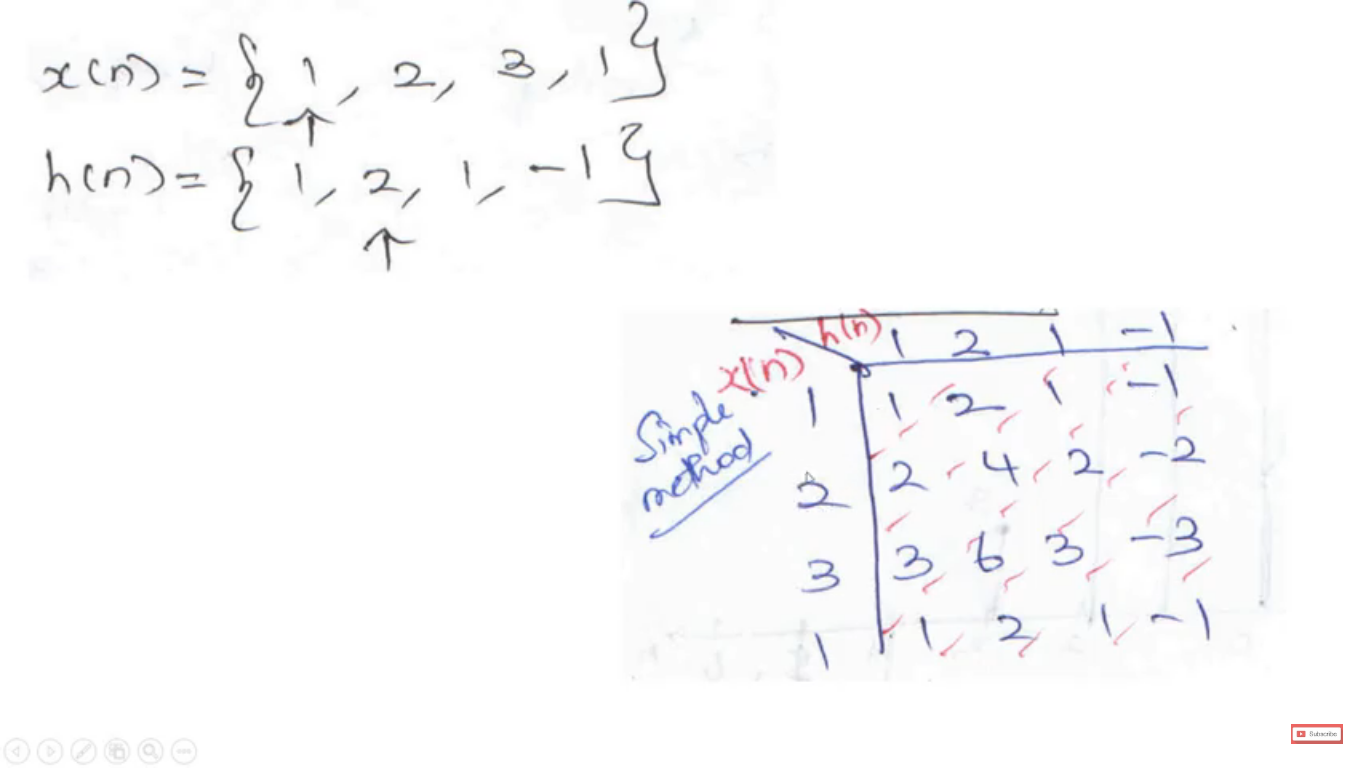
**سوال اول:** تابع با توجه به خواسته سوال در فایل p1.m نوشته شده است. برای طراحی این تابع مشکل خواصی وجود نداشت. عملکرد این تابع تا استفاده از دستور det مقایسه شد و در تمام آزمون ها تابع بع درستی کار میکرد.

در نوشن کامنت هایی به ابتدای تابع اضافه شد که به ترتیب عملکرد تابع، فرمول ریاضی مورد استفاده و نوع ورودی و خروجی ها مشخص شده است که درک تابع را راحت تر کند.

**سوال دو :** برای این حل این سوال هم مشکل خاصی وجود نداشت فقط فرمول ریاضی معکوس ماتریس 2 در 2 به زبان متلب تبدیل کردم. با توجه به خواسته صورت سوال برای ماتریس های که معکوس پذیر نیستند متنی نمایش داده می شود که کاربر از معکوس ناپذیر بود ماتریس آگاه شود.

