

```
pragma solidity ^0.5.0;
```

```
// lvl 1: equal split
```

```
// three addresses created
```

```
contract AssociateProfitSplitter {
```

```
    address payable employee_one;
```

```
    address payable employee_two;
```

```
    address payable employee_three;
```

```
    constructor(address payable _one, address payable _two, address payable _three) public {
```

```
        employee_one = _one;
```

```
        employee_two = _two;
```

```
        employee_three = _three;
```

```
    }
```

```
    function balance() public view returns(uint) {
```

```
        return address(this).balance;
```

```
    }
```

```
    function deposit() public payable {
```

```
        // split `msg.value` into three
```

```
        uint amount = msg.value/3;
```

```
// transfer the amount to each employee
employee_one.transfer(amount);
employee_two.transfer(amount);
employee_three.transfer(amount);

// take care of a potential remainder by sending back to HR which is msg.sender
msg.sender.transfer(msg.value - (amount * 3));

}

// enforce that the `deposit` function is called in the fallback function!
function() external payable {
    deposit();
}

}
```