Quantstamp Security Assessment Certificate

December 18th 2020 — Quantstamp Verified

# SyncDAO/GrowUSD

This security assessment was prepared by Quantstamp, the leader in blockchain security

## Executive Summary

| | |
|---|---|
| Type | DeFi protocol |
| Auditors | Shunsuke Tokoshima, Software Engineer<br>Kacper Bąk, Senior Research Engineer<br>Kevin Feng, Blockchain Researcher |
| Timeline | 2020-10-08 through 2020-12-11 |
| EVM | Muir Glacier |
| Languages | Solidity, Javascript |
| Methods | Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review |
| Specification | None |
| Documentation Quality | Low |
| Test Quality | Medium |

Source Code

| Repository | Commit |
|---|---|
| core | 1a803d3 |
| rtoken-monorepo | 8b757d2 |
| core | 2d94480 |
| rtoken-monorepo | a30cb60 |

| | |
|---|---|
| Total Issues | **5** (5 Resolved) |
| High Risk Issues | **1** (1 Resolved) |
| Medium Risk Issues | **1** (1 Resolved) |
| Low Risk Issues | **3** (3 Resolved) |
| Informational Risk Issues | 0 (0 Resolved) |
| Undetermined Risk Issues | 0 (0 Resolved) |


ALL ISSUES ADDRESSED

| | |
|---|---|
| High Risk | The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users. |
| Medium Risk | The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact. |
| Low Risk | The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances. |
| Informational | The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth. |
| Undetermined | The impact of the issue is uncertain. |
| Unresolved | Acknowledged the existence of the risk, and decided to accept it without engaging in special efforts to control it. |
| Acknowledged | The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings). |
| Resolved | Adjusted program implementation, requirements or constraints to eliminate the risk. |
| Mitigated | Implemented actions to minimize the impact or likelihood of the risk. |

# Summary of Findings

Through reviewing the code, we found **5 potential issues** of various levels of severity: one high, one medium, three low severity. We recommend addressing the findings before using them in production. In addition, we provided ideas for further code and documentation improvements. We strongly recommend adding some tests as noted in `QSP-1`.

**Update:** As of commit `2d94480` in the `core` repo and commit `a30cb60` in the `rtoken-monorepo` repo, all QSP issues have been fixed.

| ID | Description | Severity | Status |
|---|---|---|---|
| QSP-1 | Lack of Tests | High | Fixed |
| QSP-2 | Old Deployment Script | Medium | Fixed |
| QSP-3 | Unlocked Pragma | Low | Fixed |
| QSP-4 | Insufficient Input Validation | Low | Fixed |
| QSP-5 | Possibly reverting `rewardPerToken()` | Low | Fixed |

# Quantstamp Audit Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

### Methodology

The Quantstamp auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

### Toolset

The notes below outline the setup and steps performed in the process of this audit.

#### Setup

Tool Setup:

- Slither v0.6.12

Steps taken to run the tools:

1. Installed the Slither tool: `pip install slither-analyzer`
2. Run Slither from the project directory: `slither .`s

# Findings

## QSP-1 Lack of Tests

Severity: *High Risk*

**Status:** Fixed

**Description:** As of commits `1a803d3` and `8b757d2`, `syncdao/core` repo does not contain test scripts for the contracts and `growreturns/rtoken-monorepo` does not contain test scripts corresponding to the diff with `rtoken-project/rtoken-monorepo` repo (`41cfd3b`).

**Recommendation:** We highly recommend implementing some basic tests to achieve full test coverage. Given the brevity of the codebase, this should not be too difficult to achieve, and would go a long way in bringing confidence and assurance to the functionality, consistency and security of the code.
Especially, it would be beneficial to add:

- Unit tests for staking/ unstaking functionalities in `gDaiStaking.sol`
- Integration tests for `transfer()` function in `Rtoken.sol`.

**Update:** As of commit `2d94480` in `core` and commit `a30cb60` in `rtoken-monorepo`, tests as been added to both repositories.

## QSP-2 Old Deployment Script

Severity: *Medium Risk*

**Status:** Fixed

**Description:** `deploy-rtoken.js` does not refer to the implementation of `gDAI` and `CompoundAllocationStrategyV2`. If the Mosendo team deploys contracts with the script, there would be some unintentional behaviors such as the lack of interfaces to claim `COMP` tokens.

**Recommendation:** Update the deployment script given the new contracts.
**Update:** As of commit `2d94480` in `core`, the deploy script `deploy_dao.js` has been added.

## QSP-3 Unlocked Pragma

Severity: *Low Risk*

**Status:** Fixed

**Description:** Every Solidity file specifies in the header a version number of the format `pragma solidity (^)0.4.*`. The caret (`^`) before the version number implies an unlocked pragma, meaning that the compiler will use the specified version *and above*, hence the term "unlocked". Several contracts (e.g. `gDaiStaking.sol`, `CompoundAllocationStrategy.sol`, `CompoundAllocationStrategyV2.sol`, `IRToken.sol`, `ISdgStaking.sol`, `RToken.sol`, `RTokenStorage.sol`, `StorageLayout.sol`, `rDAI.sol`, `rSAI.sol`) have this risk.

**Recommendation:** For consistency and to prevent unexpected behavior in the future, it is recommended to remove the caret to lock the file onto a specific Solidity version.
**Update:** As of commit `2d94480` in `core` and commit `a30cb60` in `rtoken-monorepo`, the carets has been removed for the files in scope.

## QSP-4 Insufficient Input Validation

Severity: *Low Risk*

**Status:** Fixed

**Description:** Functions should always validate input parameters before using their values. One common mistake that could occur due to missing input validation is that funds are sent to the `0x0` address.

**Recommendation:** It is recommended to implement validations including:

- `gDaiStaking.sol`: Checks for non-zero addresses in L58, L59 (`constructor ()`), L103 (`stake()`), L110 (`withdraw()`), and L155 (`updateReward()`).
- `CompoundAllocationStrategyV2.sol`: Checks for non-zero addresses in L20 (`constructor ()`) and L32 (`transferCompReceiverRights ()`). Additionally, it is also recommended to take `cToken_` as an `address` in L20, check it is not a non-zero address, and set it as `cTokenContract = CErc20Interface(cToken_);`.
- `RToken.sol`: Checks for non-zero addresses in L493 (`setStakingPool()`).

