pragma solidity ^0.5.0;

// lvl 2: tiered split

contract TieredProfitSplitter {

address payable employee\_one; // ceo

address payable employee\_two; // cto

address payable employee\_three; // bob

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

employee\_one = \_one;

employee\_two = \_two;

employee\_three = \_three;

}

// Should always return 0! Use this to test your `deposit` function's logic

function balance() public view returns(uint) {

return address(this).balance;

}

function deposit() public payable {

uint points = msg.value / 100; // Calculates rudimentary percentage by dividing msg.value into 100 units

uint total;

uint amount;

// @TODO: Calculate and transfer the distribution percentage

// Step 1: Set amount to equal `points` \* the number of percentage points for this employee

// Step 2: Add the `amount` to `total` to keep a running total

// Step 3: Transfer the `amount` to the employee

amount = points \* 60;

total += amount;

employee\_one.transfer(amount);

// @TODO: Repeat the previous steps for `employee\_two` and `employee\_three`

// Your code here!

amount = points \* 25;

total += amount;

employee\_two.transfer(amount);

amount = points \* 15;

total += amount;

employee\_two.transfer(amount);

employee\_one.transfer(msg.value - total); // ceo gets the remaining wei

}

function() external payable {

deposit();

}

}