// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract MyToken {

string public name = "My Token";

string public symbol = "MYT";

uint256 public totalSupply = 1000000;

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() {

balanceOf[msg.sender] = totalSupply;

}

function transfer(address to, uint256 value) public returns (bool) {

require(balanceOf[msg.sender] >= value, "Insufficient balance");

balanceOf[msg.sender] -= value;

balanceOf[to] += value;

emit Transfer(msg.sender, to, value);

return true;

}

function approve(address spender, uint256 value) public returns (bool) {

allowance[msg.sender][spender] = value;

emit Approval(msg.sender, spender, value);

return true;

}

function transferFrom(address from, address to, uint256 value) public returns (bool) {

require(balanceOf[from] >= value, "Insufficient balance");

require(allowance[from][msg.sender] >= value, "Not authorized");

balanceOf[from] -= value;

balanceOf[to] += value;

allowance[from][msg.sender] -= value;

emit Transfer(from, to, value);

return true;

}

}