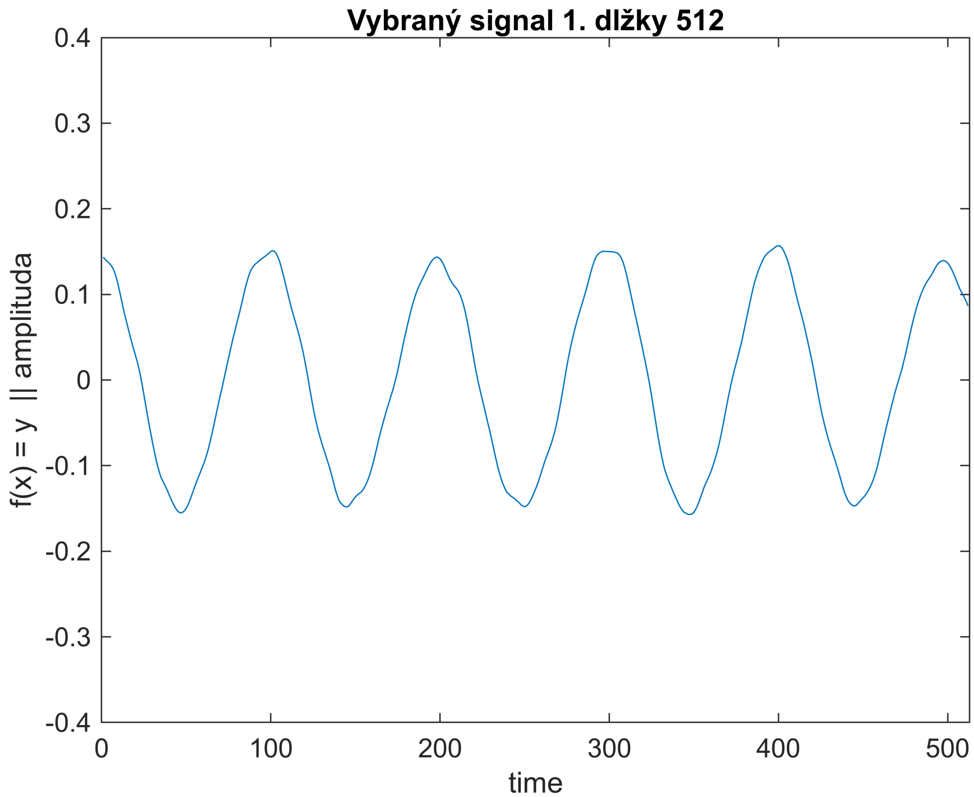

```
plot(rawSignal)
xlim([65000 70000])
ylim([-0.4 0.4])
title("Raw signal 1.  dlžky 5000")
xlabel("time")
ylabel("f(x) = y  || amplituda")
```



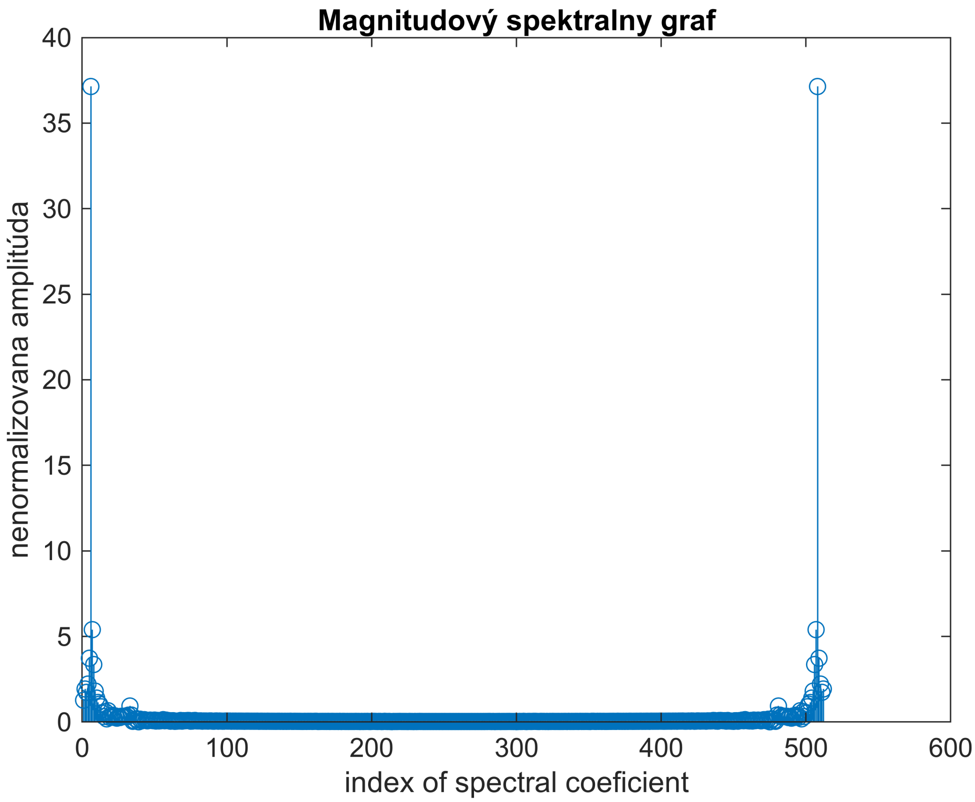

```
plot(rawSignal)
xlim([65000 66000])
ylim([-0.4 0.4])
title("Raw signal 1. dlžky 1000")
xlabel("time")
ylabel("f(x) = y  || amplituda")
```