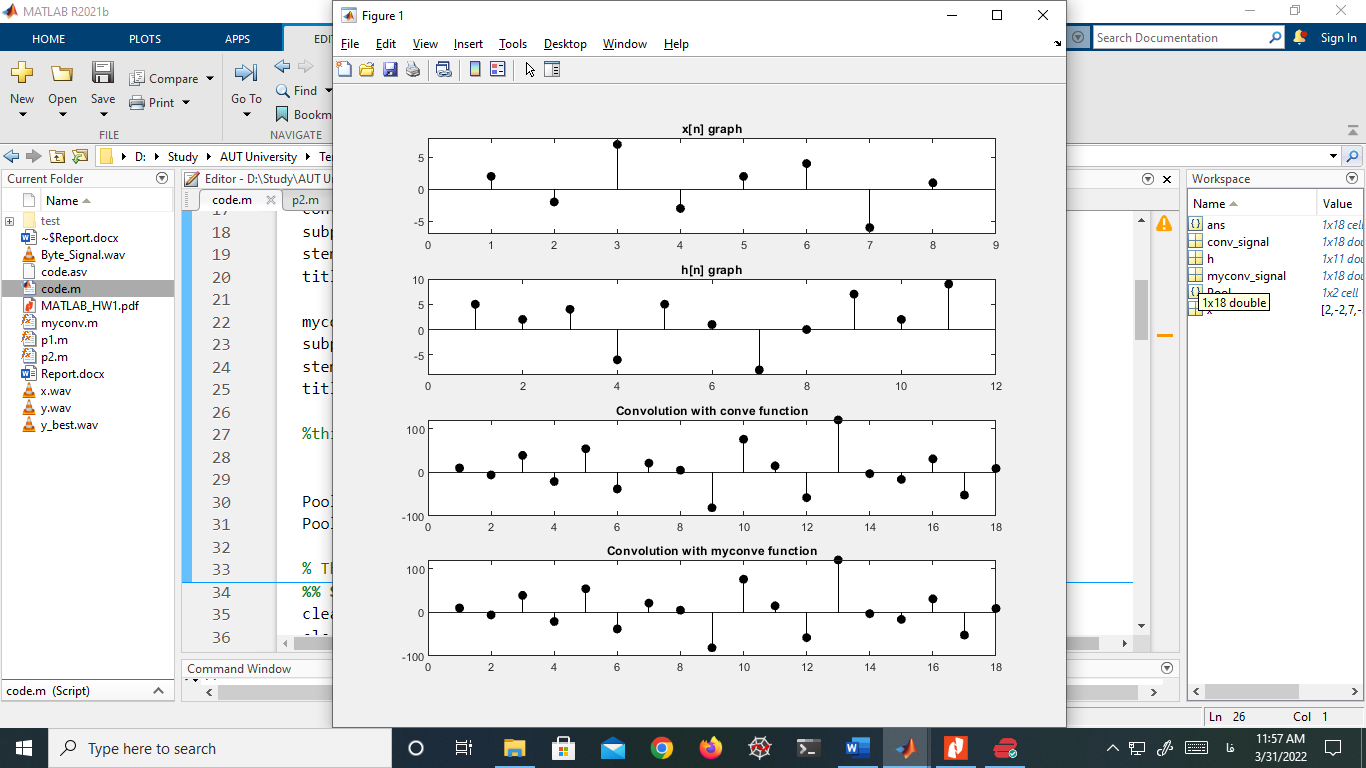
**سوال سه :** برای این سوال یک تابع کانولوشن نام myconv طراحی کردم که برای طراحی از روش بدست آوردن کانولوشن گسسته ماتریسی استفاده کردم زیرا نتوانستم به روش های موجود در کتاب برای بدست آوردن کانولوشن گسسته این تابع را طراحی کنم.

در این روش بردار x در سطر و بردار h را در ستون ها قرار میدهیم ماتریس بالا را شکل می دهیم و برای بدست آوردن کانولوشن اعداد قطر های فرعی ماتریس را باهم ضرب میکنیم و نتیجه حاصل را باهم جمع می کنیم.