**Update:** As of commit `2d94480` in `core` and commit `a30cb60` in `rtoken-monorepo`, all points have been fixed.

## QSP-5 Possibly reverting `rewardPerToken()`

Severity: *Low Risk*

**Status:** Fixed

**File(s) affected:** `gDaiStaking.sol`

**Description:** The function `rewardPerToken()` reverts if `lastTimeRewardApplicable() < lastUpdateTime`.

**Recommendation:** It would be beneficial to check if this behavior is intentional and consider re-implementing the function so that it returns 0 if `lastTimeRewardApplicable() < lastUpdateTime` as needed.
**Update:** As of commit `2d94480` in `core`, this has been fixed.

# Automated Analyses

### Slither

The analysis was completed successfully. No issues were detected except for some false-positives.

# Code Documentation

- `gDaiStaking.sol`: It would be beneficial to note that `stakingController` is `gDai` token contract and `rewardsToken` is `SDG` token.
- `gDaiStaking.sol`, `CompoundAllocationStrategyV2.sol`: It is preferable to have comments for each function.
- It would be beneficial to prepare README files showing the intended behavior of contracts.
- `RTokenStorage.sol` L50: Incomplete comment.

## Adherence to Best Practices

- `gDaiStaking.sol`: `uint` should be changed to `uint256` for consistency in L46, L60, L61, L137. **Fixed**
- `gDaiStaking.sol` L145: This validation is always passed. This can be removed. **Fixed**
- `gDaiStaking.sol`: L41, L42, L47: No need for explicitly setting default values. **Fixed**
- `gDaiStaking.sol`: Confusing variable naming `userRewardPerTokenPaid`, where this value isn't necessarily paid but instead is the reward that is checkpointed. **Fixed**
- `CompoundAllocationStrategyV2.sol` L54: It would be beneficial to rename `claimAndTransferComp(uint amount_)` to a name more explicitly suggesting that this function withdraws `COMP` tokens partially.

## Test Results

### Test Suite Results

**Note:** As of commits `1a803d3` and `8b757d2`, `syncdao/core` repo does not contain test scripts for the contracts and `growreturns/rtoken-monorepo` does not contain test scripts corresponding to the diff with `rtoken-project/rtoken-monorepo` repo (`41cfd3b`).

