Safa Keskin
150140137
BLG354E - HW2

NOTE1 : In order to run my code, you should have an “output” directory under the path that you are executing my code.

NOTE 2: My output files are accessible on dropbox:
https://www.dropbox.com/s/lr9o2ubv8v4q7jz/output.zip?dl=0

Q1) In first question, I have done research and found a code[1] that is written in python using numpy functions at https://stackoverflow.com/a/43964107/6013366. I thought that it is applicable to our problem and I used it implementing it into my function. In that source, Real Fast Fourier Transform is applied to signal and after that, it’s frequency is shifted by 100 Hz. After that, using python’s default wave library, output signal is written to the output file.

I have used one of the songs given by you by changing its name to “prodigy-omen.mp3”. I thought that program should be run with both mp3 and wav files. So, using pydub[1] library and AudioSegment class that is implemented in that library, I have changed its format to wav and saved as another file under “output” directory if the input file is in “mp3” format. Considering all use cases, I created a variable that holds “wav” file path. Then, I have read that file, manipulated with my function “readAndProcessWavUsingRftt“ and saved my answer as p1.wav.

All program can be run using command:
python q1.py “input_file.wav”
or
python q1.py “input_file.mp3”

Q2) In second question, I used the pitched up file that is generated with part 1 and speed up it by framing it with a higher frequency. That scaling is given as parameter to the system program and it’s fastened form is saved under same directory of input file with name “input_file_fastened.wav” (assumed that input file’s name is “input_file.wav”).

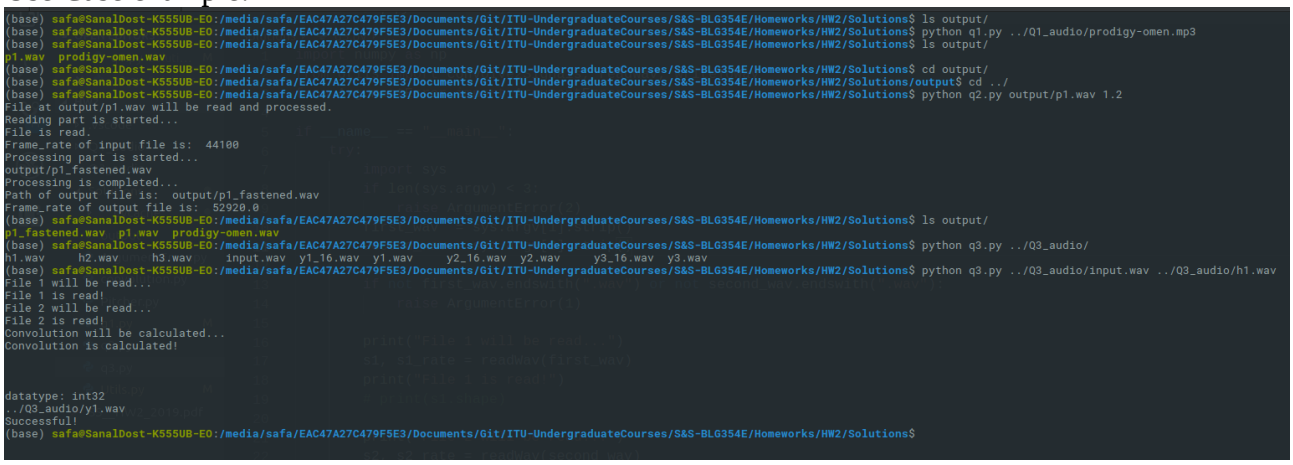
All program can be run using command:
python q2.py “input_file.wav” “scale_rate”

Q3) Using default convolution formula and numpy multiply function, I have implemented my convolution result and converted result to int32 format. Then, I have saved them under the same directory with names given in homework sheet.

All program can be run using command:
python q3.py “signal_1_path.wav” “signal_2_path.wav”

Example use cases are put as an image in page 3.