در خط 8 تا 12 ماتریس بالا پیاده سازی شده است. در بقیه کد فرمول بمحاسبه کانولوشن با ماتریس بالا پیاده سازی شده است.

برسی عملکرد تابع myconv که من طراحی کردم نسبت به تابع conv که مال متلب است در فایل code.m انجام شده است.

* فایل code.m چهار بخش دارد که به ترتیب مربوط به سوالات 3 و 4.1 و 4.3 و 4.4 می باشد. **برای اینکه کد به خوبی اجرا شود باید برای ران کردن کد از گزینه Run Section استفاده شود** زیرا این بخش ها به صورت مجزا از هم کار میکنند.

در بخش سوال 3 با توجه به خواسته سوال ابتدا بردار های x[n] و h[n]و conv و myconv رسم شده اند. برای رسم آنها از تابع stem استفاده شده است که مخصوص نمایش این نوع بردار ها است. از subplot استفاده شده است تا تمام نمودار ها در یک پنجره نمایش داده شوند. و برای اینکه نمودار ها زیبا تر شوند از xlim و ylim استفاده شده است.

برای استفاده از تابع subplot از داکیومنت متلب استفاده شده است : <https://www.mathworks.com/help/matlab/ref/subplot.html>

برا استفاد از تابع stem از داکیومنت متلب استفاده شده است:

<https://www.mathworks.com/help/matlab/ref/stem.html>

من رو بدست آوردن کانولوشن به روش ماتریس را از فیلم زیر یادگرفتم:

<https://youtu.be/nzVPoc4PRgI>

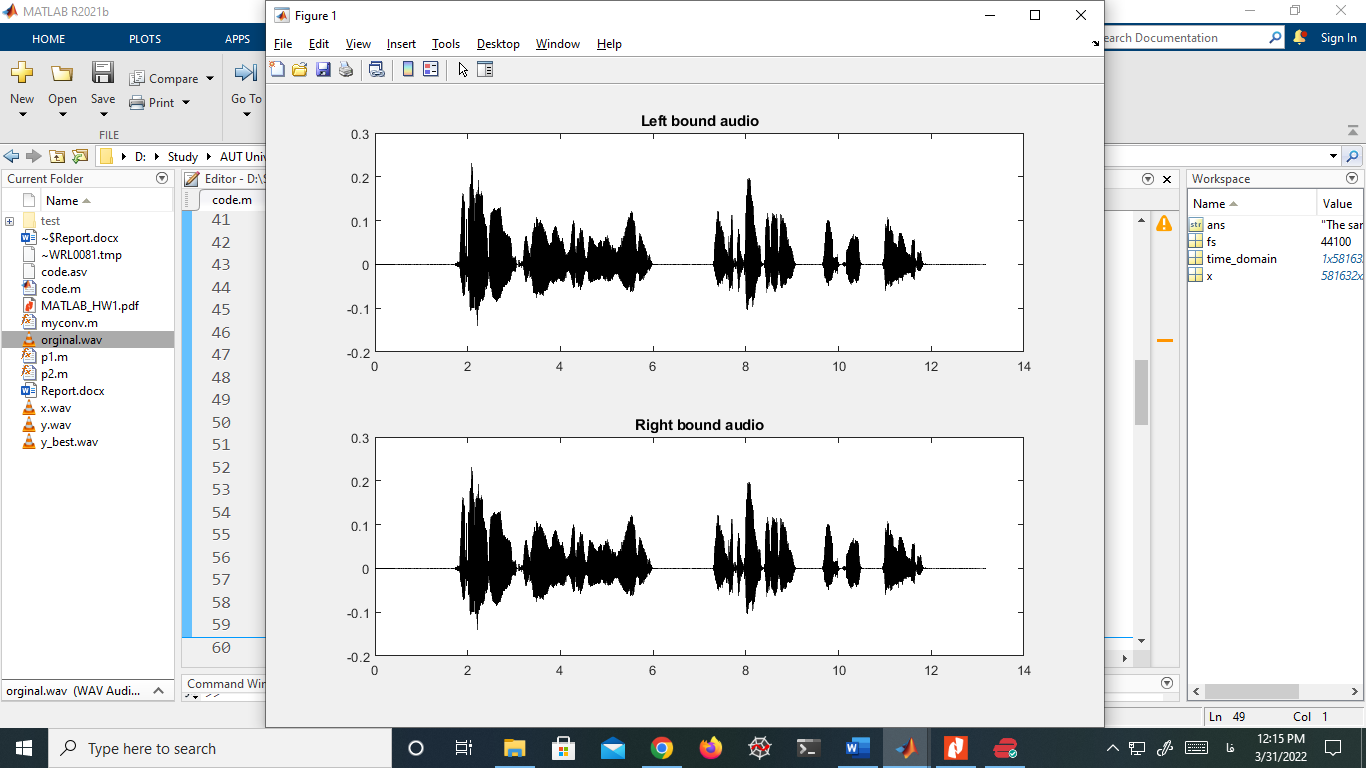
در انتهای کد بخش سوال 3.**تمام مقادیر بردار های** conv و myconv مقایسه شده که آیا باهم یکسان هستند یا خیر. در متلب از صفر و 1 برای نمایش ارزش گزاره استفاده می شود. من برای تبدیل 0 و 1 به true و false از سایت زیر کمک گرفتم.