**Update:** As of commit `2d94480` in `core`, tests has been added for the contract `gDaiStaking.sol`. As of commit `a30cb60` in `rtoken-monorepo`, tests from the original rtoken fork has been added with addition to tests for staking. The outputs of these tests are shown below.

```
  Contract: CToken
admin is 0x627306090abaB3A6e1400e9345bC60c78a8BEf57
bingeBorrower is 0xf17f52151EbEF6C7334FAD080c5704D77216b732
customer1 is 0xC5fdf4076b8F3A5357c5E395ab970B5B54098Fef
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
cToken.mint 100 to customer1: started
cToken.mint 100 to customer1: done, gas used 147962, gas price 20 Gwei
cToken.redeem 100: started
cToken.redeem 100: done, gas used 112536, gas price 20 Gwei
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
cToken.redeem 100: started
cToken.redeem 100: done, gas used 116736, gas price 20 Gwei
token redeemed 920.00011
cToken.redeemUnderlying 10: started
cToken.redeemUnderlying 10: done, gas used 117306, gas price 20 Gwei
     ✓ #1 cToken basic operations (1884ms)

  Contract: RToken
admin is 0x627306090abaB3A6e1400e9345bC60c78a8BEf57
bingeBorrower is 0xf17f52151EbEF6C7334FAD080c5704D77216b732
customer1 is 0xC5fdf4076b8F3A5357c5E395ab970B5B54098Fef
customer2 is 0x821aEa9a577a9b44299B9c15c88cf3087F3b5544
customer3 is 0x0d1d4e623D10F9FBA5Db95830F7d3839406C6AF2
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
     ✓ #0 initial test condition (56ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1: started
rToken.mint 100 to customer1: done, gas used 417952, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00000 expected 100.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 1000.00000
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00100 expected 100.00100
customer1 interestPayable 0.00100 expected 0.00100
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 1000.00000
rToken.redeem 10 by customer1: started
rToken.redeem 10 by customer1: done, gas used 343938, gas price 20 Gwei
customer1 tokenBalance 90.00102 expected 90.00102
customer1 receivedLoan 90.00000 expected 90.00000
customer1 receivedSavings 90.00102 expected 90.00102
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00102 expected 0.00102
customer1 lDebt 90.00000
customer1 rInterest 0.00102
customer1 sInternalAmount 900.00102
rToken.payInterest to customer1: started
```

```
rToken.payInterest to customer1: done, gas used 117778, gas price 20 Gwei
customer1 tokenBalance 90.00103 expected 90.00103
customer1 receivedLoan 90.00000 expected 90.00000
customer1 receivedSavings 90.00103 expected 90.00103
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00103 expected 0.00103
customer1 lDebt 90.00000
customer1 rInterest 0.00103
customer1 sInternalAmount 900.00102
rToken.payInterest to customer1 again: started
rToken.payInterest to customer1 again: done, gas used 117778, gas price 20 Gwei
customer1 tokenBalance 90.00104 expected 90.00104
customer1 receivedLoan 90.00000 expected 90.00000
customer1 receivedSavings 90.00104 expected 90.00104
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00104 expected 0.00104
customer1 lDebt 90.00000
customer1 rInterest 0.00104
customer1 sInternalAmount 900.00102
rToken.redeem 2 of customer1 to customer2: started
rToken.redeem 2 of customer1 to customer2: done, gas used 327931, gas price 20 Gwei
rToken.transfer 10 from customer1 to customer3: started
rToken.transfer 10 from customer1 to customer3: done, gas used 345884, gas price 20 Gwei
customer1 tokenBalance 78.00109 expected 78.00109
customer1 receivedLoan 78.00000 expected 78.00000
customer1 receivedSavings 78.00109 expected 78.00109
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00109 expected 0.00109
customer1 lDebt 78.00000
customer1 rInterest 0.00109
customer1 sInternalAmount 780.00233
customer3 tokenBalance 10.00000 expected 10.00000
customer3 receivedLoan 10.00000 expected 10.00000
customer3 receivedSavings 10.00000 expected 10.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 10.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 99.99890
rToken.transfer 5 from customer3 to customer1: started
rToken.transfer 5 from customer3 to customer1: done, gas used 351236, gas price 20 Gwei
customer1 tokenBalance 83.00110 expected 83.00110
customer1 receivedLoan 83.00000 expected 83.00000
customer1 receivedSavings 83.00110 expected 83.00110
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00110 expected 0.00110
customer1 lDebt 83.00000
customer1 rInterest 0.00110
customer1 sInternalAmount 830.00177
customer3 tokenBalance 5.00000 expected 5.00000
customer3 receivedLoan 5.00000 expected 5.00000
customer3 receivedSavings 5.00000 expected 5.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 5.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 49.99946
     ✓ #2 normal operations with zero hatter (4781ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1 with a hat benefiting admin(90%) and customer2(10%): started
rToken.mint 100 to customer1 with a hat benefiting admin(90%) and customer2(10%): done, gas used 711351, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
admin tokenBalance 0.00000 expected 0.00000
admin receivedLoan 90.00000 expected 90.00000
admin receivedSavings 90.00000 expected 90.00000
admin interestPayable 0.00000 expected 0.00000
admin cumulativeInterest 0.00000 expected 0.00000
admin lDebt 90.00000
admin rInterest 0.00000
admin sInternalAmount 900.00000
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 10.00000 expected 10.00000
customer2 receivedSavings 10.00000 expected 10.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 10.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 100.00000
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
admin tokenBalance 0.00000 expected 0.00000
admin receivedLoan 90.00000 expected 90.00000
admin receivedSavings 90.00090 expected 90.00090
admin interestPayable 0.00090 expected 0.00090
admin cumulativeInterest 0.00000 expected 0.00000
admin lDebt 90.00000
admin rInterest 0.00000
admin sInternalAmount 900.00000
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 10.00000 expected 10.00000
customer2 receivedSavings 10.00010 expected 10.00010
customer2 interestPayable 0.00010 expected 0.00010
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 10.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 100.00000
rToken.redeem 10 to customer1: started
rToken.redeem 10 to customer1: done, gas used 363155, gas price 20 Gwei
customer1 tokenBalance 90.00000 expected 90.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
admin tokenBalance 0.00000 expected 0.00000
admin receivedLoan 81.00000 expected 81.00000
admin receivedSavings 81.00091 expected 81.00091
admin interestPayable 0.00091 expected 0.00091
admin cumulativeInterest 0.00000 expected 0.00000
admin lDebt 81.00000
admin rInterest 0.00000
admin sInternalAmount 810.00091
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 9.00000 expected 9.00000
customer2 receivedSavings 9.00010 expected 9.00010
customer2 interestPayable 0.00010 expected 0.00010
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 9.00000
customer2 rInterest 0.00000
```

```
customer2 sInternalAmount 90.00010
rToken.transfer 10 customer1 -> customer3: started
rToken.transfer 10 customer1 -> customer3: done, gas used 559236, gas price 20 Gwei
customer1 tokenBalance 80.00000 expected 80.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
admin tokenBalance 0.00000 expected 0.00000
admin receivedLoan 81.00000 expected 81.00000
admin receivedSavings 81.00092 expected 81.00092
admin interestPayable 0.00092 expected 0.00092
admin cumulativeInterest 0.00000 expected 0.00000
admin lDebt 81.00000
admin rInterest 0.00000
admin sInternalAmount 810.00091
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 9.00000 expected 9.00000
customer2 receivedSavings 9.00010 expected 9.00010
customer2 interestPayable 0.00010 expected 0.00010