Use Case example:



Q4)

4-)

- $x[n] = u[n+10] - 2u[n] + u[n-4] = \begin{cases} 0 & n < -10 \\ 1 & , -10 \le n < 0 \\ -1 & , 0 \le n < 4 \\ 0 & , 4 \le n \end{cases}$ for $n \in \mathbb{N}$

- $h[n] = \cos\left(\frac{\pi}{3} n\right) = \begin{cases} 1 & , n \bmod 6 = 0 \\ 1/2 & , n \bmod 6 = 1 \\ -1/2 & , n \bmod 6 = 2 \\ -1 & , n \bmod 6 = 3 \\ -1/2 & , n \bmod 6 = 4 \\ 1/2 & , n \bmod 6 = 5 \end{cases}$

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[n-k] = h[n] * x[n]$$

	$n<-10$	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	3
$x[n]$	0	1	1	1	1	1	1	1	1	1	1	-1	-1	-1	-1	0
$h[n]$	changing	-1/2	-1	-1/2	1/2	1	1/2	-1/2	-1	-1/2	1/2	1	1/2	-1/2	-1	changing
$x[-10]\,h[n+10]$	changing 1/2	1	1/2	-1/2	-1	-1/2	1/2	1	1/2							
$x[-9]\,h[n+9]$		1/2	1	1/2	-1/2	-1	-1/2	1/2	1	1/2						
$x[-8]\,h[n+8]$		-1/2	1/2													
$x[-7]\,h[n+7]$		-1														
$x[-6]\,h[n+6]$		-1/2														
$x[-5]\,h[n+5]$		1/2														
$x[-4]\,h[n+4]$		1	1/2	-1/2	-1	-1/2	1/2	1	1/2	-1/2	-1	-1/2	1/2	1	1/2	
$x[-3]\,h[n+3]$		1/2	1	1/2	-1/2	-1	-1/2	1/2	1	1/2	-1/2	-1	-1/2	1/2	1	
$x[-2]\,h[n+2]$		-1/2	1/2	1	1/2	-1/2	-1	-1/2	1/2	1	1/2	-1/2	-1	-1/2	1/2	
$x[-1]\,h[n+1]$		-1	-1/2	1/2	1	1/2	-1/2	-1	-1/2	1/2	1	1/2	-1/2	-1	-1/2	
$x[0]\,h[n]$		1/2	1	1/2	-1/2	-1	-1/2	1/2	1	1/2	-1/2	-1	-1/2	1/2	1	
$x[1]\,h[n-1]$		-1/2	1/2	1	1/2	-1/2	-1	-1/2	1/2	1	1/2	-1/2	-1	-1/2	1/2	
$x[2]\,h[n-2]$		-1	-1/2	1/2	1	1/2	-1/2	-1	-1/2	1/2	1	1/2	-1/2	-1	-1/2	
$x[3]\,h[n-3]$		-1/2	-1	-1/2	1/2	1	1/2	-1/2	-1	-1/2	1/2	1	1/2	-1/2	-1	
$h[n]$		-3/2	3/2	3/2	3/2	-3/2	-6/2	-3/2	3/2	6/2	3/2	-3/2	-6/2	-3/2	3/2	

$h[n] = 3 \cdot \cos\left(\frac{\pi}{3} n\right)$ where $-10 \leq n < 4$

$h[n] = \cos\left(\frac{\pi}{3} n\right)$ where $n < -10 \vee n \geq 4$

b) $h[-8] = -1$ $h[-7] = -1$ $h[-6] = 0$ $h[-5] = 2$ $h[-4] = 5$ $h[-3] = 7$

$h[-2] = 8$ $h[-1] = 9$ $h[0] = 8$ $h[1] = 7$ $h[2] = 5$ $h[3] = 2$

$h[4] = -1$ $h[5] = -2$ $h[6] = -1$ $h[7] = 2$ $h[8] = 5$ $h[9] = 7$

$h[10] = 8$ $h[11] = 9$ $h[12] = 8$ $h[13] = 7$ $h[14] = 5$ $h[15] = 2$

$h[16] = 0$ $h[17] = -1$ $h[18] = -1$ $h[19] = 0$ ...

Calculated observing plots. Second signal ($h[n]$) is inversed and moved over $x[n]$.

References:

1) https://stackoverflow.com/a/43964107/6013366
2) http://pydub.com/