|  |  |
| --- | --- |
| Name: | Bodhisatya Ghosh |
| UID: | 2021700026 |
| Class: | CSE DS |
| Batch: | B |
| Experiment: | 5 |

**Aim:**  To perform filtering of Long Data Sequence using Overlap Add Method and Overlap Save Method.

**Objective:**

To Develop a function to implement Fast Overlap Add Algorithm and Overlap Save Algorithm.

**Theory:**

Overlap Add Method (OAM) and Overlap Save Method (OSM) are two widely used techniques for efficient implementation of linear convolution using Fast Fourier Transform (FFT) in digital signal processing. Both methods are used to break down long convolutions into smaller convolutions, making the computation more manageable and faster. Here's an explanation of each method along with their pros and cons:

**Overlap Add Method (OAM):**

Method: - In OAM, the input sequence is divided into blocks, and each block is convolved with the filter impulse response. The outputs are then added together with overlapping regions properly managed.

Pros: - Efficiency: OAM is efficient for long input sequences since it divides the computation into smaller blocks. - No need for circular convolution: OAM does not require circular convolution because the FFT of the input blocks and filter impulse response is linearly convolved.

Cons: - Overlap management: Proper management of overlapping regions can be complex, requiring careful handling. - Wasteful computation: It may involve unnecessary computation for zero-padded regions if the input sequence length is not a multiple of the block size

**Overlap Save Method (OSM):**

Method: - In OSM, the input sequence is divided into blocks, but only non-overlapping parts are convolved. Overlapping regions are saved and added to the next block's output to compensate for the lost samples.

Pros: - No overlap handling: OSM avoids the complexity of managing overlapping regions, as only non-overlapping parts are convolved. No wasteful computation: It doesn't involve unnecessary computations for zero-padded regions.

Cons: - Circular convolution needed: OSM requires circular convolution because the FFT of the input blocks and filter impulse response results in circular convolution. Efficiency for long sequences: OSM may be less efficient for very long input sequences because of the potential overhead in circular convolution and overlap management.

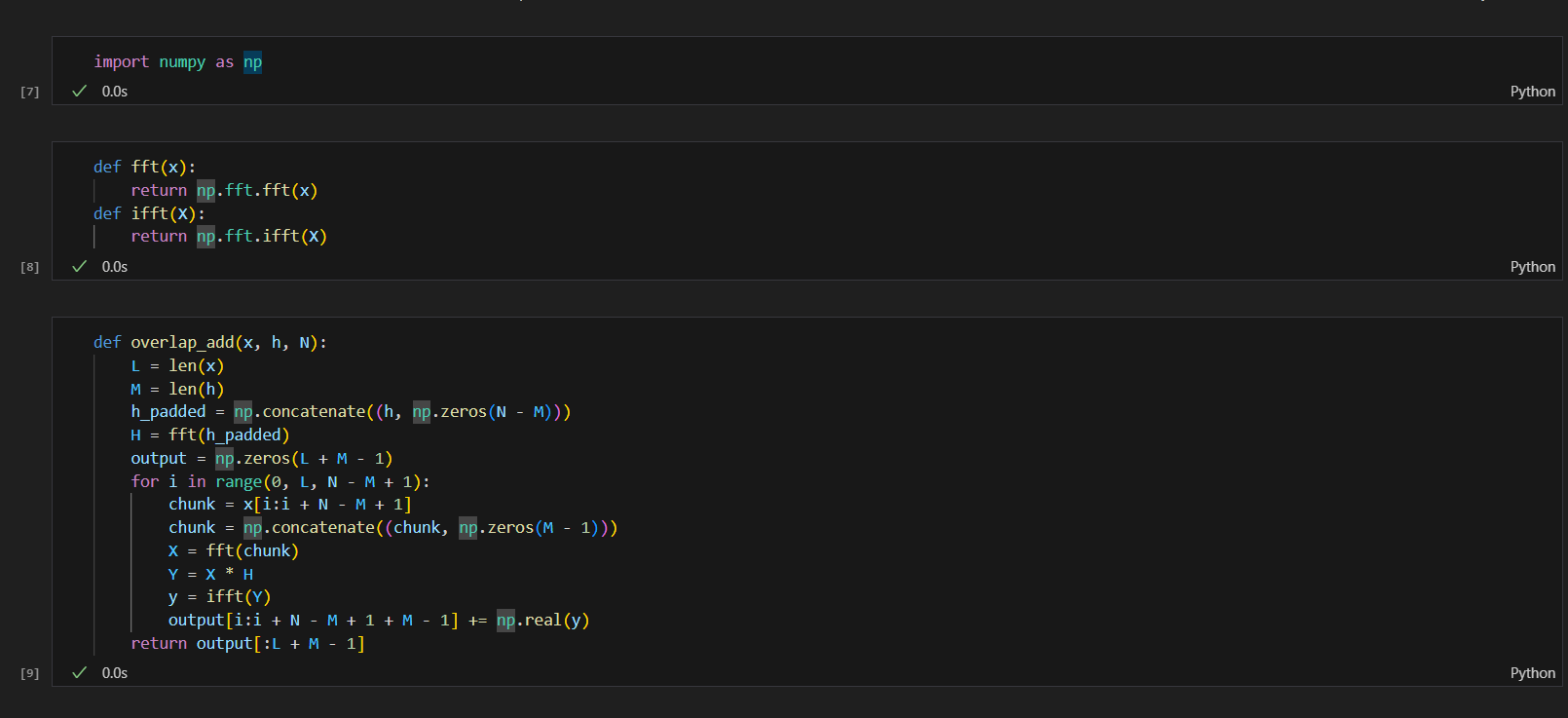
In summary, OAM is more efficient for longer sequences but requires careful handling of overlapping regions, while OSM simplifies the overlap management but might be less efficient for very long sequences due to circular convolution.