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 9.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 90.00010
customer3 tokenBalance 10.00000 expected 10.00000
customer3 receivedLoan 0.00000 expected 0.00000
customer3 receivedSavings 0.00000 expected 0.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 0.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 0.00000
rToken.payInterest to admin: started
rToken.payInterest to admin: done, gas used 162778, gas price 20 Gwei
customer1 tokenBalance 80.00000 expected 80.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
admin tokenBalance 0.00093 expected 0.00093
admin receivedLoan 81.00000 expected 81.00000
admin receivedSavings 81.00093 expected 81.00093
admin interestPayable 0.00000 expected 0.00000
admin cumulativeInterest 0.00093 expected 0.00093
admin lDebt 81.00000
admin rInterest 0.00093
admin sInternalAmount 810.00091
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 9.00000 expected 9.00000
customer2 receivedSavings 9.00010 expected 9.00010
customer2 interestPayable 0.00010 expected 0.00010
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 9.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 90.00010
customer3 tokenBalance 10.00000 expected 10.00000
customer3 receivedLoan 0.00000 expected 0.00000
customer3 receivedSavings 0.00000 expected 0.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 0.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 0.00000
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
customer1 tokenBalance 80.00000 expected 80.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
admin tokenBalance 0.00093 expected 0.00093
admin receivedLoan 81.00000 expected 81.00000
admin receivedSavings 81.00183 expected 81.00183
admin interestPayable 0.00090 expected 0.00090
admin cumulativeInterest 0.00093 expected 0.00093
admin lDebt 81.00000
admin rInterest 0.00093
admin sInternalAmount 810.00091
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 9.00000 expected 9.00000
customer2 receivedSavings 9.00020 expected 9.00020
customer2 interestPayable 0.00020 expected 0.00020
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 9.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 90.00010
customer3 tokenBalance 10.00000 expected 10.00000
customer3 receivedLoan 0.00000 expected 0.00000
customer3 receivedSavings 0.00000 expected 0.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 0.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 0.00000
rToken.transfer 10 customer3 -> customer1: started
rToken.transfer 10 customer3 -> customer1: done, gas used 368606, gas price 20 Gwei
customer1 tokenBalance 85.00000 expected 85.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
admin tokenBalance 0.00093 expected 0.00093
admin receivedLoan 81.00000 expected 81.00000
admin receivedSavings 81.00184 expected 81.00184
admin interestPayable 0.00091 expected 0.00091
admin cumulativeInterest 0.00093 expected 0.00093
admin lDebt 81.00000
admin rInterest 0.00093
admin sInternalAmount 810.00091
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 9.00000 expected 9.00000
customer2 receivedSavings 9.00020 expected 9.00020
customer2 interestPayable 0.00020 expected 0.00020
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 9.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 90.00010
customer3 tokenBalance 5.00000 expected 5.00000
customer3 receivedLoan 0.00000 expected 0.00000
customer3 receivedSavings 0.00000 expected 0.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 0.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 0.00000
rToken.transfer 10 customer1 -> customer2: started
rToken.transfer 10 customer1 -> customer2: done, gas used 613890, gas price 20 Gwei
customer1 tokenBalance 65.00000 expected 65.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
admin tokenBalance 0.00093 expected 0.00093
admin receivedLoan 81.00000 expected 81.00000
admin receivedSavings 81.00185 expected 81.00185
admin interestPayable 0.00092 expected 0.00092
admin cumulativeInterest 0.00093 expected 0.00093
admin lDebt 81.00000
admin rInterest 0.00093
admin sInternalAmount 810.00091
customer2 tokenBalance 20.00021 expected 20.00021
customer2 receivedLoan 9.00000 expected 9.00000
customer2 receivedSavings 9.00021 expected 9.00021
customer2 interestPayable 0.00000 expected 0.00000
```

```
customer2 cumulativeInterest 0.00021 expected 0.00021
customer2 lDebt 9.00000
customer2 rInterest 0.00021
customer2 sInternalAmount 90.00010
customer3 tokenBalance 5.00000 expected 5.00000
customer3 receivedLoan 0.00000 expected 0.00000
customer3 receivedSavings 0.00000 expected 0.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 0.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 0.00000
rToken.transferAll customer2 -> customer1: started
rToken.transferAll customer2 -> customer1: done, gas used 409568, gas price 20 Gwei
customer1 tokenBalance 85.00021 expected 85.00021
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
admin tokenBalance 0.00093 expected 0.00093
admin receivedLoan 81.00019 expected 81.00019
admin receivedSavings 81.00204 expected 81.00204
admin interestPayable 0.00093 expected 0.00093
admin cumulativeInterest 0.00093 expected 0.00093
admin lDebt 81.00019
admin rInterest 0.00093
admin sInternalAmount 810.00276
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 9.00002 expected 9.00002
customer2 receivedSavings 9.00002 expected 9.00002
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00021 expected 0.00021
customer2 lDebt 9.00002
customer2 rInterest 0.00000
customer2 sInternalAmount 89.99825
customer3 tokenBalance 5.00000 expected 5.00000
customer3 receivedLoan 0.00000 expected 0.00000
customer3 receivedSavings 0.00000 expected 0.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 0.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 0.00000
     ✓ #3 normal operations with hat (7928ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 10 to customer1: started
rToken.mint 10 to customer1: done, gas used 432940, gas price 20 Gwei
customer1 tokenBalance 10.00000 expected 10.00000
customer1 receivedLoan 10.00000 expected 10.00000
customer1 receivedSavings 10.00000 expected 10.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 10.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 100.00000
rToken.mint 5 to customer1: started
rToken.mint 5 to customer1: done, gas used 289378, gas price 20 Gwei
customer1 tokenBalance 15.00000 expected 15.00000
customer1 receivedLoan 15.00000 expected 15.00000
customer1 receivedSavings 15.00000 expected 15.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 15.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 150.00000
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56677, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10007 Token
customer1 tokenBalance 15.00000 expected 15.00000
customer1 receivedLoan 15.00000 expected 15.00000
customer1 receivedSavings 15.01000 expected 15.01000
customer1 interestPayable 0.01000 expected 0.01000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 15.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 150.00000
     ✓ #4 mint multiple times (2002ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381024, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 200 to customer1 with a hat benefiting customer1(10%) and customer2(90%): started
rToken.mint 200 to customer1 with a hat benefiting customer1(10%) and customer2(90%): done, gas used 711396, gas price 20 Gwei
rToken.transfer 100 customer1 -> customer2: started
rToken.transfer 100 customer1 -> customer2: done, gas used 555183, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 20.00000 expected 20.00000
customer1 receivedSavings 20.00000 expected 20.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 20.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 200.00000
customer2 tokenBalance 100.00000 expected 100.00000
