pragma solidity ^0.8.0;

contract TRC20Token {

    string public name = "My Token";

    string public symbol = "MTK";

    uint256 public totalSupply = 1000000 \* 10\*\*18;

    uint8 public decimals = 18;

    mapping(address => uint256) public balanceOf;

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

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

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

    constructor() public {

        balanceOf[msg.sender] = totalSupply;

    }

    function transfer(address \_to, uint256 \_value) public returns (bool success) {

        require(balanceOf[msg.sender] >= \_value);

        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) {

        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);

        require(allowance[\_from][msg.sender] >= \_value);

        balanceOf[\_from] -= \_value;

        balanceOf[\_to] += \_value;

        allowance[\_from][msg.sender] -= \_value;

        emit Transfer(\_from, \_to, \_value);

        return true;

    }

}