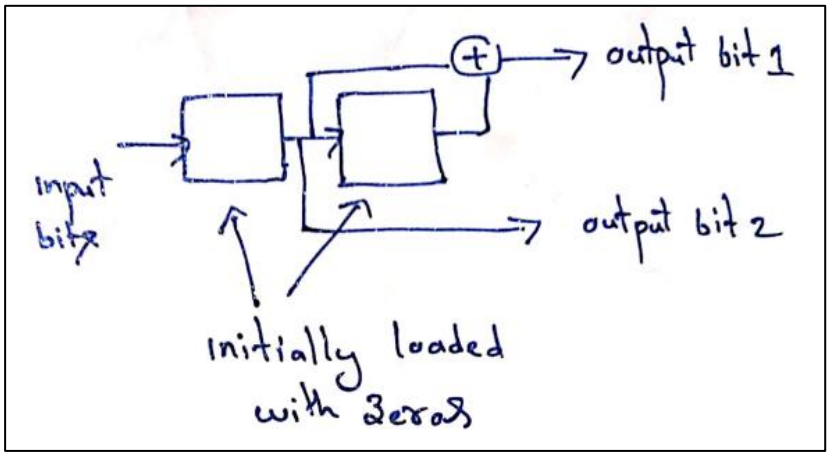
ELL710- CODING THEORY
ASSIGNMENT-2

Vansh Gupta
(2019EE10143)

Working

- First off, I started with generation of the convolution code, and spent a lot of time reading all sorts of books before realising it was not required
- For the decoding, the first part was to make a trellis diagram. Instead of showing it in cascading chain, I represent the same information here in a consolidated table (The same can be found in the first function of my code)
  - 

Current State	Input	Output	New State
00	0	00	00
00	1	11	10
01	0	00	00
01	1	11	10
10	0	10	01
10	1	01	11
11	0	10	01
11	1	01	11

- For the Viterbi decoding, I follow 2 methods- Greedy and Recursion
- For the greedy method, my code finds the state which gives minimum hamming weight and proceeds for that alone
- For the recursive algorithm, I do recursion which leads to a high time complexity but a better accuracy
- In general, something ideal would be a trade-off between the two where we might store the minimum 2-3 and proceed for those alone, but even they would have high complexity
- I test my code for codes of length 6 and 8, and almost always decode it correctly

    - There are some instances where hamming difference from erroneous codeword is same for 2 different trellis output. In these cases, the algorithm might return the alternative as there is no way to know for sure
- For running the code in .ipynb file, execute import cell, and the cell under the heading ‘Viterbi Decoding’. Under the ‘Testing’ section, make changes to the variable ‘init_code’. Upon execution, the loop will automatically test by adding error bit to all the possible locations in the code