customer2 receivedLoan 180.00000 expected 180.00000
customer2 receivedSavings 180.00000 expected 180.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 180.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 1800.00000
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 20.00000 expected 20.00000
customer1 receivedSavings 20.00010 expected 20.00010
customer1 interestPayable 0.00010 expected 0.00010
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 20.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 200.00000
customer2 tokenBalance 100.00000 expected 100.00000
customer2 receivedLoan 180.00000 expected 180.00000
```

```
customer2 receivedSavings 180.00090 expected 180.00090
customer2 interestPayable 0.00090 expected 0.00090
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 180.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 1800.00000
rToken.payInterest to customer2: started
rToken.payInterest to customer2: done, gas used 147778, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 20.00000 expected 20.00000
customer1 receivedSavings 20.00010 expected 20.00010
customer1 interestPayable 0.00010 expected 0.00010
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 20.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 200.00000
customer2 tokenBalance 100.00091 expected 100.00091
customer2 receivedLoan 180.00000 expected 180.00000
customer2 receivedSavings 180.00091 expected 180.00091
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00091 expected 0.00091
customer2 lDebt 180.00000
customer2 rInterest 0.00091
customer2 sInternalAmount 1800.00000
rToken.redeem 100.00091 for customer2: started
rToken.redeem 100.00091 for customer2: done, gas used 405835, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 10.00000 expected 10.00000
customer1 receivedSavings 10.00010 expected 10.00010
customer1 interestPayable 0.00010 expected 0.00010
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 10.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 100.00051
customer2 tokenBalance 0.00002 expected 0.00002
customer2 receivedLoan 90.00000 expected 90.00000
customer2 receivedSavings 90.00002 expected 90.00002
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00093 expected 0.00093
customer2 lDebt 90.00000
customer2 rInterest 0.00002
customer2 sInternalAmount 899.99554
     ✓ #5 redeem all including paid interest from single hat (3413ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
rToken.createHat for customer1 benefiting admin and customer3 10/90: started
rToken.createHat for customer1 benefiting admin and customer3 10/90: done, gas used 209663, gas price 20 Gwei
rToken.createHat for customer2 benefiting admin and customer4 20/80: started
rToken.createHat for customer2 benefiting admin and customer4 20/80: done, gas used 209663, gas price 20 Gwei
rToken.createHat but not using it: started
rToken.createHat but not using it: done, gas used 158337, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1: started
rToken.mint 100 to customer1: done, gas used 531740, gas price 20 Gwei
admin tokenBalance 0.00000 expected 0.00000
admin receivedLoan 20.00000 expected 20.00000
admin receivedSavings 20.00000 expected 20.00000
admin interestPayable 0.00000 expected 0.00000
admin cumulativeInterest 0.00000 expected 0.00000
admin lDebt 20.00000
admin rInterest 0.00000
admin sInternalAmount 200.00000
customer1 tokenBalance 200.00000 expected 200.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
customer3 tokenBalance 0.00000 expected 0.00000
customer3 receivedLoan 180.00000 expected 180.00000
customer3 receivedSavings 180.00000 expected 180.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 180.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 1800.00000
customer4 tokenBalance 0.00000 expected 0.00000
customer4 receivedLoan 0.00000 expected 0.00000
customer4 receivedSavings 0.00000 expected 0.00000
customer4 interestPayable 0.00000 expected 0.00000
customer4 cumulativeInterest 0.00000 expected 0.00000
customer4 lDebt 0.00000
customer4 rInterest 0.00000
customer4 sInternalAmount 0.00000
rToken.transfer 100 from customer1 to customer 2: started
rToken.transfer 100 from customer1 to customer 2: done, gas used 511418, gas price 20 Gwei
admin tokenBalance 0.00000 expected 0.00000
admin receivedLoan 30.00000 expected 30.00000
admin receivedSavings 30.00000 expected 30.00000
admin interestPayable 0.00000 expected 0.00000
admin cumulativeInterest 0.00000 expected 0.00000
admin lDebt 30.00000
admin rInterest 0.00000
admin sInternalAmount 300.00000
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
customer2 tokenBalance 100.00000 expected 100.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
customer3 tokenBalance 0.00000 expected 0.00000
customer3 receivedLoan 90.00000 expected 90.00000
customer3 receivedSavings 90.00000 expected 90.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 90.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 900.00000
customer4 tokenBalance 0.00000 expected 0.00000
customer4 receivedLoan 80.00000 expected 80.00000
customer4 receivedSavings 80.00000 expected 80.00000
customer4 interestPayable 0.00000 expected 0.00000
customer4 cumulativeInterest 0.00000 expected 0.00000
customer4 lDebt 80.00000
customer4 rInterest 0.00000
customer4 sInternalAmount 800.00000
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
```

```
admin tokenBalance 0.00000 expected 0.00000
admin receivedLoan 30.00000 expected 30.00000
admin receivedSavings 30.00015 expected 30.00015
admin interestPayable 0.00015 expected 0.00015
admin cumulativeInterest 0.00000 expected 0.00000
admin lDebt 30.00000
admin rInterest 0.00000
admin sInternalAmount 300.00000
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
customer2 tokenBalance 100.00000 expected 100.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
customer3 tokenBalance 0.00000 expected 0.00000
customer3 receivedLoan 90.00000 expected 90.00000
customer3 receivedSavings 90.00045 expected 90.00045
customer3 interestPayable 0.00045 expected 0.00045
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 90.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 900.00000
customer4 tokenBalance 0.00000 expected 0.00000
customer4 receivedLoan 80.00000 expected 80.00000
customer4 receivedSavings 80.00040 expected 80.00040
customer4 interestPayable 0.00040 expected 0.00040
customer4 cumulativeInterest 0.00000 expected 0.00000
customer4 lDebt 80.00000
customer4 rInterest 0.00000
customer4 sInternalAmount 800.00000
     ✓ #6 transfer and switch hats (4206ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.transfer 100 from customer 1 to customer 2: started
token.transfer 100 from customer 1 to customer 2: done, gas used 51162, gas price 20 Gwei
token.approve 100 by customer2: started
token.approve 100 by customer2: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer2: started
rToken.mint 100 to customer2: done, gas used 402952, gas price 20 Gwei
customer2 tokenBalance 100.00000 expected 100.00000
customer2 receivedLoan 100.00000 expected 100.00000
customer2 receivedSavings 100.00000 expected 100.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 100.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 1000.00000
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1: started
rToken.mint 100 to customer1: done, gas used 319390, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00000 expected 100.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 1000.00000
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
rToken.payInterest to customer1: started
rToken.payInterest to customer1: done, gas used 147778, gas price 20 Gwei
customer1 tokenBalance 100.00051 expected 100.00051
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00051 expected 100.00051
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00051 expected 0.00051
customer1 lDebt 100.00000
