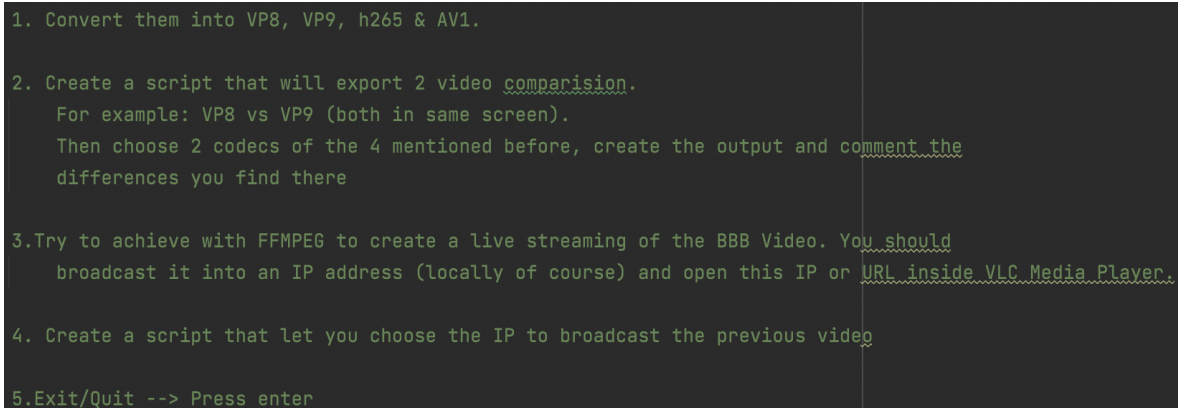


S3: FINAL EXERCISES

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 main() where you find the menu. Also a file called s3.py where there is a class called seminar3 and inside there are the functions for each exercise.

You can see here the image of the menu,



```
1. Convert them into VP8, VP9, h265 & AV1.

2. Create a script that will export 2 video comparision.
   For example: VP8 vs VP9 (both in same screen).
   Then choose 2 codecs of the 4 mentioned before, create the output and comment the
   differences you find there

3. Try to achieve with FFMPEG to create a live streaming of the BBB Video. You should
   broadcast it into an IP address (locally of course) and open this IP or URL inside VLC Media Player.

4. Create a script that let you choose the IP to broadcast the previous video

5. Exit/Quit --> Press enter
```

Exercise 1:

For the first exercise we were asked to create a script where you can select the size of the video and then convert it into VP8, VP9, h265 & AV1. For this I have created a menu where you can select the size and the type of codec you want. To change the type of codec I used the following command,

```
ffmpeg -i resized.mp4 -c:v libvpx -b:v 1M -c:a libvorbis vp8.webm
```

Exercise 2:

Now we are asked to create a script that will export 2 video comparison, for this i will use the video that I have converted previously (vp8.webm and vp9.webm) and then I use this command,

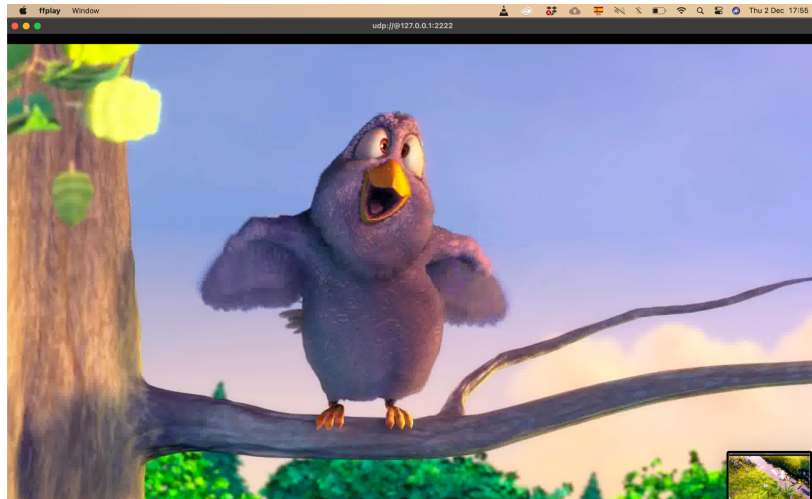
```
ffmpeg -i vp8.webm -i vp9.webm -filter_complex [0][1]vstack
compare_video.mp4
```

Between both video I cannot see much difference but i do know that VP9 video should have more quality than the VP8.

Exercise 3:

In this exercises we are asked to create a live streaming of the bbb.mp4 video, for that I use the 1 minute version of this video so that it is not very heavy to send and then I use the following command,

```
ffmpeg -i 1min.mp4 -f mpegts udp://@127.0.0.1:2222
```



Exercise 4:

Finally I have implemented a variable so you can choose the IP where you want the stream and the i pass it to the previous command.

```
def ex4(self):  
    inp = input('Enter the IP to broadcast the video')  
    ip = 'udp://@' + str(inp)  
    subprocess.call(['ffmpeg', '-i', '1min.mp4', '-f', 'mpegts',  
                    ip])
```