I. Operations on binary strings
    A. Binary strings are different from binary numbers
        1. Binary numbers are all binary strings, but binary strings are much more abstract
            a) Binary Strings can mean a lot of things
    B. String Operations
        1. Some of the operations will be defined for binary strings, and others for any string
            a) Reverse
                (1) Input: a string
                (2) Output: A string of the same length, with the characters (bits) in the opposite order
                    (a) Example: Reverse(1011)=1101
            b) Complement (Two's complement)
                (1) Input: Binary String
                (2) Output: A binary string where all the original 1s are changed to zeros and original string zeros are changed to ones.
                    (a) Example: Complement(10110001) = 01001110
            c) Transposition (Transpose)
                (1) Linear: Transpose a string n places
                    (a) Input: A string
                    (b) Output: a string of the same length, the bit in position zero moves n places to the left, position 1 moves to n+1, position 2 to n+2, and so on. Any bits that are moved beyond the original string length are brought around to the right side of the string.
                        (i) Example: Transpose (ABCDEF) 3 places
                            (a) DEFABC
            d) Stringlength
                (1) Input: A string
                (2) Output: a whole number equal to the number of characters in the string
                    (a) Example: stringlength (computer) = 8
                        (i) Stringlength (compsei) = 7
            e) Checksum
                (1) Input: A Binary String
                (2) Output: a whole number equal to the sum of the bit values
                    (a) The number of 1s in the string
            f) MSB/LSB
                (1) Most Significant / Least Significant
                (2) Left Most / Right Most

(3) Bits (the number of bits must be stated or known)
(4) Bit
(5) Byte
(a) B is ambiguous

g) Concatenation
(1) Input: Two strings
(2) Output: a string where the characters of the second string are written to the right of the first in the same order as the originals.
(a) Concatenate 1101 with (or and) 0011: 11010011
(i) A string length of the concatenation is the sum of the length of both the two strings

h) Pad
(1) Input: a binary string of length
(a) Or hexadecimal
(2) Output: a string of length m, starting with m-n zeros, concatenate-l with the original string
(a) Left fill with zeros till you reach the desired string length

i) Split a string into n partitions
(1) Input: a string of length m
(2) Output: If m is divisible by n, when n/m r=0, then write the first m/n bits as an individual string, the next m/n bits as a string and so on.
(a) If m is not divisible by n, keep padding the string until m is divisible by n