customer1 rInterest 0.00051
customer1 sInternalAmount 1000.00000
customer2 tokenBalance 100.00000 expected 100.00000
customer2 receivedLoan 100.00000 expected 100.00000
customer2 receivedSavings 100.00051 expected 100.00051
customer2 interestPayable 0.00051 expected 0.00051
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 100.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 1000.00000
rToken.redeem all for customer1: started
rToken.redeem all for customer1: done, gas used 318516, gas price 20 Gwei
customer1 tokenBalance 0.00000 expected 0.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00051 expected 0.00051
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1: started
rToken.mint 100 to customer1: done, gas used 308590, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00000 expected 100.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00051 expected 0.00051
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 999.99470
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
rToken.payInterest to customer1: started
rToken.payInterest to customer1: done, gas used 117778, gas price 20 Gwei
customer1 tokenBalance 100.00051 expected 100.00051
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00051 expected 100.00051
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00102 expected 0.00102
customer1 lDebt 100.00000
customer1 rInterest 0.00051
customer1 sInternalAmount 999.99470
rToken.transfer all from customer1 to customer2: started
rToken.transfer all from customer1 to customer2: done, gas used 365801, gas price 20 Gwei
customer1 tokenBalance 0.00000 expected 0.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00102 expected 0.00102
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
customer2 tokenBalance 200.00155 expected 200.00155
customer2 receivedLoan 200.00051 expected 200.00051
customer2 receivedSavings 200.00155 expected 200.00155
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00104 expected 0.00104
```

```
customer2 lDebt 200.00051
customer2 rInterest 0.00104
customer2 sInternalAmount 1999.99470
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1: started
rToken.mint 100 to customer1: done, gas used 293590, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00000 expected 100.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00102 expected 0.00102
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 999.98950
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
rToken.payInterest to customer1: started
rToken.payInterest to customer1: done, gas used 132778, gas price 20 Gwei
customer1 tokenBalance 100.00034 expected 100.00034
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00034 expected 100.00034
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00136 expected 0.00136
customer1 lDebt 100.00000
customer1 rInterest 0.00034
customer1 sInternalAmount 999.98950
customer2 tokenBalance 200.00155 expected 200.00155
customer2 receivedLoan 200.00051 expected 200.00051
customer2 receivedSavings 200.00224 expected 200.00224
customer2 interestPayable 0.00069 expected 0.00069
customer2 cumulativeInterest 0.00104 expected 0.00104
customer2 lDebt 200.00051
customer2 rInterest 0.00104
customer2 sInternalAmount 1999.99470
     ✓ #7 redeem all including paid interest from zero hatter (5890ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
     ✓ #8 special hats (43ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30941, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44049, gas price 20 Gwei
rToken.mintWithSelectedHat 100 to customer1 with the self hat: started
rToken.mintWithSelectedHat 100 to customer1 with the self hat: done, gas used 468314, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00000 expected 100.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 1000.00000
rToken.transfer all from customer1 to customer2: started
rToken.transfer all from customer1 to customer2: done, gas used 364933, gas price 20 Gwei
customer1 tokenBalance 80.00000 expected 80.00000
customer1 receivedLoan 80.00000 expected 80.00000
customer1 receivedSavings 80.00000 expected 80.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 80.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 800.00000
customer2 tokenBalance 20.00000 expected 20.00000
customer2 receivedLoan 20.00000 expected 20.00000
customer2 receivedSavings 20.00000 expected 20.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 20.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 200.00000
rToken changeHat for customer3 with selfhat: started
rToken changeHat for customer3 with selfhat: done, gas used 120415, gas price 20 Gwei
rToken.transfer all from customer1 to customer3: started
rToken.transfer all from customer1 to customer3: done, gas used 316419, gas price 20 Gwei
customer1 tokenBalance 60.00000 expected 60.00000
customer1 receivedLoan 60.00000 expected 60.00000
customer1 receivedSavings 60.00000 expected 60.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 60.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 600.00000
customer3 tokenBalance 20.00000 expected 20.00000
customer3 receivedLoan 20.00000 expected 20.00000
customer3 receivedSavings 20.00000 expected 20.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 20.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 200.00000
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
customer1 tokenBalance 60.00000 expected 60.00000
customer1 receivedLoan 60.00000 expected 60.00000
customer1 receivedSavings 60.00060 expected 60.00060
customer1 interestPayable 0.00060 expected 0.00060
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 60.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 600.00000
customer2 tokenBalance 20.00000 expected 20.00000
customer2 receivedLoan 20.00000 expected 20.00000
customer2 receivedSavings 20.00020 expected 20.00020
customer2 interestPayable 0.00020 expected 0.00020
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 20.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 200.00000
customer3 tokenBalance 20.00000 expected 20.00000
customer3 receivedLoan 20.00000 expected 20.00000
customer3 receivedSavings 20.00020 expected 20.00020
customer3 interestPayable 0.00020 expected 0.00020
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 20.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 200.00000
     ✓ #9 normal operations with self hatter (2989ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
```

```
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mintWithSelectedHat 100 to customer1 with the self hat: started
rToken.mintWithSelectedHat 100 to customer1 with the self hat: done, gas used 468314, gas price 20 Gwei
redeemUnderlying by admin: started
     ✓ #10 CompoundAs ownership protection (326ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381024, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mintWithSelectedHat 100 to customer1 with the self hat: started
rToken.mintWithSelectedHat 100 to customer1 with the self hat: done, gas used 468314, gas price 20 Gwei
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
rToken.transferAll from customer1 to customer2: started
rToken.transferAll from customer1 to customer2: done, gas used 414838, gas price 20 Gwei
customer1 tokenBalance 0.00000 expected 0.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00101 expected 0.00101
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
customer2 tokenBalance 100.00101 expected 100.00101
customer2 receivedLoan 100.00101 expected 100.00101
customer2 receivedSavings 100.00101 expected 100.00101
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 100.00101
customer2 rInterest 0.00000
customer2 sInternalAmount 1000.00000
     ✓ #11 transferAll (2216ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mintWithSelectedHat 100 to customer1 with the self hat: started
rToken.mintWithSelectedHat 100 to customer1 with the self hat: done, gas used 468314, gas price 20 Gwei