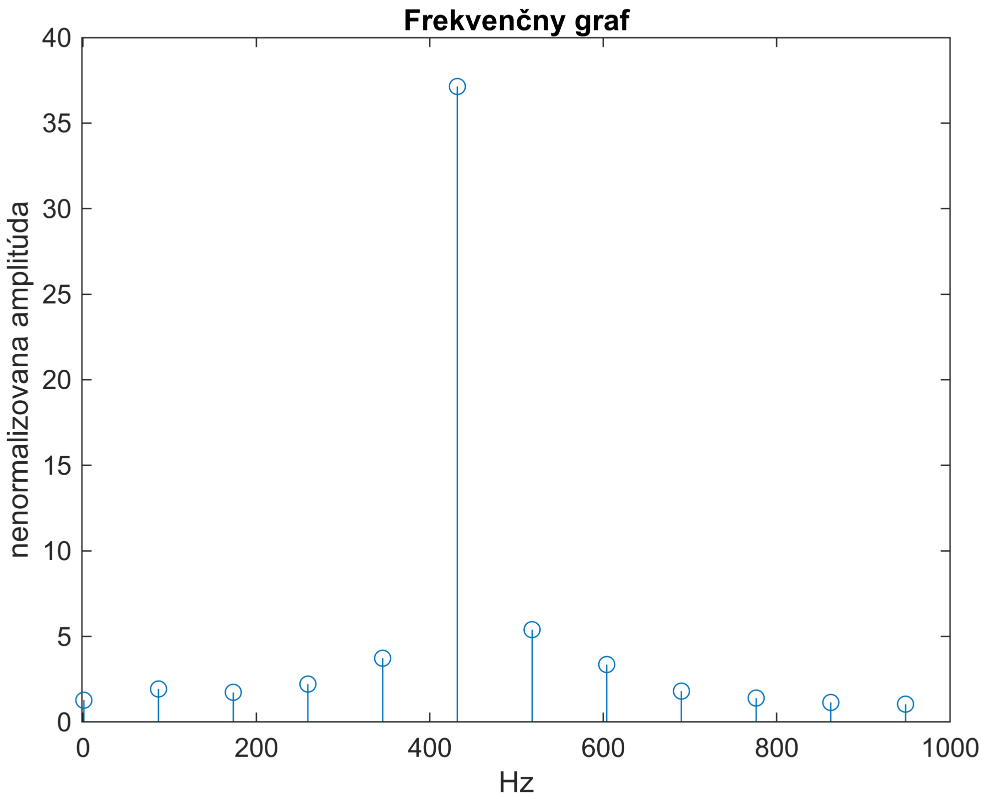

```
% 1.2.1 Cez funkciu fft na malej vzorke dát (napr. 512)
 choosenSignal = rawSignal(65001:65512);
 plot(choosenSignal)
 xlim([0 513])
 ylim([-0.4 0.4])
 title("Vybraný signal 1. dlžky 512")
 xlabel("time")
 ylabel("f(x) = y  || amplituda")
```



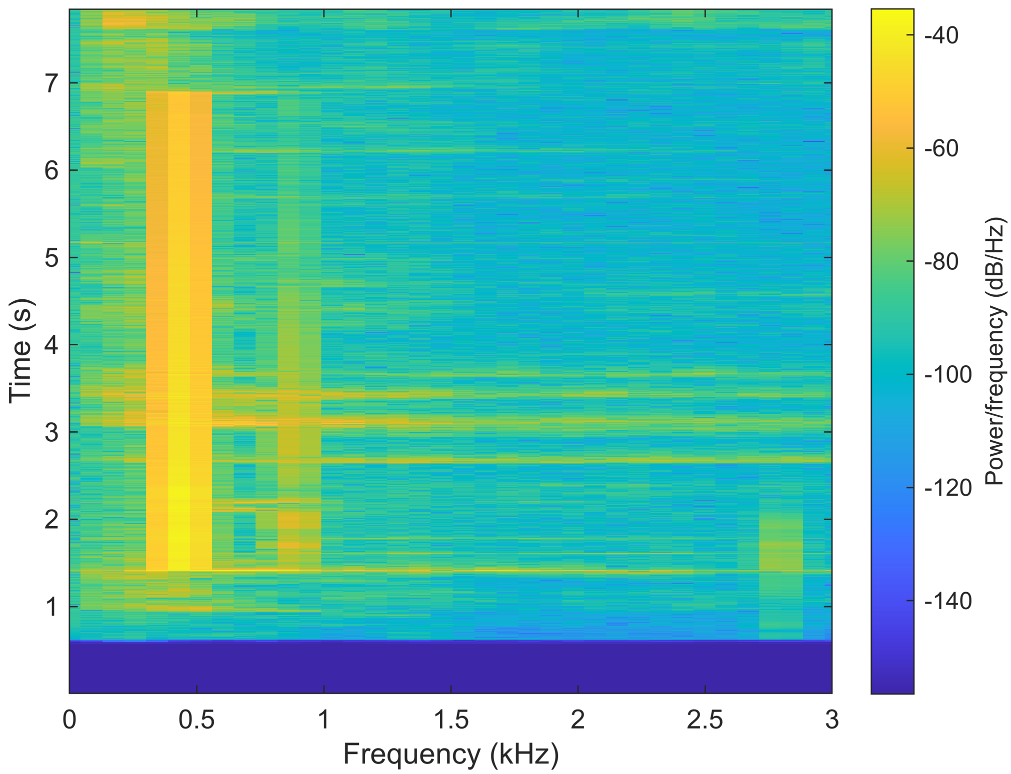

```
imDFT1 = fft(choosenSignal);
dft1 = abs(imDFT1);
stem(dft1)
title("Magnitudový spektralny graf")
xlabel("index of spectral coeficient")
ylabel("nenormalizovana amplitúda")
```




```
    %Prepočítajte index spektrálneho koeficienta na frekvenciu pre najväčsiu
hodnotu v grafe
     xStem = -fs/2 + 1:fs/512:fs/2;
     stem(xStem, fftshift(dft1))
     title("Frekvenčny graf")
     xlabel("Hz")
     ylabel("nenormalizovana amplitúda")
     xlim([-1 1000])
```




```
    %Cez funkciu spectrogram zobraz frekvenčné spektrum, celého raw signálu (a
uprataj magnitudove spektrum s hammingtonovym oknom)
     spectrogram(rawSignal(:,1), hamming(512), 256, 512,fs)%celý signal, okienko
pre elimináciu of spectral leakage, velkosti overlapu medzi jednotlivými okienkami
(dlžkami na vzorkovanie), dlžka vzorky do fft funkcie, rozsah frequency sampling
     xlim([0 3] %vysledok je graf 3-rozmerov, 1. frekvenčný rozsah, 2. časom
trvania nahravky, 3. silou danej frekvencie v danom čase
```