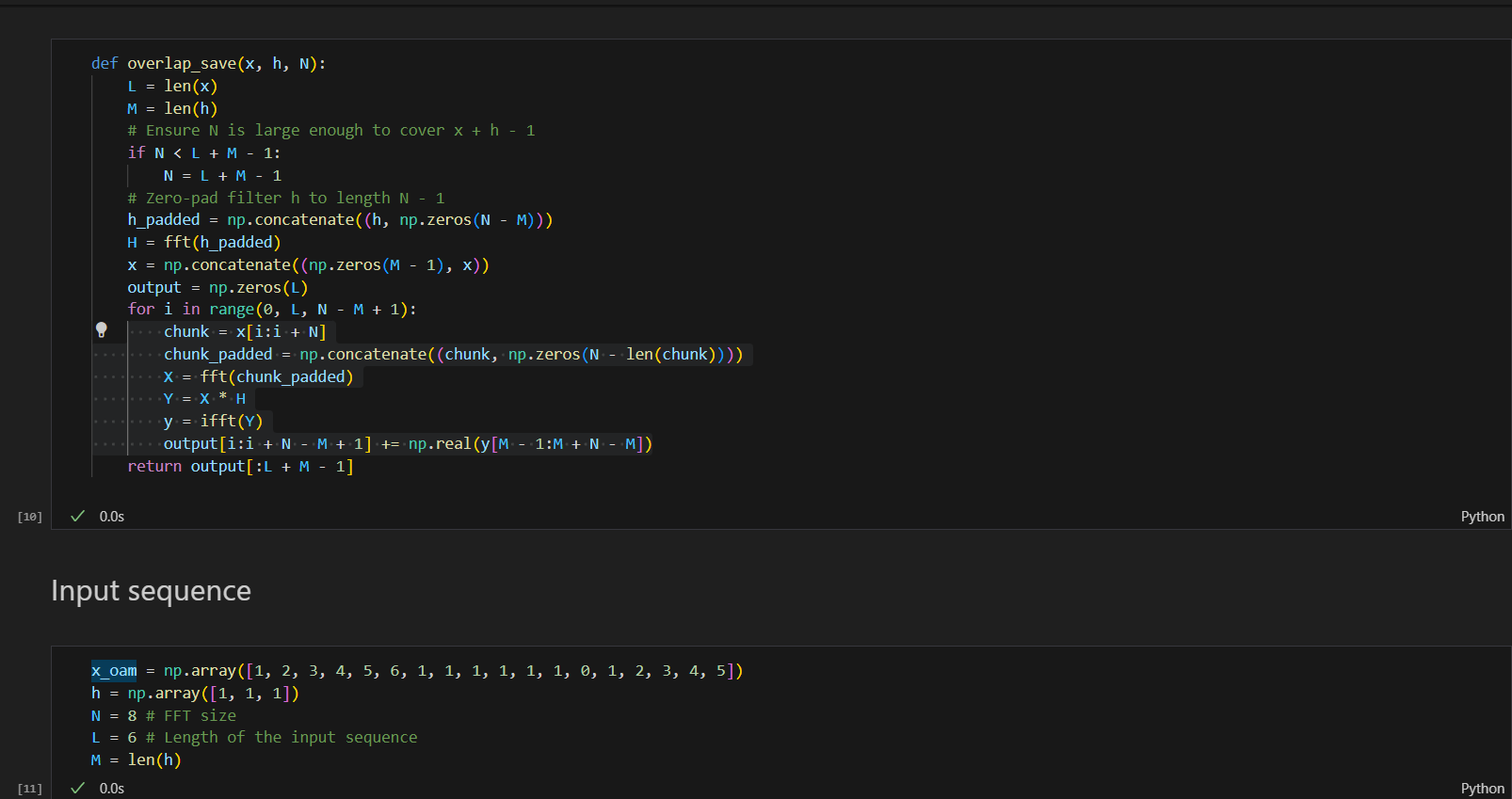
**Problem definition**:

