Paula Osés Noguero
NIA: 207310

P2: MORE FFMPEG

In this Lab we had to resolve 4 exercises using python and ffmpeg. For this I have created a Python project using PyCharm where you have a menu and you can select the different options for each exercise.
You can see here the image of the menu,

```
1.Create a script which can cut N seconds the BBB video.
2.Create a script which will extract the YUV histogram from the previous BBB video you've done and create a new video with both images at the same time.
3.Create a script which can resize the BBB video into 4 differents video outputs (doesn't need to be at the same time)
    a) 720p
    b) 480p
    c) 360x240
    d) 160x120
4.Create a script which can change the audio into mono output and in a different audio codec
    a)Change to mono audio
    b)Change to mp3
5.Exit/Quit --> Press enter
```

Exercise 1:
For the first exercise we were asked to create a script which can cut N seconds from the BBB video.
For this I have used the command, where "sec" is the variable which stores the user's input to choose how many seconds of the video you want.

```
ffmpeg -ss 00:00:00 -i bbb.mp4 -to' sec -c copy output.mp4'
```

The result of this command is a video of 12 seconds named output.mp4 that is stored in the same folder as the program, you can see the duration of 12s in the following image.

Exercise 2:
In this exercise we were asked to create a script which will extract the YUV histogram from the previous BBB video you've done and create a new video with both images at the same time. Running this command we create a video that shows the histogram of each frame.

```
ffplay bbb.mp4 -vf
'split=2[a][b],[b]histogram,format=yuva444p[hh],[a][hh]overlay'
```

You can see here the capture of the video where we have the video playing and it shows the histogram.

Paula Osés Noguero
NIA: 207310

Exercise 3:
Now we use FFMPEG to resize the BBB video into 4 different video outputs, 720p, 480p, 360x240, 160x120. To do so I have created a menu where you select which size do you want and then use the following command to save the video with the chosen size.

```
ffmpeg -i output.mp4 -s 1280x720 -c:a copy 1280x720.mp4
```

Exercise 4:
For this last exercise we have to create a script which can change the audio into mono output and in a different audio codec. For this I have created a menu with these two options. For the first one I have used this command which returns the video with mono audio.

```
ffmpeg -i output.mp4 -ac 1 mono.mp4
```

And then for the second option I convert the audio of the video to format acc using,

```
ffmpeg -i output.mp4 -acodec aac -vcodec copy audio_aac.mp4
```

With this command we read the video audio, transform it to a different codec, in this case aac, and then encode the video with the new audio.

Finally, for all the exercises I have used the video that we cut in exercise 1, so it doesn't take so much time.