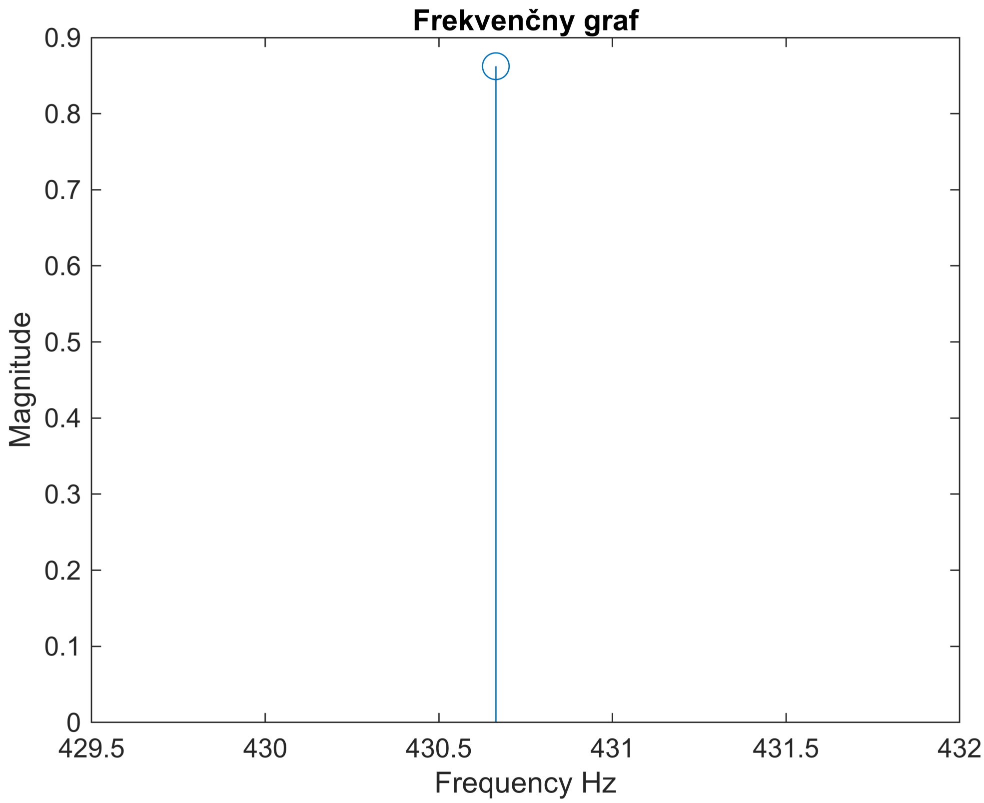

```
%Zistite dominantnú frekvenciu daného signálu a vyznačte ju v grafe.
[Max, indexOfMax] = max(dft1)
```

```
Max = 37.1393
indexOfMax = 6
```

```
foundFreq = (indexOfMax-1)/(512/2) * (fs/2);
display(foundFreq + "Hz")
```

```
"430.6641Hz"
```

```
stem(foundFreq, Max*(1024/fs), 'o','MarkerSize',10)
 title("Frekvenčny graf")
 xlabel("Frequency Hz")
 ylabel("Magnitude")
```




```
% xlim([-1 1000])
```

Úloha 2 - Zero Padding v spektrálnej doméne

Vygenerujte si náhodne zložený signál s dĺžkou 1 sekunda a s voľne zvolenou malou vzorkovacou frekvenciou. Zvolte si parametre signálu tak, aby najväčšia frekvencia generovaného signálu neprekročila 16Hz a vzorkovacia frekvencia pod 64Hz. Dbajte pri tom na to, aby bol splnený Nyquistov teorém.