token.transfer 100 from customer 1 to customer 2: started
token.transfer 100 from customer 1 to customer 2: done, gas used 51162, gas price 20 Gwei
token.transfer 100 from customer 1 to customer 3: started
token.transfer 100 from customer 1 to customer 3: done, gas used 51162, gas price 20 Gwei
token.approve 100 by customer2: started
token.approve 100 by customer2: done, gas used 44061, gas price 20 Gwei
rToken.mintWithSelectedHat 100 to customer2 with customer1 as recipient: started
rToken.mintWithSelectedHat 100 to customer2 with customer1 as recipient: done, gas used 492729, gas price 20 Gwei
token.approve 100 by customer3: started
token.approve 100 by customer3: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer3 with zero hat: started
rToken.mint 100 to customer3 with zero hat: done, gas used 334390, gas price 20 Gwei
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 200.00000 expected 200.00000
customer1 receivedSavings 200.00067 expected 200.00067
customer1 interestPayable 0.00067 expected 0.00067
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 200.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 2000.00000
customer2 tokenBalance 100.00000 expected 100.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
rToken.redeem 50 for customer2: started
rToken.redeem 50 for customer2: done, gas used 322755, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 150.00000 expected 150.00000
customer1 receivedSavings 150.00067 expected 150.00067
customer1 interestPayable 0.00067 expected 0.00067
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 150.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 1500.00168
customer2 tokenBalance 50.00000 expected 50.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
rToken.redeemAll for customer1: started
rToken.redeemAll for customer1: done, gas used 347706, gas price 20 Gwei
customer1 tokenBalance 0.00000 expected 0.00000
customer1 receivedLoan 50.00000 expected 50.00000
customer1 receivedSavings 50.00000 expected 50.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00068 expected 0.00068
customer1 lDebt 50.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 499.99830
customer2 tokenBalance 50.00000 expected 50.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
```

```
customer2 sInternalAmount 0.00000
rToken.redeemAll for customer2: started
rToken.redeemAll for customer2: done, gas used 248302, gas price 20 Gwei
customer1 tokenBalance 0.00000 expected 0.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00068 expected 0.00068
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00003
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
customer3 tokenBalance 100.00000 expected 100.00000
customer3 receivedLoan 100.00000 expected 100.00000
customer3 receivedSavings 100.00035 expected 100.00035
customer3 interestPayable 0.00035 expected 0.00035
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 100.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 1000.00000
     ✓ #12 redeem all including paid interest from multiple hats (4088ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381024, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mintWithSelectedHat 100 to customer1 with the self hat: started
rToken.mintWithSelectedHat 100 to customer1 with the self hat: done, gas used 468314, gas price 20 Gwei
token.approve customer 2 by customer1: started
token.approve customer 2 by customer1: done, gas used 46167, gas price 20 Gwei
rToken transferFrom customer1 -> customer3 by customer2 more than approved: started
rToken transferFrom customer1 -> customer3 by customer2 all approved: started
rToken transferFrom customer1 -> customer3 by customer2 all approved: done, gas used 356611, gas price 20 Gwei
token.approve customer 2 by customer1: started
token.approve customer 2 by customer1: done, gas used 46179, gas price 20 Gwei
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
rToken transferAllFrom customer1 -> customer3 by customer2: started
rToken transferAllFrom customer1 -> customer3 by customer2: done, gas used 396937, gas price 20 Gwei
customer1 tokenBalance 0.00000 expected 0.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00051 expected 0.00051
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
customer3 tokenBalance 100.00101 expected 100.00101
customer3 receivedLoan 100.00051 expected 100.00051
customer3 receivedSavings 100.00101 expected 100.00101
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00051 expected 0.00051
customer3 lDebt 100.00051
customer3 rInterest 0.00051
customer3 sInternalAmount 1000.00000
     ✓ #13 approve & transferFrom & transferAllFrom (2320ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337242, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mintWithSelectedHat 100 to customer1 with the self hat: started
rToken.mintWithSelectedHat 100 to customer1 with the self hat: done, gas used 468314, gas price 20 Gwei
token.transfer 100 from customer 1 to customer 2: started
token.transfer 100 from customer 1 to customer 2: done, gas used 51162, gas price 20 Gwei
token.approve 100 by customer2: started
token.approve 100 by customer2: done, gas used 44061, gas price 20 Gwei
rToken.mintWithSelectedHat 100 to customer2 with the self hat: started
rToken.mintWithSelectedHat 100 to customer2 with the self hat: done, gas used 339752, gas price 20 Gwei
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
rToken.redeemAndTransferAll for customer1: started
rToken.redeemAndTransferAll for customer1: done, gas used 363282, gas price 20 Gwei
customer1 tokenBalance 0.00000 expected 0.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00051 expected 0.00051
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
     ✓ #14 redeemAndTransferAll (1903ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
rToken.updateCode: started
rToken.updateCode: done, gas used 33737, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1: started
rToken.mint 100 to customer1: done, gas used 417952, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00000 expected 100.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
```

```
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 1000.00000
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00100 expected 100.00100
customer1 interestPayable 0.00100 expected 0.00100
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 1000.00000
rToken.redeem 10 to customer1: started
rToken.redeem 10 to customer1: done, gas used 343938, gas price 20 Gwei
customer1 tokenBalance 90.00101 expected 90.00101
customer1 receivedLoan 90.00000 expected 90.00000
customer1 receivedSavings 90.00101 expected 90.00101
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00101 expected 0.00101
customer1 lDebt 90.00000
customer1 rInterest 0.00101
customer1 sInternalAmount 900.00101
rToken.payInterest to customer1: started
rToken.payInterest to customer1: done, gas used 117778, gas price 20 Gwei
customer1 tokenBalance 90.00102 expected 90.00102
customer1 receivedLoan 90.00000 expected 90.00000
customer1 receivedSavings 90.00102 expected 90.00102
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00102 expected 0.00102
customer1 lDebt 90.00000
customer1 rInterest 0.00102
customer1 sInternalAmount 900.00101
rToken.payInterest to customer1 again: started
rToken.payInterest to customer1 again: done, gas used 117778, gas price 20 Gwei
customer1 tokenBalance 90.00103 expected 90.00103
customer1 receivedLoan 90.00000 expected 90.00000
customer1 receivedSavings 90.00103 expected 90.00103
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00103 expected 0.00103
customer1 lDebt 90.00000
customer1 rInterest 0.00103
customer1 sInternalAmount 900.00101
rToken.redeem 2 of customer1 to customer3: started
rToken.redeem 2 of customer1 to customer3: done, gas used 327931, gas price 20 Gwei