<https://www.mathworks.com/matlabcentral/answers/1449214-converting-1-and-2-to-false-and-true>

بعد از ران گرفتن از کد متوجه میشویم که تابع myconv تمام مقادیر را مانند conv محاسبه کرد است و عملکرد خوبی دارد.

**سوال 1\_4 :**

یک بیت از حافظ شیرازی خوانده شده و به فرمت wav در دایرکتوری فعلی ذخیره شده است. اسم فایل orginal.wav است. با توجه خواسته صورت سوال برای اینکه محور افقی بر حسب ثانیه باشد از کد خط 50 استفاده شده است. این کد فاصله بین هر ثانیه را به fs بخش مساوی تقسیم میکند که بتوان نمودار را بر حسب زمان نوشت. این کد از کلیپ زیر الهام گرفته شده است.

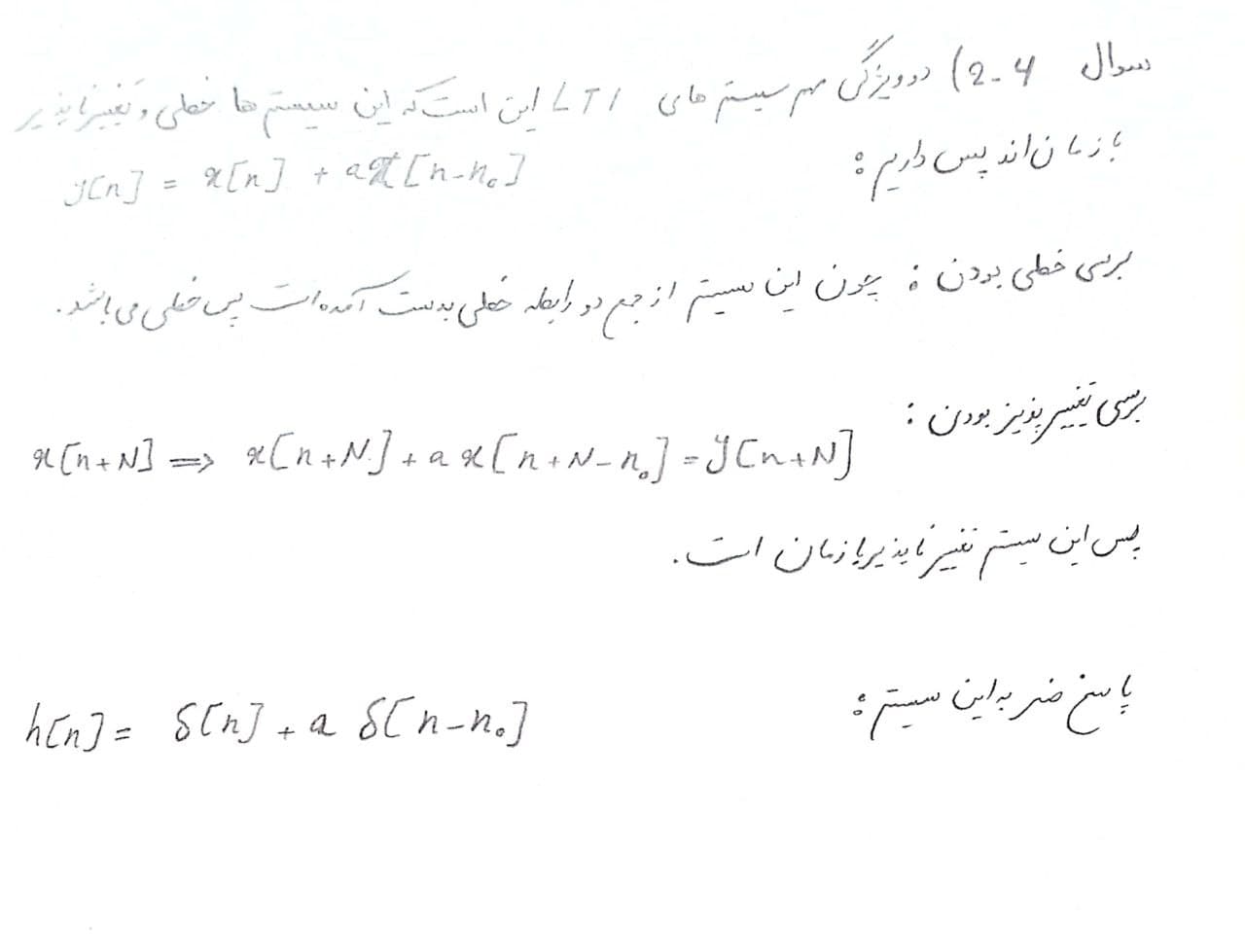
<https://youtu.be/UGChHxnuLSE>

همین طور مشخص است فایل صوتی یکبار بخش شده و در به صورت فایل x.wav ذخیره می شود.

و در نهایت هر دو بعد(باند) صوت رسم شده است و در یک پنجره نمایش داده میشود.

من کار با سیگنال صوتی را بلد نبودم و آن را توسط فیلم سری فیلم های ارسالی TA یاد گرفتم:

<https://www.aparat.com/v/1rAj0>

**سوال 2\_4:**

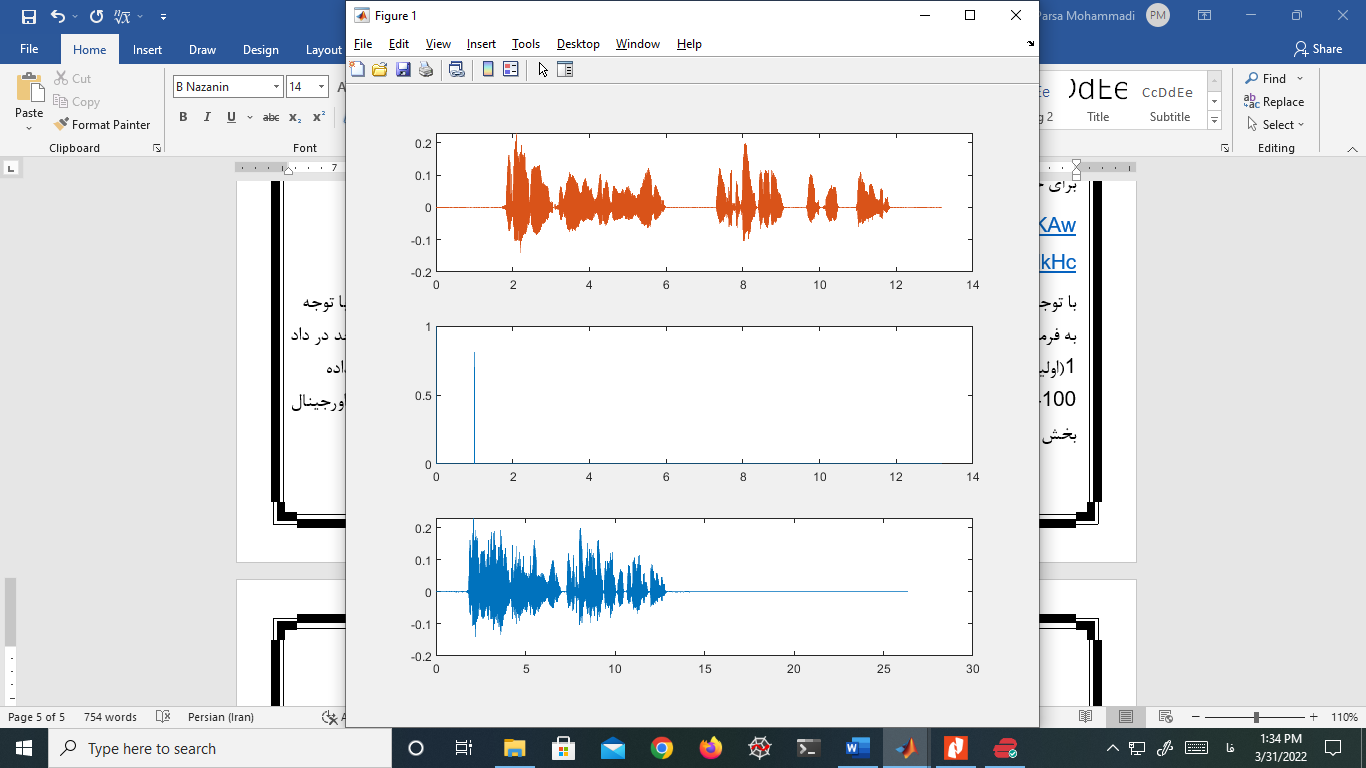
**سوال 4\_3:**

برای حل این سوال از ویدیو های زیر دیده شد.

<https://youtu.be/vmiRcJNJKAw>

<https://youtu.be/2UoECMpMkHc>

با توجه به فرمول پاسخ ضربه در سوال قبل پاسخ ضربه برای تولید صدای اکو در خط های 74 تا 76 طراحی میکنیم. با توجه به فرمول ابتدا باید خودسیگنال اصلی اجرا شود و بعد از آن صدای الکو بخش شود پس به این خاطر ابتدا یک ضربه واحد در داد 1(اولین داده صوت) قرار می دهیم تا بعد از کانوالو ابتدا صدای اصلی اجرا شود و بعد در یک ضربه به اندازه 0.81 در داده 44100 قرار میدهیم زیرا 1\*fs برابر 44100 میشود.در این صورت بعد از کانوالو صدای اکو یک ثانیه بعد از صدای اورجینال بخش میشود و قدرت آن 81 درصد صدای اصلی است.

به ترتیب از بالا به پایین نمودار های سیگنال اورجینال و پاسخ ضربه و سیگنال اکو رسم شده است.

در نهایت سیگنال حاصل کانوالو x و h در فایل y.wav ذخیره شده است.

