Part1:

```
Untitled4.m × +
data × ans ×
1x1 struct with 10 fields
                                                 clc
                                                 clear
                                          3
Filename
                  'C:\Users\ApeX-\Spe...
                                                 clear y Fs
🖟 CompressionMeth...
                 'Uncompressed'
                                                 [y,Fs] = audioread('Speech1.wav');
 NumChannels
                                                 [P,Q] = rat(8000/Fs);

→ SampleRate

                 8000
                                                 sound_new = resample(y,P,Q); %adjusting sampling rate to 8k
TotalSamples
                 82530
                                                 audiowrite('Speech1.wav', sound new, 8000)
 Duration
                 10.3163
                                                 sound(sound_new,Fs*P/Q);
Title
                                         10 -
                                                 audioinfo('Speech1.wav');
Comment
                 []
                                         11
Artist
                 [1
BitsPerSample
                 16
```

Part2:

```
Untitled4.m × Untitled8.m × +
       digits=find_digits(speech1);
3 -
       figure(1)
4 -
       plot(speech1(digits(:,1)))
6 -
        sound(speech1(digits(:,1)),8000)
7
     function intervals = find_digits(x);
8 -
           curPos = 1;
9 -
           \underline{L} = length(x);
10 -
           w = zeros(40,1);
11 -
           for (i=1:40)
12 -
               window = (x(curPos:curPos+4000-1));
13 -
                w(i) = (1/(4000)) * sum(abs(window.^2));
14 -
                curPos = curPos + 2000;
15 -
           end
16 -
           w(w<0.0011)=0;
17 -
           ii = zeros(size(w));
18 -
           jj = w > 0;
19 -
           ii(strfind([0,jj(:)'],[0 1])) = 1;
20 -
           idx = cumsum(ii).*jj;
21 -
           out = accumarray( idx(jj), w(jj)', [], (x) \{x'\}); Getting intervals seperated by 0
22 -
           for(i=1:10)
23 -
                w2=w==out{i};
                [row, col]=find(w2==1);
25 -
                Cp(i) mean(row)
26 -
                intervals (:,i) = [(Cp(i)-1)*2000:(Cp(i)+1)*2000-1]
27 -
```

We calculated energy of the signal in different intervals with some overlaping to increase accuracy, then we seperated each interval by zeros(silent intervals), then we specified the interval of each digit's sound.

Part2:

