Task 2

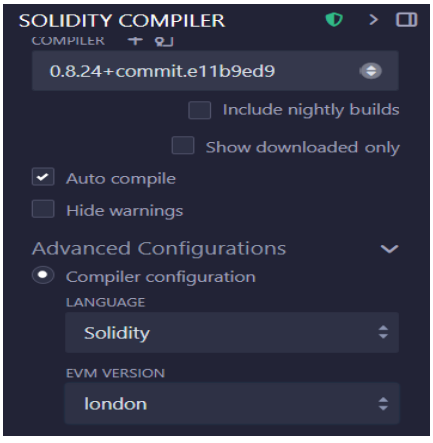
Overview: Explored the concept of smart contracts, ERC-20 token standards, and the Ethereum Virtual Machine (EVM).
Learned about Solidity programming language, constructors, state variables, mappings, events, and functions.
Applied this knowledge to write and deploy a token contract on Remix VM with a fixed initial supply.

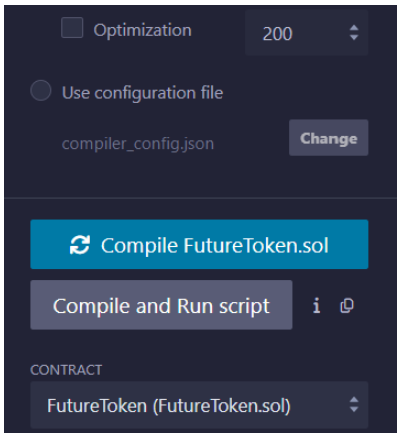
Steps:

1. Turn on Remix IDE and create a smart contract named FutureToken.sol
2. Write code for smart contract.

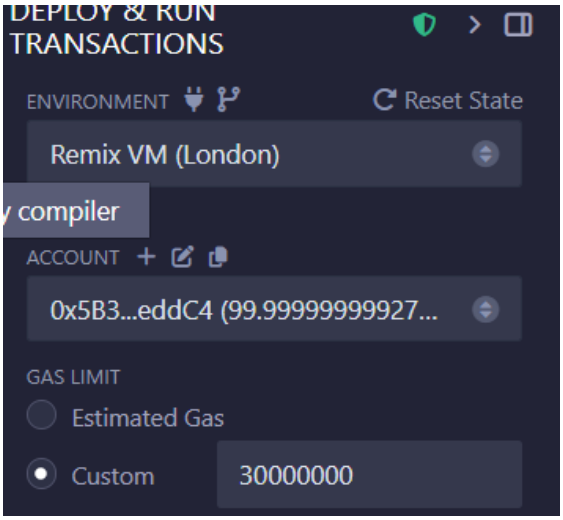
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract FutureToken {
    string public name = "FutureToken";
    string public symbol = "FUT";
    uint8 public decimals = 18;
    uint256 public totalSupply;
    mapping(address => uint256) public balanceOf;

    event Transfer(address indexed from, address indexed to, uint256 value);

    constructor(uint256 initialSupply) {    infinite gas 417200 gas
        totalSupply = initialSupply * 10 ** uint256(decimals);
        balanceOf[msg.sender] = totalSupply;
        emit Transfer(address(0), msg.sender, totalSupply);
    }

    function transfer(address _to, uint256 _value) public returns (bool success) {    infinite gas
        require(balanceOf[msg.sender] >= _value, "Not enough balance");
        balanceOf[msg.sender] -= _value;
        balanceOf[_to] += _value;
        emit Transfer(msg.sender, _to, _value);
        return true;
    }
}
```

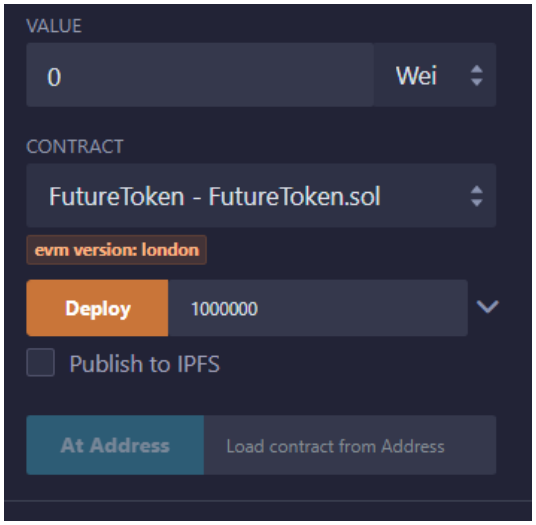
3. In the compiler's tab, select 0.8.24 version and select EVM to London.
4. Click on Compile FutureToken.sol



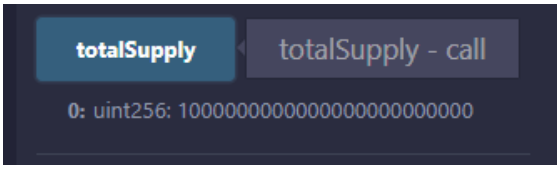


5. In Deploy and Run Transaction tab, select Environment Remix VM(London).
6. Click on custom Gas limit if you want to increase gas limit.
7. Provide initial supply to unit:256 and click on Deploy.

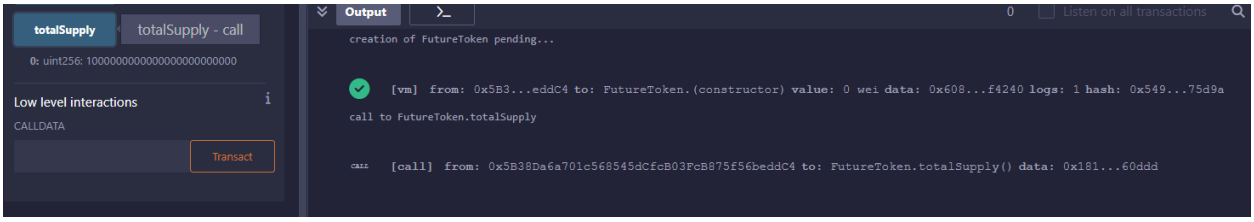




8. It shows total supply of unit256.



9. Smart contract deployed successfully.



Conclusion: I gained practical experience writing a real Solidity contract, compiling it, and deploying it on a blockchain environment.
This task gave me confidence to interact with EVM-based networks and build my own decentralized applications in the future.