

```

pragma solidity ^0.8.2;

contract Token {
    mapping(address => uint) public balances;
    mapping(address => mapping(address => uint)) public allowance;
    uint public totalSupply = 500000000 * 10 ** 18;
    string public name = "Media Investment Coin";
    string public symbol = "MIC20";
    uint public decimals = 18;

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

    constructor() {
        balances[msg.sender] = totalSupply;
    }

    function balanceOf(address owner) public returns(uint) {
        return balances[owner];
    }

    function transfer(address to, uint value) public returns(bool) {
        require(balanceOf(msg.sender) >= value, 'balance too low');
        balances[to] += value;
        balances[msg.sender] -= value;
        emit Transfer(msg.sender, to, value);
        return true;
    }

    function transferFrom(address from, address to, uint value) public
returns(bool) {
        require(balanceOf(from) >= value, 'balance too low');
        require(allowance[from][msg.sender] >= value, 'allowance too low');
        balances[to] += value;
        balances[from] -= value;
        emit Transfer(from, to, value);
        return true;
    }

    function approve(address spender, uint value) public returns (bool) {
        allowance[msg.sender][spender] = value;
        emit Approval(msg.sender, spender, value);
        return true;
    }
}

```