Figure: 1, digit: 0

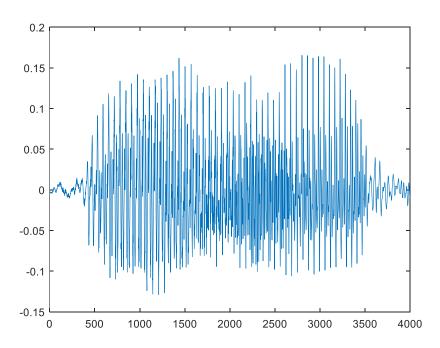


Figure2 digit: 1

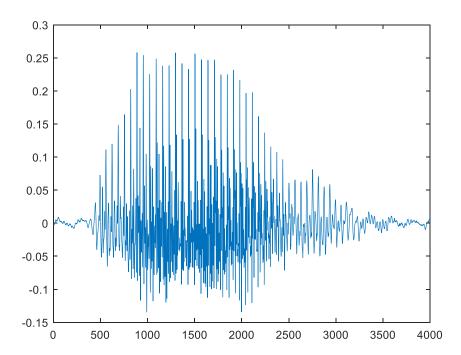


Figure 3 digit: 2

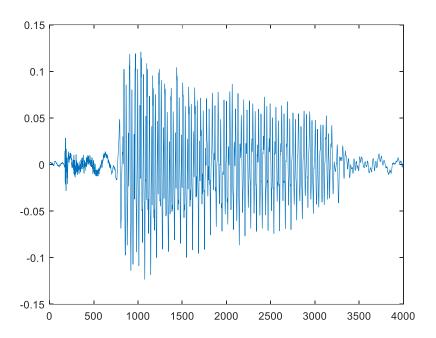


Figure: 4 digit:3

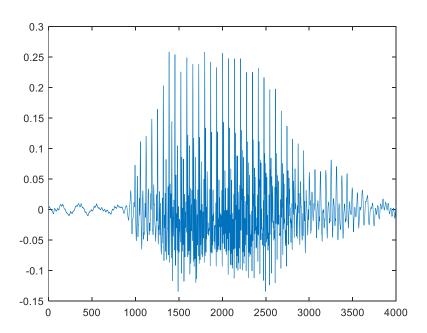


Figure 5 digit: 4

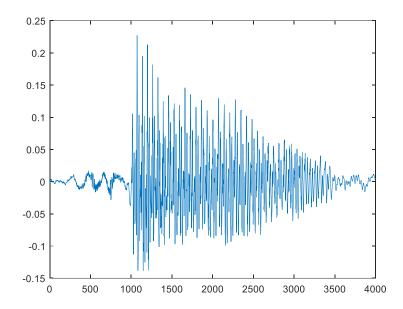


Figure 6 digit: 5

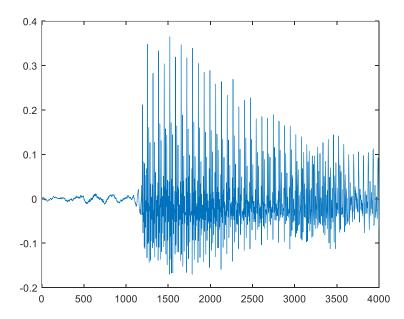


Figure 7 digit: 6

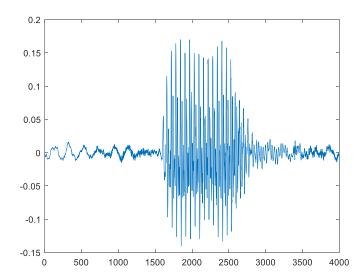


Figure 8 digit: 7

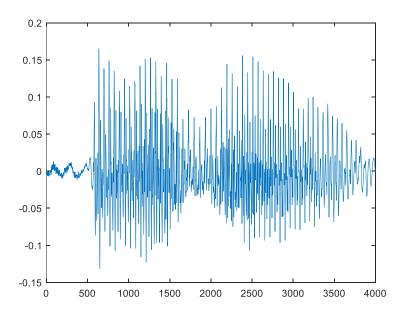


Figure: 9 sound: 8

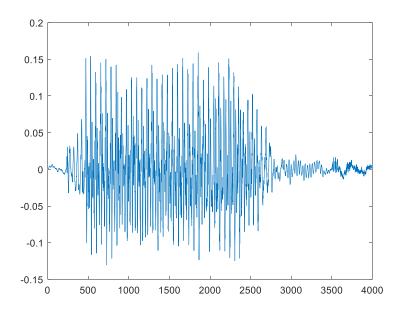
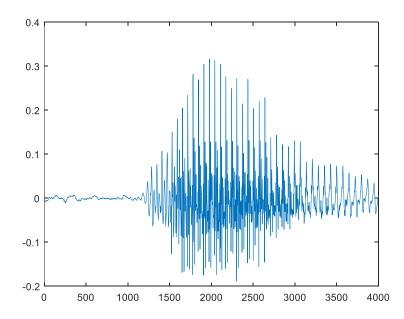


Figure10 digit: 9



Part3:

```
Untitled4.m × Untitled8.m × Part3.m × Untitled3.m × +
      N=810197644;
3 -
       speech2=int2speech(digits,N);
4 -
      noise=randn(length(speech2),1);
5 -
      E1=sum(abs(noise).^2);
      E2=sum(abs(speech2).^2);
7 -
      audiowrite('speech3.wav',speech2+(E2/E1/10)^(0.5)*noise,8000);
9 -
          speech=audioread('speech1.wav');
10 -
          m = floor(log10(N));
11 -
          D = mod(floor(N ./ 10 .^ (m:-1:0)), 10);
12 -
          y =zeros(length(D) *4000+(length(D)-1) *2000,1)
13 -
          curpos=1;
14 -
          for (i=1:length(D))
15 -
              y(curpos:curpos+4000-1)=speech(digits(:,D(i)+1));
16 -
              curpos=curpos+6000;
          end
17 -
18 -
          audiowrite('speech2.wav', y, 8000);
19 -
      L end
20
```

We specify a new vector corresponding to sounds of pronunciations of N digits with 0.25s pauses in between.

Then we add some noise with a coefficient so that the energy of signal is 10 times energy of the noise.

Part4:



Codes, and the result N is perfectly correct.

First we specify intervals of digits pronunciations, then we build up the correlation matrix using 2 'for' loops, after that we determine at what number there is max correlation between the input voice and the one we used as our reference. That's the digit pronounced in the input voice, then we concatenate the digits and it gives us the output.