Chellenge 206

// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <0.9.0;

// Import SafeMath library

import "@openzeppelin/contracts/utils/math/SafeMath.sol";

contract BettingHelper {

using SafeMath for uint256;

address public owner;

// This event is emitted when a bet is placed.

event BetPlaced(address indexed bettor, uint8 betAmount, uint8 odds);

// This event is emitted when a payout is made.

event Payout(address indexed bettor, uint256 payout);

constructor() {

owner = msg.sender;

}

// Function to place a bet

function placeBet(uint8 betAmount, uint8 odds) external returns (uint256) {

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

require(betAmount > 0, "Bet amount must be greater than 0");

require(odds > 0, "Odds must be greater than 0");

// Emit the event for a new bet.

emit BetPlaced(msg.sender, betAmount, odds);

// Calculate the potential payout with SafeMath.

uint256 potentialPayout = calculatePayout(betAmount, odds);

// Emit the event for the payout.

emit Payout(msg.sender, potentialPayout);

return potentialPayout;

}

// Function to calculate the payout.

function calculatePayout(uint8 betAmount, uint8 odds) internal pure returns (uint256) {

// Use SafeMath for multiplication to prevent overflow.

uint256 payout = uint256(betAmount).mul(uint256(odds));

return payout;

}

}