

Assignment 1

- **Note.** Create separate google colab answer files for both the questions uploading them in a single folder on your drive and share the link of this folder in the submission. Valid submissions to both questions are compulsory for the assignment to be considered for evaluation.

Questions

1. In the science fiction story "*Now Inhale*", by Eric Frank Russell, a human is held prisoner on a planet where the local custom is to make the prisoner play a game until it is won or lost before his execution. The protagonist knows that a rescue ship might take a year or more to arrive, so he chooses to play Towers of Hanoi with 64 disks. This story makes reference to the legend about the Buddhist monks playing the game until the end of the world.

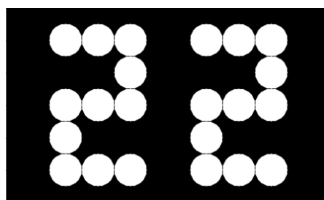
The **Tower of Hanoi** (also called The problem of Benares Temple or Tower of Brahma or Lucas' Tower) is a mathematical puzzle consisting of three rods and a number of disks of various diameters, which can slide onto any rod. The puzzle begins with the disks stacked on one rod in order of decreasing size, the smallest at the top, thus approximating a conical shape. The objective of the puzzle is to move the entire stack to one of the other rods, obeying the following rules:

- Only one disk may be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and
- placing it on top of another stack or on an empty rod.
- No disk may be placed on top of a disk that is smaller than it.

Now your **task** is to write a python code that calculates how many steps it will take to solve the Tower of Hanoi for a given input of number of disks and then display the result on a custom width Dot Matrix Display :-

Dot Matrix displays are used in LED sea markers or places where we need to put numbers for people's attention. There are some rules to it though. Each dot in the dotmatrix should be a circular disc of white or black colour with a radius of 25 pixels. Height of the dot matrix image should be 300px and width according to the input. It should be grayscale (unsigned int) having any pixel value to be either 0 or 255.

Bonus : Try to solve for Cyclic Hanoi as well :)



2. If you punch your hostel room's wall in distress or in over-excitement, you might create a window there. That would be a very high-quality punch; on the other side, there are poor-quality bricks lying. There are four qualities of bricks used in the construction, having different costs according to their quality. Sounds interesting? So they do, via making different sounds when struck together. We will consider only 2 quality bricks here - low and high. Properly cooked red soil bricks are of high quality. The furnace melts ferrites and imparts a metal-like quality to these bricks. Therefore, when high-quality bricks are struck together, they produce a sound resembling that of metal, while low-quality bricks sound like cardboard banged together. You will be provided with audio (.mp3 files) of these brick sounds. You can analyze it in the image domain via audio-to-spectrogram conversion using windowed Fourier transforms. You may use various python libraries for getting spectrogram, like the below pseudo-code:

```
import librosa
# Load audio file
y, sr = librosa.load(audio_path, sr=None)
# sr=None preserves sampling rate

# Set parameters for spectrogram computation
n_fft = 2048 # FFT points
hop_length = 512 # Sliding amount for windowed FFT
# Compute mel spectrogram
spec = librosa.feature.melspectrogram(y, sr, n_fft=n_fft,
                                     hop_length=hop_length, fmax=22000)
# Perform power to decibel (dB) image transformation
spec_db = librosa.power_to_db(spec)
# Display spectrogram images before designing your algorithm
```

Write a python program, which can process the brick sounds and evaluates their quality in terms of metal (high) or cardboard (low) quality. Maintain a sum initialized to 0 with metal adding 1 and cardboard adding 2 to it. Display this sum on the previously made dot matrix display but this time in a creative manner using RGB values for different pixels by defining a custom function taking pixel co-ordinates as parameters. You will be provided with 4 test cases and submission code will be tested on total 20 test cases.

Good Luck!!