Vykonajte zero padding signálu v spektrálnej doméne tak, že doplníte nulami spektrum od jeho stredu (stred symetrie musí ostať zachovaný). Doplnenie vykonajte na veľkosť vzorky 1024. Následne vykonajte spätnú transformáciu do časovej domény. Pôvodný signál, ako aj novo vytvorený spoločne vizualizujte v jednom grafe, kde x-os bude reprezentovať čas v rozsahu 1 sek.

Zodpovedajte nasledovné otázky:

- Akú výhodu nam priniesol zero padding v spektre?
- Vie nám zero padding v spektre pomôcť v prípade, ak bol porušený Nyquistov teorém v pôvodnom signále? (áno/nie)

Pomôcky:

- complex()
- zeros()
- ifft()

Poznámka:

- Spektrum signálu obsahuje magnitúdovú aj fázovú zložku. Pri dopĺňaní spektra nulami teda pracujeme s komplexnými číslami (0 + 0*i)
- Po spätnom prepočte inverznej transformacie prenásobte výsledný signál konštantou 1024/Fs pre normalizáciu hodnôt na veľkosť okienka

Riešenie:

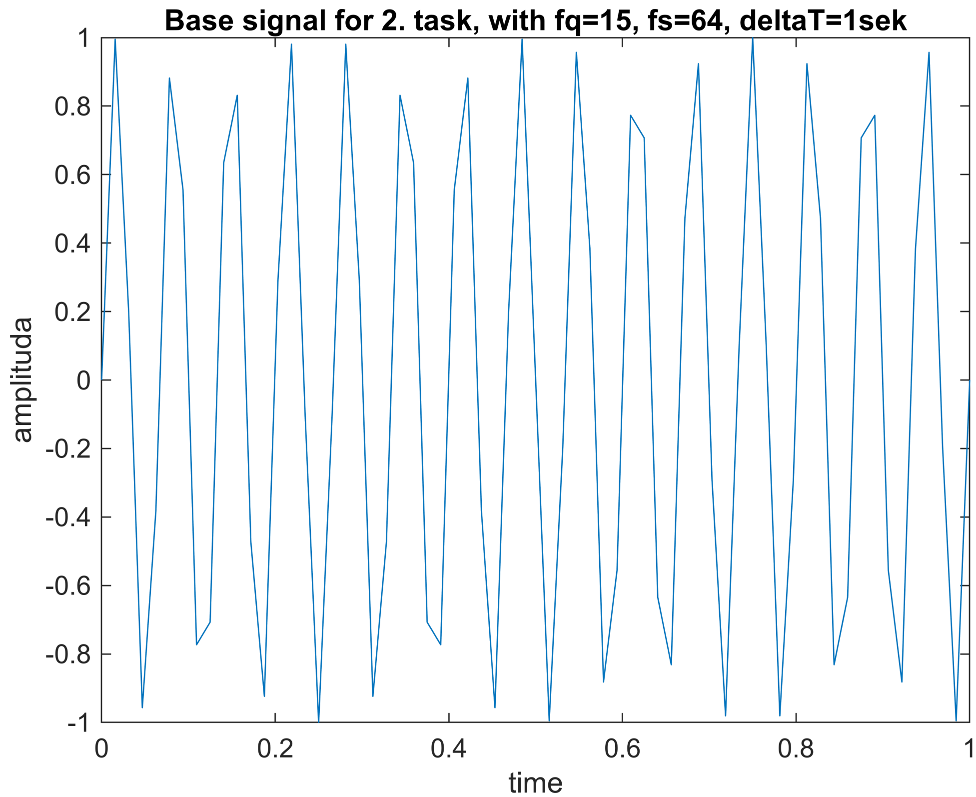
```
% 1.1 Vygenerujte si náhodne zložený signál s dĺžkou 1 sekunda a s voľne
% zvolenou malou vzorkovacou frekvenciou Zvolte si parametre signálu tak,
% aby najväčšia frekvencia generovaného signálu neprekročila 16Hz a
% vzorkovacia frekvencia pod 64Hz. Dbajte pri tom na to, aby bol splnený
% Nyquistov teorém. (stačí keď freq*2 > FS)
% čize fq < 16, fs = 64
fq2=15;
fs2=64
```

```
fs2 = 64
```

```
x_0 = 0:1/fs2:1
```

```
x_0 = 1×65
         0    0.0156    0.0312    0.0469    0.0625    0.0781    0.0938    0.1094 ···
```

```
signal = sin(2*pi*15*x_0);
plot(x_0, signal)
     title("Base signal for 2. task, with fq=15, fs=64, deltaT=1sek")
     xlabel("time")
     ylabel("amplituda")
```



