pragma solidity >=0.5.1 <0.6.0;

contract FlawedVoting {

mapping (address => uint256) public remainingVotes; uint256[] public candidates;

address owner;

bool hasEnded = false;

modifier notEnded() { require(!hasEnded);

\_;

}

constructor(uint256 amountOfCandidates) public { candidates.length = amountOfCandidates;

owner = msg.sender;

}

function buyVotes() public payable notEnded { require(msg.value >= 1 ether); remainingVotes[msg.sender] += msg.value / 1e18; msg.sender.transfer(msg.value % 1e18);

}

function vote(uint256 \_candidateID, uint256 \_amountOfVotes) public notEnded { require(\_candidateID < candidates.length); require(remainingVotes[msg.sender] - \_amountOfVotes >= 0); remainingVotes[msg.sender] -= \_amountOfVotes;

candidates[\_candidateID] += \_amountOfVotes;

}

function payoutVotes(uint256 \_amount) public notEnded { require(remainingVotes[msg.sender] >= \_amount); msg.sender.transfer(\_amount \* 1e18); remainingVotes[msg.sender] -= \_amount;

}

function endVoting() public notEnded { require(msg.sender == owner); hasEnded = true;

msg.sender.transfer(address(this).balance);

}

function displayBalanceInEther() public view returns(uint256 balance) { uint balanceInEther = address(this).balance / 1e18;

return balanceInEther;

}

}