

Homework 3 of #1604

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Provider - MetaMask

ACCOUNT: 0x818...09370 (0 ether)

GAS LIMIT: Custom 3000000

VALUE: 0 Wei

CONTRACT: IERC20 - contracts/MyERC-20FT.sol

Deploy

Transactions recorded: 0

Deployed Contracts: 0

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

interface IERC20 {
    function totalSupply() external view returns (uint256);
    function balanceOf(address account) external view returns (uint256);
    function transfer(address recipient, uint256 amount) external returns (bool);
    function transferFrom(address internal sender, address internal recipient, uint256 internal amount) external nonpayable returns (bool);
}

contract ERC20Token is IERC20 {
    string public name;
    string public symbol;
    uint8 public decimals;
    uint256 public totalSupply;

    mapping(address => uint256) public balanceOf;
    mapping(address => uint256) public allowance;
```

```

    constructor(string memory _name, string memory _symbol, uint256 _initialSupply) {
        name = _name;
        symbol = _symbol;
        decimals = 18;
        totalSupply = _initialSupply * 10 ** 18;
        balanceOf[msg.sender] = totalSupply;
    }

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

    function transfer(address _to, uint256 _value) public returns (bool success) {
        require(balanceOf[msg.sender] >= _value, "Insufficient balance");
        require(_to != address(0), "Invalid address");
        balanceOf[msg.sender] -= _value;
        balanceOf[_to] += _value;
        emit Transfer(msg.sender, _to, _value);
        return true;
    }

    function approve(address _spender, uint256 _value) public returns (bool success) {
        require(_spender != address(0), "Invalid address");
        allowance[msg.sender][_spender] = _value;
        emit Approval(msg.sender, _spender, _value);
        return true;
    }
```

```

    function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {
        require(balanceOf[_from] >= _value, "Insufficient balance");
        require(_to != address(0), "Invalid address");
        require(allowance[_from][msg.sender] >= _value, "Allowance exceeded");
        balanceOf[_from] -= _value;
        balanceOf[_to] += _value;
        return true;
    }

    function mint(uint256 _amount) public nonpayable {
        totalSupply += _amount * 10 ** 18;
        balanceOf[msg.sender] += _amount * 10 ** 18;
        emit Transfer(address(0), msg.sender, _amount * 10 ** 18);
    }
}
```