


```
contract Proxy {

    bytes32 private constant implementationPos =
        keccak256("proxy.implementation");

    function setAddress(address _implementation) {
        assembly {
            sstore(implementationPos, _implementation)
        }
    }

    function getAddress() returns(address _implementation) {
        assembly {
            _implementation := sload(implementationPos)
        }
    }

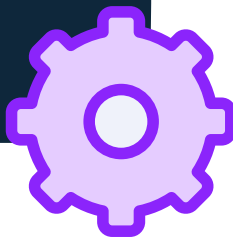
    function () payable public {
        ... delegatecall() ...
    }

}
```



Proxy

```
contract Poll {  
  
    uint cokeVotes = 0;  
    uint pepsiVotes = 0;  
  
    function voteForCoke() returns(uint) {  
        cokeVotes++;  
        return cokeVotes;  
    }  
  
    function voteForPepsi() returns(uint) {  
        pepsiVotes++;  
        return pepsiVotes;  
    }  
  
}
```



Poll

















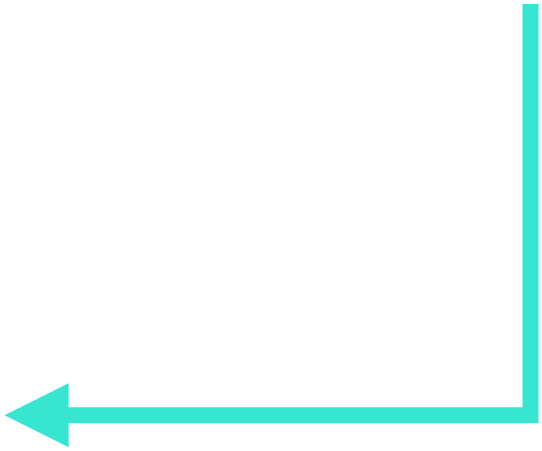




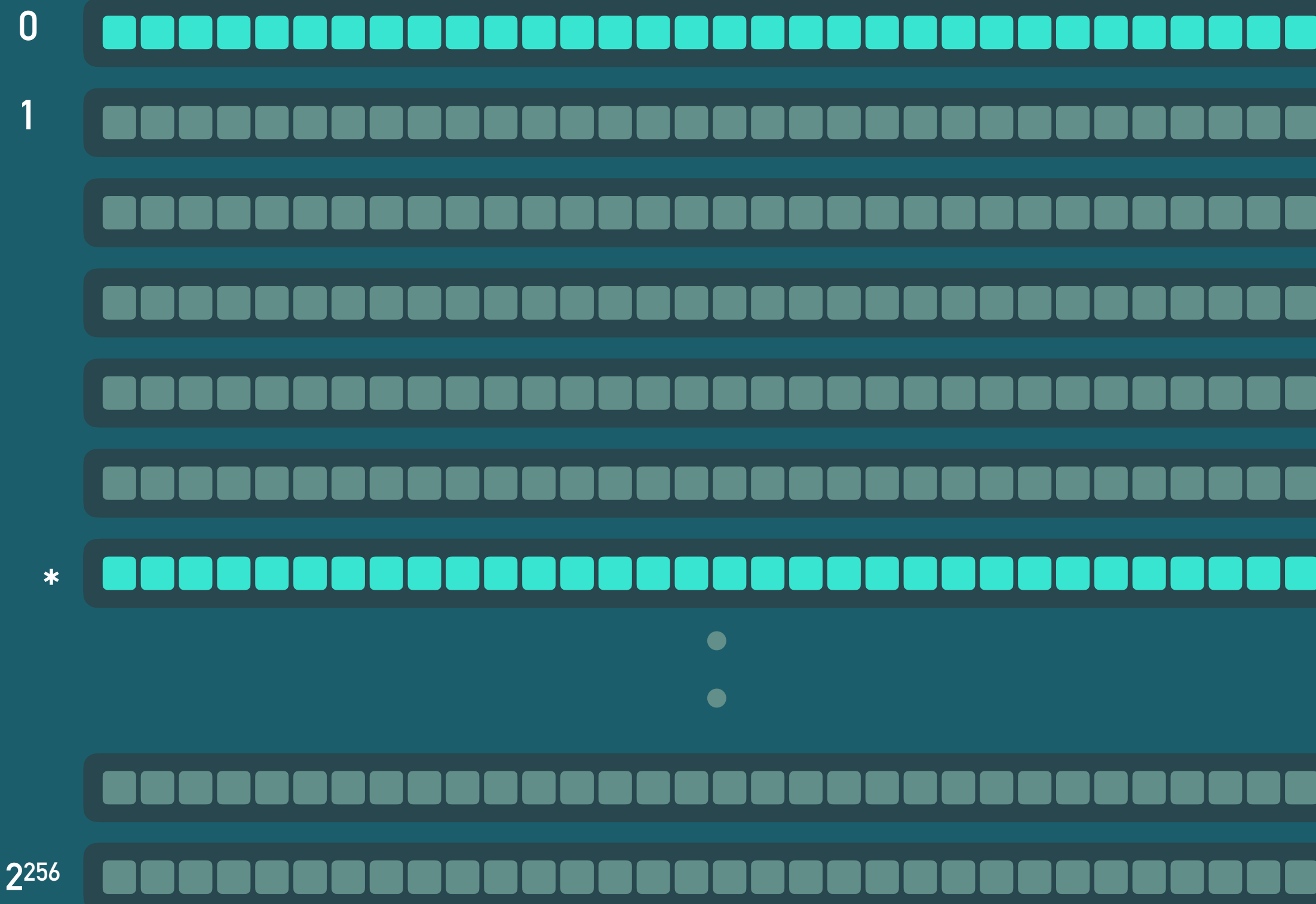






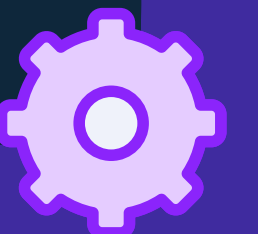


```
contract Proxy {  
  
    bytes32 private constant implementationPos =  
        keccak256("proxy.implementation");  
  
    function setAddress(address _implementation) {  
        assembly {
```

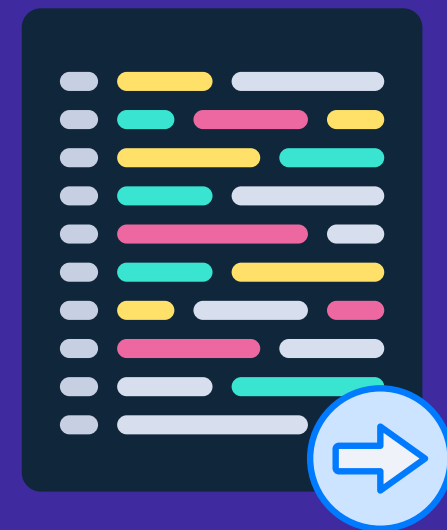


STORAGE

```
contract Poll {  
  
    uint cokeVotes = 0;  
    uint pepsiVotes = 0;  
  
    function voteForCoke() returns(uint) {  
        cokeVotes++;  
        return cokeVotes;  
    }  
  
    function voteForPepsi() returns(uint) {  
        pepsiVotes++;  
        return pepsiVotes;  
    }  
  
}
```



Poll



Proxy



Logic

UNSTRUCTURED STORAGE