

Syntax Errors:

a. What will be the output of the following function? If there is a problem in it, explain the solution.

```
function add() public pure returns (uint) {  
    uint a = 7;  
    uint b = 9;  
    return a - b;  
}
```

The answer to the difference between 7 and 9 is a negative value which is out of the scope of uint. We have to return int in order to make it work. Also, the type of a and b should be int.

The output should be -2.

Correct working function:

```
function add() public pure returns (int) {  
    int a = 7;  
    int b = 9;  
    return a - b;  
}
```

b. What will be the output of the following function? If there is a problem in it, explain the solution.

```
function divide() public pure returns (uint) {  
    uint a = 19;  
    uint b = 9;  
    return a / b;  
}
```

The output of the function is 2. Although the exact output is 2.1, as solidity does not support decimal values so the value after the point is ignored. The division would result in a floor of the calculation with the remainder discarded.

C. Debug the following function and explain how it can be fixed.

```
function transfer(address receiver, uint numberOfTokens) public returns (bool) {
    require(numberOfTokens <= balances[receiver]);
    balances[msg.sender] = balances[msg.sender] - numberOfTokens;
    balances[receiver] = balances[receiver] + numberOfTokens;
    return true;
}
```

In the require, number of tokens should compare with the balances of msg.sender and not the receiver.

Capital O in numberOfTokens in require statement.

```
function transfer(address receiver, uint numberOfTokens) public returns (bool) {
    require(numberOfTokens <= balances[msg.sender]);
    balances[msg.sender] = balances[msg.sender] - numberOfTokens;
    balances[receiver] = balances[receiver] + numberOfTokens;
    return true;
}
```

This function alone will not work as first we have to set some amount to msg.sender address. Only after that, we will be able to transfer tokens to other addresses. The whole implementation will look something like this:

```
address public minter;
    constructor() {
        minter = msg.sender;
    }
    mapping(address => uint) public balances;

    function setNew(address _receiver, uint _amount) public {
        require(msg.sender == minter, "unauthorized");
```

```
    balances[_receiver] += _amount;
}
```

```
function transfer(address receiver, uint numberOfTokens) public returns (bool) {
    require(numberOfTokens <= balances[msg.sender]);
    balances[msg.sender] = balances[msg.sender] - numberOfTokens;
    balances[receiver] = balances[receiver] + numberOfTokens;
    return true;
}
```

d. How many times can this function be called?

```
uint private papersChecked = 1;
function check() private {
    require(papersChecked < 10);
    Paperschecked++;
}
```

Correct the spelling of papersChecked in function when incrementing.

This function can be called 9 times after deploying.

e. Explain the error in the following function. How can it be fixed?

```
String _totalSupply = 0;
function mint(address account, uint256 amount) onlyOwner public {
    require(account != address(0));

    _totalsupply += amount;
}
```

Total supply should be a uint.

Spelling of _totalSupply corrected.

```
uint _totalSupply = 0;  
function mint(address account, uint256 amount) public {  
    require(account != address(0));  
  
    _totalSupply += amount;  
}
```