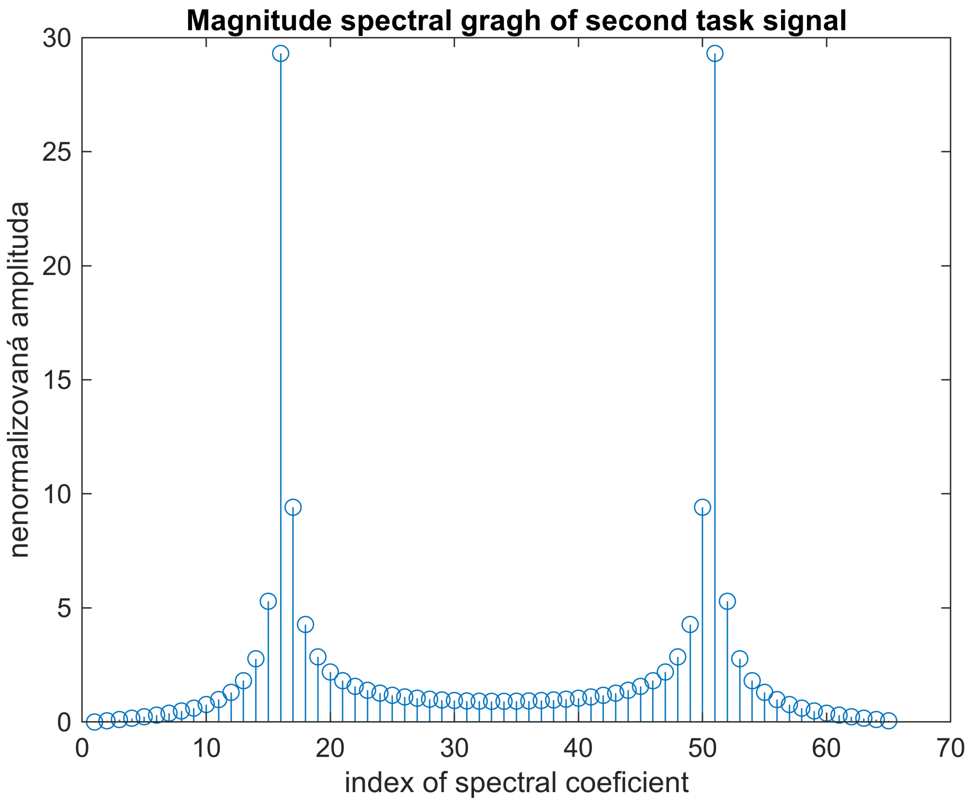

```
% Vykonajte zero padding signálu v spektrálnej doméne tak,
% že doplníte nulami spektrum od jeho stredu (stred symetrie musí ostať zachovaný).
% Doplnenie vykonajte na velkosť vzorky 1024. Následne vykonajte spätnú
transformáciu do časovej domény.
imDFT2 = fft(signal)
```

```
imDFT2 = 1×65 complex
  -0.0000 + 0.0000i   0.0026 - 0.0535i   0.0105 - 0.1082i   0.0242 - 0.1654i ···
```

```
dft2 = abs(imDFT2)
```

```
dft2 = 1×65
    0.0000    0.0536    0.1087    0.1672    0.2309    0.3025    0.3854    0.4846 ···
```

```
stem(dft2)
    title("Magnitude spectral gragh of second task signal")
    xlabel("index of spectral coeficient")
    ylabel("nenormalizovaná amplituda")
```




```
    [Max2, indexOfMax2] = max(dft2)
```

```
Max2 = 29.3048
indexOfMax2 = 16
```

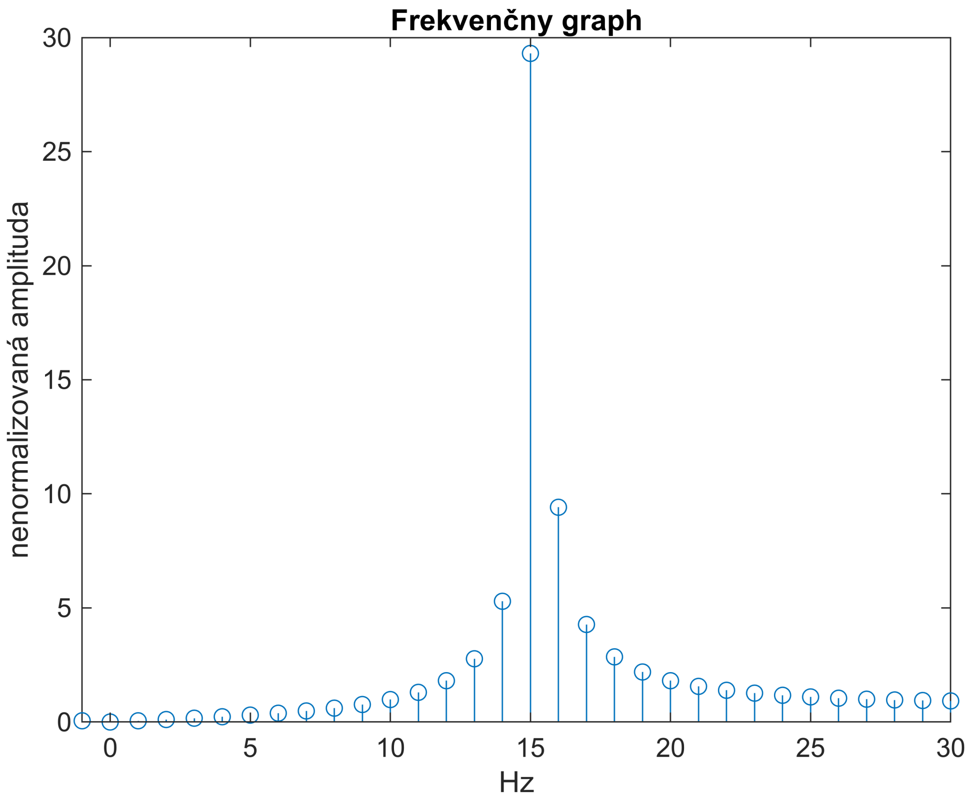
```
    foundFreq = (indexOfMax -1) / (512/2) * (fs2/2)
```

```
foundFreq = 0.6250
```

```
xStem2 = -fs2/2:fs2/2
```

```
xStem2 = 1×65
   -32   -31   -30   -29   -28   -27   -26   -25   -24   -23   -22   -21   -20 ···
```

```
stem(xStem2, fftshift(dft2))
    title("Frekvenčny graph")
    xlabel("Hz")
    ylabel("nenormalizovaná amplituda")
    xlim([-1 30])
```