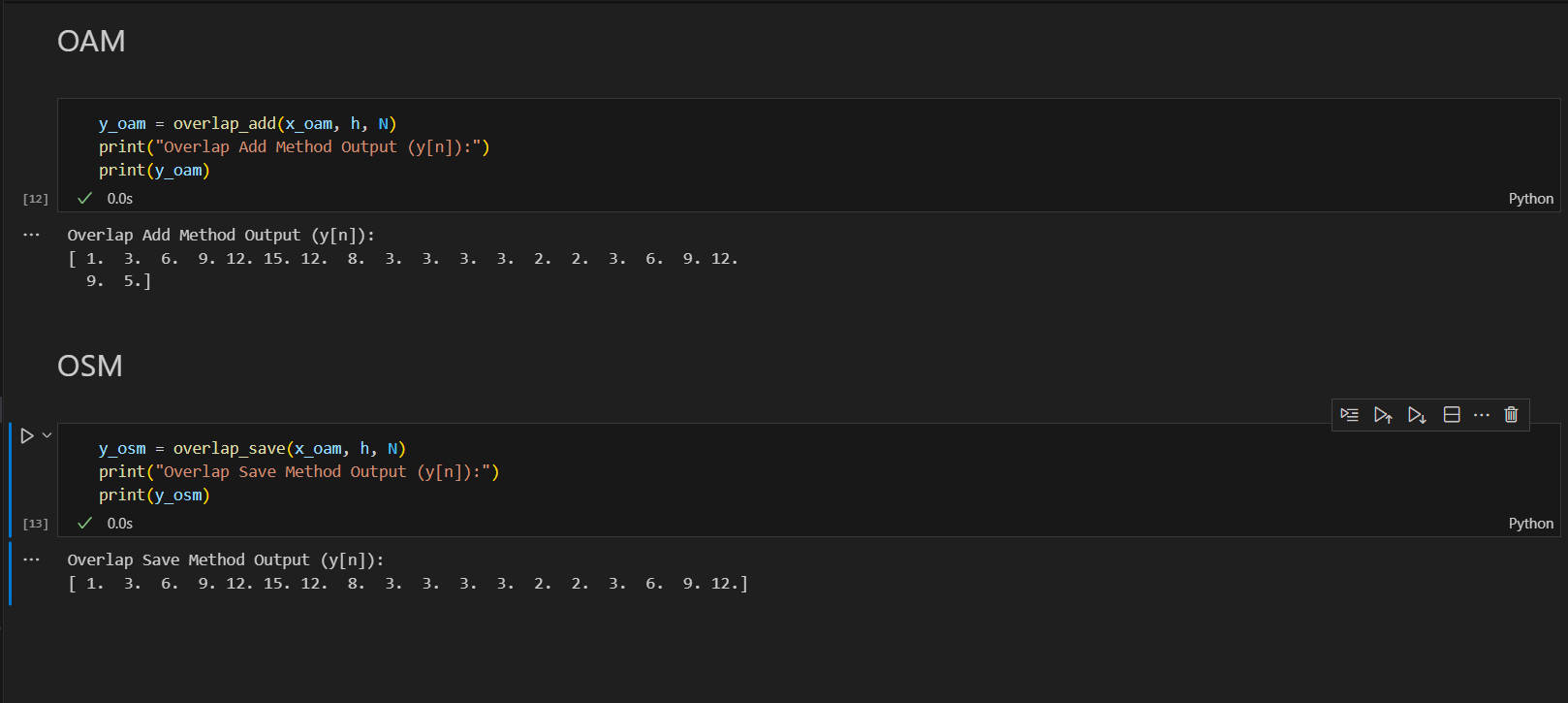
Take long input sequence x[n] and short length sequence h[n]. Find y[n] = x[n] \* h[n] using FFT based Overlap Add Algorithm and Overlap Save Algorithm.