**سوال 4-4 :**

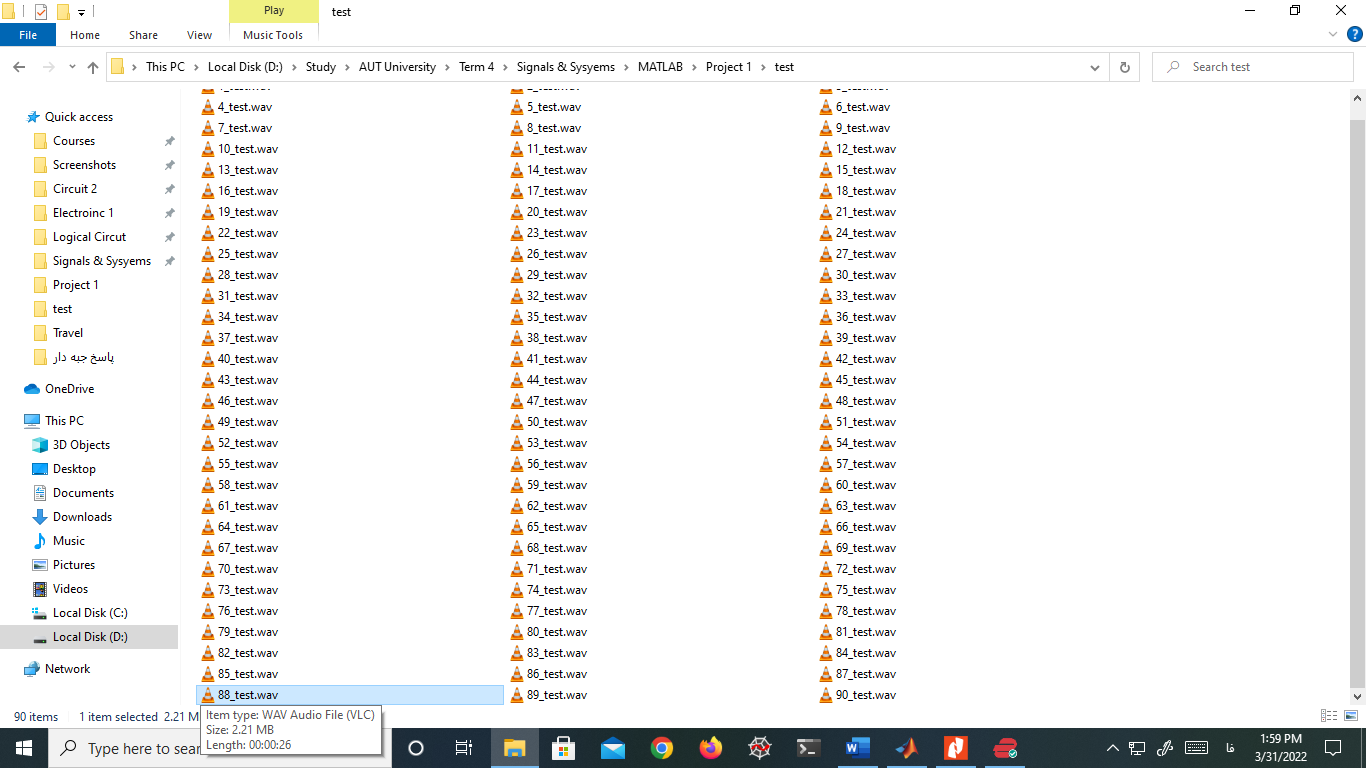
برای اینکه به این سوال به بهترین حالت ممکن پاسخ داده باشم دو حلقه درون هم طراحی کردم تا تمام حالت های ممکن برای زمان تاخیر اکو و قدرت آن برسی کنم.

ابتدا سیگنال x که همان سیگنال اورجینال ما است را لود کردم و با استفاده کد خط 97 دو بعد سیگنال را به یک بعد تبدیل کردم. با این کد دیگر در تبدیل 2 باند یه تک باند دادای از دست نمیرود.

پایین تر به حلقه ها میرسیم که حلقه اول(خارجی) زمان های 2 تا 10 ثانیه را تست میکند و حلقه داخلی قدرت سیگنال اکو را از 0.1 تا 2 مورد برسی قرار میدهد. در مجموع 90 صوت متفاوت خواهیم داشت که زمان تاخیر اکو و قدرت آن در تمام آنها متفاوت است. این فایل ها در پوشه test در همین دایرکتوری ذخیره میشوند و نام آنها از 1\_test تا 90\_test متفاوت.

من به تمام این صوت های ساخته شده گوش کردم و بهترین آنها صوت شماره 12 بوده و به آن را به استم y\_best.wav در همین دایرکتوری ذخیره کردم.

این بخش کاملا خلاقانه بوده و از منبعی استفاده نکردم.



این تصویر تعدادی از فایل های ساخته شده را نمایش می دهد.

به علت حجم بالا و تعداد بالا این صوت ها امکان آپلود آن ها نمیباشد.

**سوال 5\_4:**

به نظر من بهترین روش برای یافتن بهترین پارامتر ها استفاده از الکوریتم های هوش مصنوعی و یادگیری ماشین است. برای مثال ما صوت های مختلف با پارامتر های اکو مختلف را جمع آوری میکنیم و به آنها گوش می دهیم و آنهایی که از نظر ما گوش نوازند را انتخاب میکنیم. دادهای صوت های منتخب را به الگوریتم های curve fitting مانند ژنتیک یا دیگر الگوریتم ها میدهیم. بعد از یافتن بهترین مقادیر a و n0 دیگر میتوانیم بهترین اکو ممکن هر صوت را بیابیم.

این سوال با استفاده از دانش خودم در مورد علوم داده و پردازش سیگنال پاسخ داده شد و از منبعی استفاده نشده است.