

```

pragma solidity ^0.5.0;

// lvl 1: equal split
contract AssociateProfitSplitter {
    // @TODO: Create three payable addresses representing
    `employee_one`, `employee_two` and `employee_three`.
    address payable employee_one;
    address payable employee_two;
    address payable employee_three;

    constructor (address payable _one, address payable _two, address
payable _three) public payable {
        employee_one = _one;
        employee_two = _two;
        employee_three = _three;
    }

    function balance() public view returns(uint) {
        return address(this).balance;
    }

    function deposit() public payable {
        // @TODO: Split `msg.value` into three

        uint amount = msg.value/3; // Your code here!

        // @TODO: Transfer the amount to each employee
        // Your code here!
        employee_one.transfer(amount);
        employee_two.transfer(amount);
        employee_three.transfer(amount);
        // @TODO: take care of a potential remainder by sending back
to HR (`msg.sender`)
        // Your code here!
        msg.sender.transfer((msg.value-amount)*3);
    }

    function() external payable {
        // @TODO: Enforce that the `deposit` function is called in the
fallback function!
        // Your code here!
        deposit();
    }
}

```