**Input specifications:**

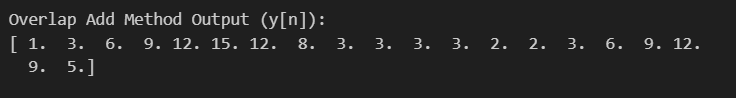
* Length of first Signal L and Signal values
* Length of impulse response of FIR filter Signal M and Signal values.

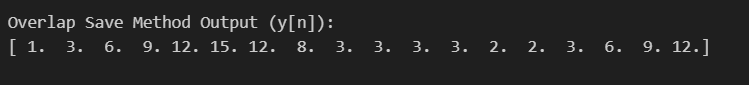
**Code:**

****

****

**Output:**





**Conclusion:**

1. The Overlap-Add and Overlap-Save Method is an efficient practical way to evaluate the discrete convolution of long input signal x[n] and finite length signal h[n].
2. The Overlap-Add and Overlap-Save Method can be implemented using FIR filters and can not be implemented using IIR filters.
3. The Overlap-Add and Overlap-Save Method is a Block Processing Technique

---------------------------------------------------------------------------------------------------------

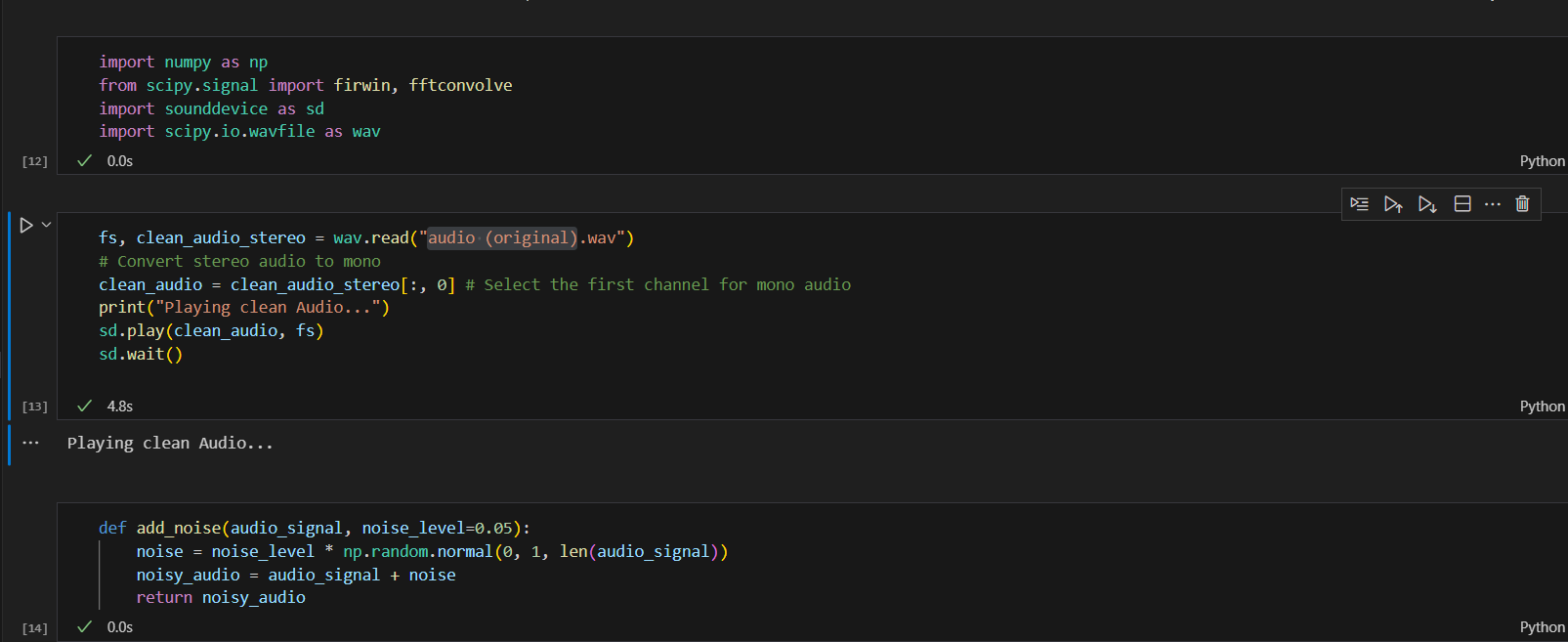
**Application**: Audio Signal Filtering

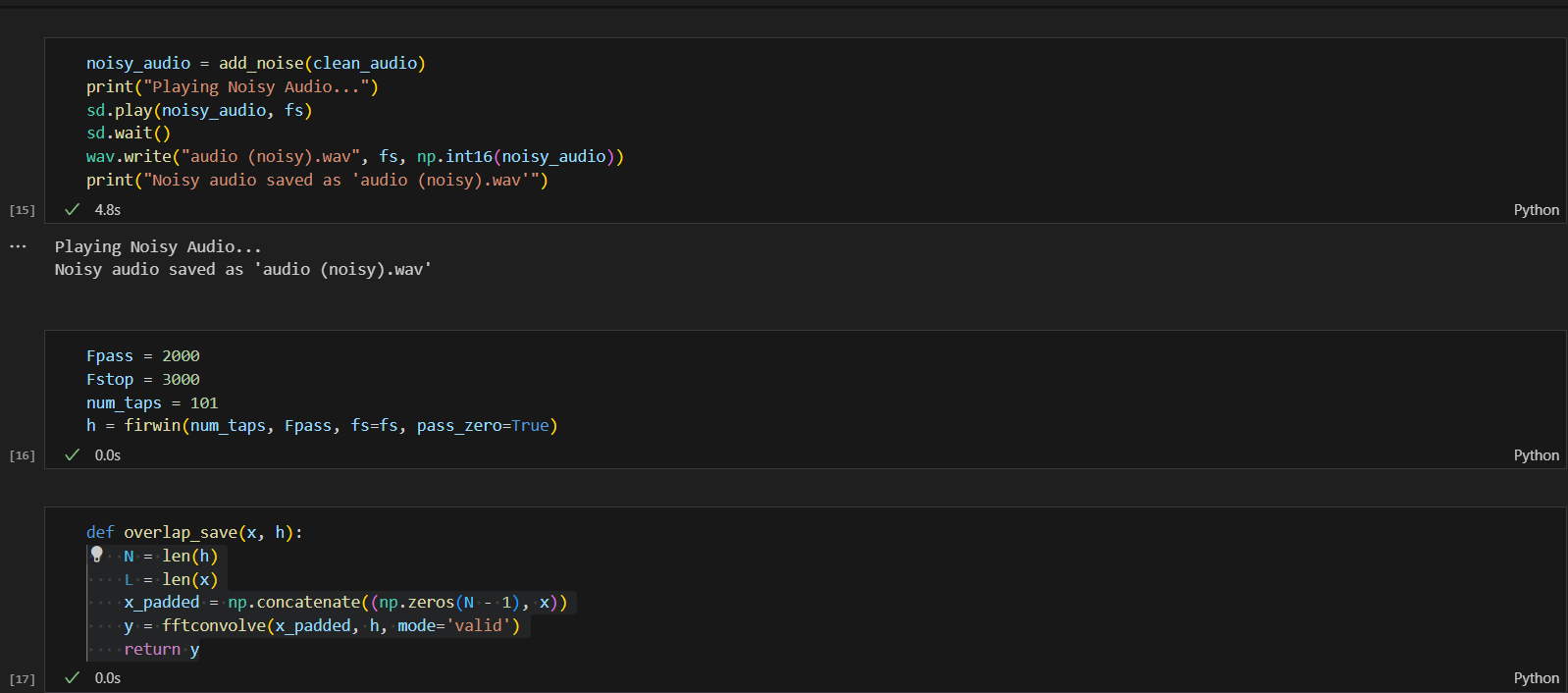
**Problem statement:** Filter the Audio Signal Captured in the presence of noise and improve the quality of sound

