pragma solidity 0.6.12;

contract Lottery {

address public manager;

address payable[] public players;

// struct Player {

// address payable investor;

// uint amount;

// }

// Player[] public players;

// constructor - set the manager

constructor () public {

manager = msg.sender;

}

modifier onlyManager() {

require(msg.sender == manager,"Only manager can call this function");

\_;

}

// event to the frontend

event playerInvested(address player, uint amount);

event winnerSelected(address winner, uint amount);

// Invest money by players - anyone in the world

function invest() payable public { // manager should not invest

require(msg.sender != manager,"Manager cannot invest");

// the person should invest minimum 0.1 ether - exactly invest 3 ether

require(msg.value >= 0.1 ether,"Invest minimum of 0.1 ether");

// i want to keep a track of who all invested

players.push(msg.sender);

// Player memory tempPlayer;

// tempPlayer.investor = msg.sender;

// tempPlayer.amount = msg.value;

emit playerInvested(msg.sender,msg.value);

}

// get balance - current balance

function getBalance() public view onlyManager returns(uint) {

// only manager should see balance

return address(this).balance;

}

//random function

function random() private view returns(uint) {

return uint(keccak256(abi.encodePacked(block.timestamp,block.difficulty,players.length)));

}

// manager clicks a function , it should

function selectWinner() public onlyManager {

// only manager can call this function

//select a random number - pseudo random number - do not use this in production

// Use ORACLES to find a random number

// first take some global variables, encode it, hash it, convert to uint

uint r = random();

//modulo it with number of players

uint index = r % players.length;

//map the reminder to a index in the array

address payable winner = players[index];

//.investor;

emit winnerSelected(winner,address(this).balance);

//transfer all the money in the contract to the address in the array

winner.transfer(address(this).balance);

//then make the array empty

players = new address payable[](0);

}

}