```
pragma solidity ^0.4.11;



/**
  * @title SafeMath
  * @dev Math operations with safety checks that throw on error
  */
library SafeMath

{

    function mul(uint256 a, uint256 b) internal constant returns (uint256)
    {

        uint256 c = a * b;
        assert(a == 0 || c / a == b);
        return c;

    }

    function div(uint256 a, uint256 b) internal constant returns (uint256)
    {

        // assert(b > 0); // Solidity automatically throws when dividing by 0
        uint256 c = a / b;
```

```
        // assert(a == b * c + a % b); // There is no case in which this doesn't hold
        return c;

    }

    function sub(uint256 a, uint256 b) internal constant returns (uint256)
    {

        assert(b <= a);
        return a - b;

    }

    function add(uint256 a, uint256 b) internal constant returns (uint256)
    {

        uint256 c = a + b;
        assert(c >= a);
        return c;

    }
}
```