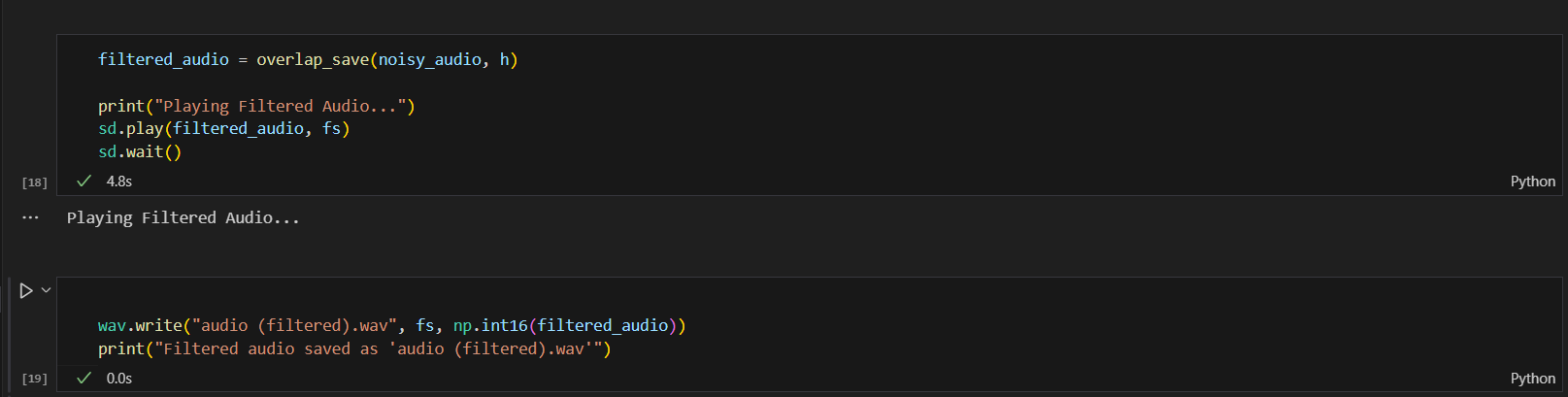
**Algorithm**:

1. Record Audio in the presence of noise with Fs = 8000 Hz==> x[n].
2. Play the recorded signal x[n] and observe the quality of sound.
3. Design FIR Low Pass Filter using MATLAB filter design Tool. Take Fpass = 2000Hz. Fstop = 3000Hs Fs = 8000
4. Filter the audio signal x[n] i.e. Perform Linear Convolution of x[n] and h[n] using either OAM/OSM based on FFT ==>. y[n]
5. Play the filtered signal y[n] and observe the quality of sound

**Code:**

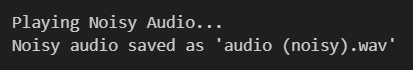






**Output:**







****

**Conclusion:**

Thus, in this experiment I implemented Overlap Add method and Overlap Save Method along with it built a real-time application for audio signal filtering.