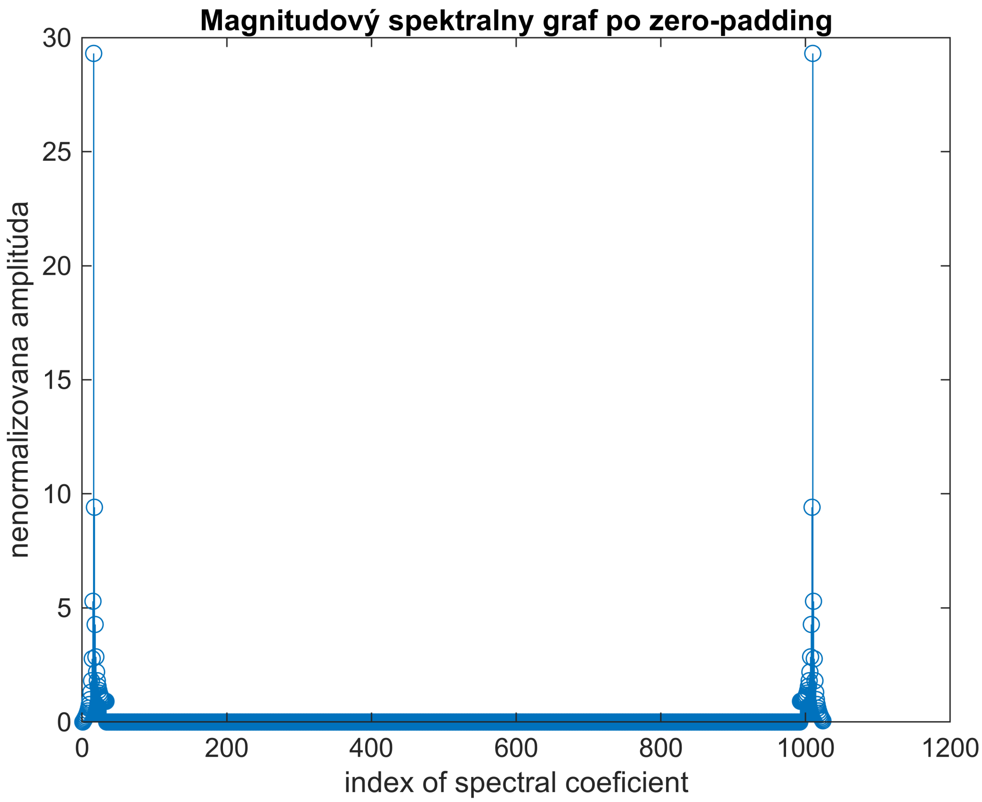

```
  %Doplnenie núl/zero padding
  zeroPaddign = zeros(1,1024-length(imDFT2));  %vytvorenie retazca potrebne dlzky pre doplnenie do aktualneho signalu
  complexZeroPadding = complex(zeroPaddign);   %naplnenie spomenuteho retazca nulovými compexnými hodnotami
  newImDFT2 = [imDFT2(1:33), complexZeroPadding, imDFT2(34:65)]  %finally zero-padding
```

```
newImDFT2 = 1×1024 complex
  -0.0000 + 0.0000i   0.0026 - 0.0535i   0.0105 - 0.1082i   0.0242 - 0.1654i ···
```

```
  stem(abs(newImDFT2))
     title("Magnitudový spektralny graf po zero-padding")
     xlabel("index of spectral coeficient")
     ylabel("nenormalizovana amplitúda")
% Pôvodný signál, ako aj novo vytvorený spoločne vizualizujte v jednom grafe,
% kde x-os bude reprezentovať čas v rozsahu 1 sek.
reSignal = ifft(newImDFT2)
```

```
reSignal = 1×1024
          0    0.0020    0.0045    0.0075    0.0111    0.0151    0.0195    0.0242 ···
```

