Recursive Length Prefix

Notation : `rlp`

Description : RLP encodes arrays of nested binary data to an arbitrary depth; it is the main serialization method for data in Ethereum. RLP encodes mainly structure and does not pay heed to what type of data it is encoding.

Positive RLP integers are represented with the most significant value stored at the lowest memory address (big endian) and without any leading zeroes. As a result, the RLP integer value for `0` is represented by an empty byte-array. If a non-empty deserialized integer begins with leading zeros it is invalid.[1]

The global state database is encoded as RLP for fast traversal and inspection of data. In structure it constitutes a mapping between *addresses* and *account states*. Since it is stored on node operator's computers, the tree can be traversed speedily and without network delay. RLP encodes values as byte-arrays, or as sequences of further values.[2]

This means that:

```
   if  rlp(x)           =  bytearray
 then  rlp(bytearray)   =  true
 elif  rlp(x)           =  value
 then  rlp(value)       =  true
 elif  rlp(x)           =  null
 then  rlp(x)           =  false
```

1. If the RLP-serialized byte-array contains a single byte integer value less than 128, then the output is exactly equal to the input.

2. If the byte-array contains between 2 and 56 bytes then the output is equal to the input number of the byte equal to the length of the byte-array plus 128.

3. In any other case the output is equal to the input prefixed by the minimal-length bytearray which, when interpreted as a big-endian integer, is equal to the length of the input bytearray which is itself prefixed by the number of bytes required to faithfully encode this length value, plus 183.

References

[1] E. Foundation, *Ethereum whitepaper*, `https://github.com/ethereum/wiki/wiki/White-Paper`, 2017.

[2] D. G. Wood, *Ethereum: A secure decentralised generalised transaction ledger*, `https://github.com/ethereum/yellowpaper`, 2017.