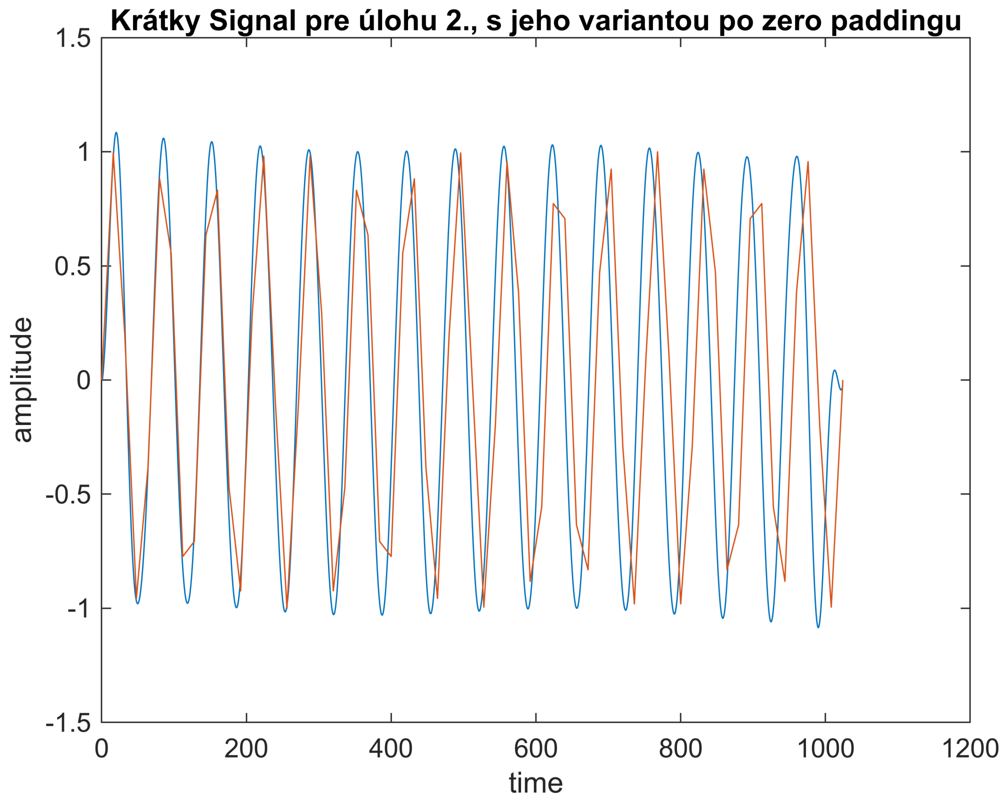
```
    %Akú výhodu nam priniesol zero padding v spektre?
hold off
```




```
plot(reSignal*(1024/fs2))%(1024/fs2) normalizácia amplitud po ich
hold on
x_1 = 0:1024/64:1024
```

```
x_1 = 1×65
     0    16    32    48    64    80    96   112   128   144   160   176   192 ···
```

```
plot(x_1,signal)
hold off
title("Krátky Signal pre úlohu 2., s jeho variantou po zero paddingu")
     xlabel("time")
     ylabel("amplitude")
```




```
%K predposlednej otázke, aký vlyv mal zero padding na spektralny graf.
    %Zlepsil rozlíšenie výsledneho signalu, po inverznej fourierovej transf.


%Vie nám zero padding v spektre pomôcť v prípade, ak bol porušený Nyquistov teorém
v pôvodnom signále? (áno/nie)
fs3 = 24
```

```
fs3 = 24
```

```
x_2 = 0:1/fs3:1
```

```
x_2 = 1×25
         0    0.0417    0.0833    0.1250    0.1667    0.2083    0.2500    0.2917 ···
```

```
signal2 = sin(2*pi*fq2*x_2);
imDFT3 = fft(signal2)
```

```
imDFT3 = 1×25 complex
  -0.0000 + 0.0000i  -0.0066 + 0.0525i  -0.0276 + 0.1076i  -0.0667 + 0.1685i ···
```

```
zeroPaddign1 = zeros(1,1024-length(imDFT3));
complexZeroPadding1 = complex(zeroPaddign1);
newImDFT3 = [imDFT3(1:13), complexZeroPadding1, imDFT3(14:25)]
```

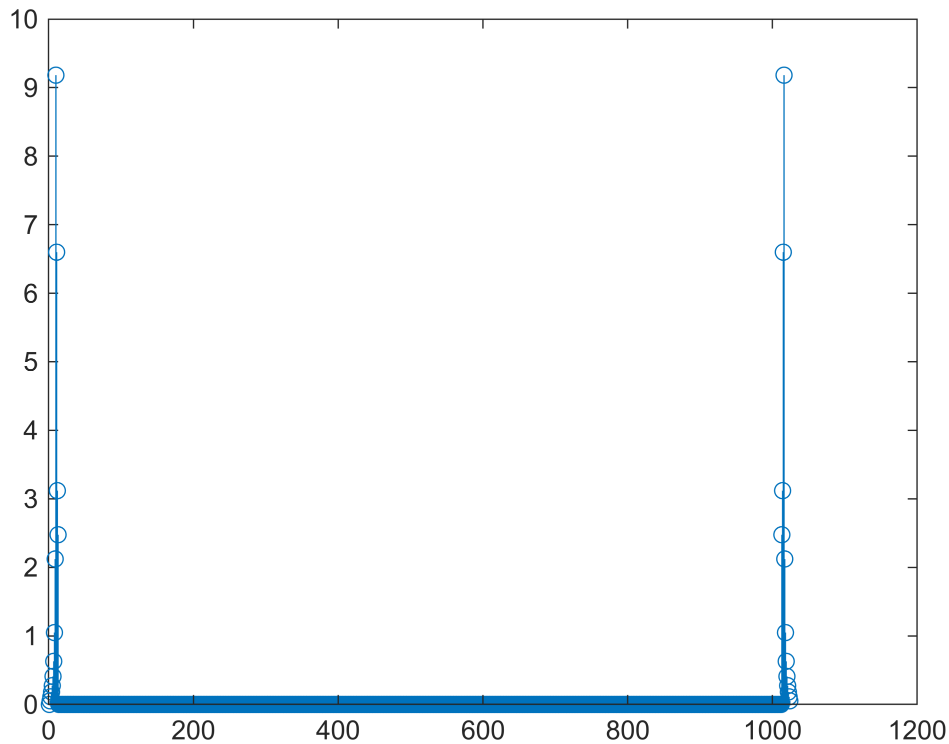
```
newImDFT3 = 1×1024 complex
  -0.0000 + 0.0000i  -0.0066 + 0.0525i  -0.0276 + 0.1076i  -0.0667 + 0.1685i ···
```

```
stem(abs(newImDFT3))
```

```
reSignal2 = ifft(newImDFT3)
```

```
reSignal2 = 1×1024
    0.0000   -0.0003   -0.0007   -0.0011   -0.0016   -0.0020   -0.0025   -0.0031 ...
```

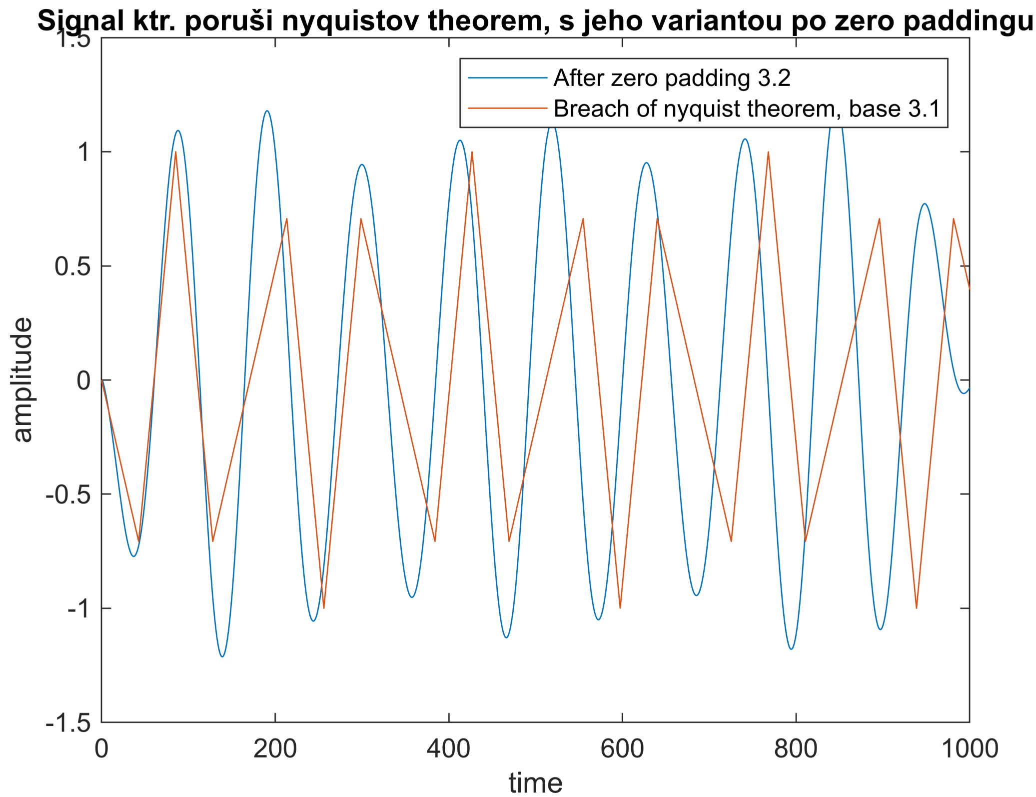
```
hold off
```




```
plot(reSignal2*(1024/fs3))%(1024/fs2) normalizácia amplitud po ich
hold on
x_3 = 0:1024/24:1024
```

```
x_3 = 1×25
10^3 ×
         0    0.0427    0.0853    0.1280    0.1707    0.2133    0.2560    0.2987 ...
```

```
plot(x_3,signal2)
hold off
xlim([0 1000])
legend('After zero padding 3.2','Breach of nyquist theorem, base 3.1')
     title("Signal ktr. poruši nyquistov theorem, s jeho variantou po zero
paddingu")
     xlabel("time")
     ylabel("amplitude")
```




```
%K poslednej otázke, vie nám zero padding pomôcť ak bol porušený nyquistov teorem
pri vzorkovaní.
    %Nie, zero padding v konečnom výsledku "otučnuje" signál, nevie kompenzovať
niečo čo tam chýba, alebo čo bolo skreslené porušením nyquistovho teoremu.
```