     ✓ #15 upgrade contract (2406ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
change owner to customer1: started
change owner to customer1: done, gas used 32662, gas price 20 Gwei
change owner to admin: started
change owner to admin: done, gas used 32662, gas price 20 Gwei
rTokenLogic.initialize first time: started
rTokenLogic.initialize first time: done, gas used 230136, gas price 20 Gwei
rTokenLogic.initialize second time: started
rTokenLogic.updateCode from non-owner: started
rTokenLogic renounceOwnership: started
rTokenLogic renounceOwnership: done, gas used 30347, gas price 20 Gwei
rToken.initialize (original): started
rToken.updateCode (original): started
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
rToken.updateCode: started
rToken.updateCode: done, gas used 33737, gas price 20 Gwei
rTokenLogic.initialize first time: started
rTokenLogic.initialize first time: done, gas used 230136, gas price 20 Gwei
rTokenLogic.initialize second time: started
rTokenLogic.updateCode from non-owner: started
rToken.initialize (original): started
rToken.updateCode (original): started
     ✓ #16 proxy security (742ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1: started
rToken.mint 100 to customer1: done, gas used 417952, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00000 expected 100.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 1000.00000
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00100 expected 100.00100
customer1 interestPayable 0.00100 expected 0.00100
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 1000.00000
rToken.redeem 10 to customer1: started
rToken.redeem 10 to customer1: done, gas used 343938, gas price 20 Gwei
customer1 tokenBalance 90.00101 expected 90.00101
customer1 receivedLoan 90.00000 expected 90.00000
customer1 receivedSavings 90.00101 expected 90.00101
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00101 expected 0.00101
customer1 lDebt 90.00000
customer1 rInterest 0.00101
customer1 sInternalAmount 900.00101
rToken.payInterest to customer1: started
rToken.payInterest to customer1: done, gas used 117778, gas price 20 Gwei
customer1 tokenBalance 90.00102 expected 90.00102
customer1 receivedLoan 90.00000 expected 90.00000
customer1 receivedSavings 90.00102 expected 90.00102
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00102 expected 0.00102
customer1 lDebt 90.00000
customer1 rInterest 0.00102
customer1 sInternalAmount 900.00101
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
rToken.updateCode: started
```

```
rToken.updateCode: done, gas used 33737, gas price 20 Gwei
rToken.payInterest to customer1 again: started
rToken.payInterest to customer1 again: done, gas used 117778, gas price 20 Gwei
customer1 tokenBalance 90.00105 expected 90.00105
customer1 receivedLoan 90.00000 expected 90.00000
customer1 receivedSavings 90.00105 expected 90.00105
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00105 expected 0.00105
customer1 lDebt 90.00000
customer1 rInterest 0.00105
customer1 sInternalAmount 900.00101
rToken.redeem 2 of customer1 to customer3: started
rToken.redeem 2 of customer1 to customer3: done, gas used 327931, gas price 20 Gwei
     ✓ #17 storage continuity during upgrade (2367ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1 with a hat benefiting admin(90%) and customer2(10%): started
rToken.mint 100 to customer1 with a hat benefiting admin(90%) and customer2(10%): done, gas used 711351, gas price 20 Gwei
rToken.changeHatFor by customer1: started
rToken.changeHatFor by customer1: started
rToken.changeHatFor by customer1: done, gas used 61819, gas price 20 Gwei
rToken.changeHatFor by customer1: started
     ✓ #18 admin.changeHatFor (387ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1 with a sombreror: started
rToken.mint 100 to customer1 with a sombreror: done, gas used 5957762, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
rToken.transfer 10 customer1 -> customer2: started
rToken.transfer 10 customer1 -> customer2: done, gas used 4961403, gas price 20 Gwei
normal transfer tx cost 4961403
customer1 tokenBalance 90.00000 expected 90.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
customer2 tokenBalance 10.00000 expected 10.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
rToken.transfer 10 customer1 -> customer2 again: started
rToken.transfer 10 customer1 -> customer2 again: done, gas used 4050768, gas price 20 Gwei
Same hat transfer tx cost 4050768
rToken.createHat by bigger sombrero: started
token.transfer 100 from customer 1 to customer 2: started
token.transfer 100 from customer 1 to customer 2: done, gas used 51162, gas price 20 Gwei
token.approve 100 by customer3: started
token.approve 100 by customer3: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer3 with a smaller sombrero: started
rToken.mint 100 to customer3 with a smaller sombrero: done, gas used 4403972, gas price 20 Gwei
     ✓ #19 Max hat numbers & same hat optimization (6023ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
     ✓ #20 change hat with invalid hat ID should fail
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
     ✓ #21 create invalid hats (134ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337254, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
```

```
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381000, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381024, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
compoundAS2.transferOwnership: started
compoundAS2.transferOwnership: done, gas used 30953, gas price 20 Gwei
compoundAS3.transferOwnership: started
compoundAS3.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.transfer 100 from customer 1 to customer 2: started
token.transfer 100 from customer 1 to customer 2: done, gas used 51162, gas price 20 Gwei
token.approve 100 by customer2: started
token.approve 100 by customer2: done, gas used 44061, gas price 20 Gwei
cToken.mint 100 to customer2: started
cToken.mint 100 to customer2: done, gas used 132962, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1: started
rToken.mint 100 to customer1: done, gas used 394390, gas price 20 Gwei
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 100 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
change to the same allocation strategy: started
change to the same allocation strategy: done, gas used 269995, gas price 20 Gwei
change allocation strategy 1st time: started
change allocation strategy 1st time: done, gas used 347768, gas price 20 Gwei
change allocation strategy 2nd time: started
change allocation strategy 2nd time: done, gas used 313568, gas price 20 Gwei
token.transfer 100 from customer 1 to customer 2: started
token.transfer 100 from customer 1 to customer 2: done, gas used 51162, gas price 20 Gwei
token.approve 100 by customer2: started
token.approve 100 by customer2: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer2: started
rToken.mint 100 to customer2: done, gas used 304390, gas price 20 Gwei
rToken.redeemAll to customer1: started
rToken.redeemAll to customer1: done, gas used 344316, gas price 20 Gwei
     ✓ #22 change allocation strategy multiple times (4372ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381024, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1 with a hat benefiting customer2(100%): started
rToken.mint 100 to customer1 with a hat benefiting customer2(100%): done, gas used 606291, gas price 20 Gwei
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 100.00000 expected 100.00000
customer2 receivedSavings 100.00000 expected 100.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 100.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 1000.00000
rToken.createHat for customer1 with a hat benefiting customer3(100%): started
rToken.createHat for customer1 with a hat benefiting customer3(100%): done, gas used 338904, gas price 20 Gwei
customer3 tokenBalance 0.00000 expected 0.00000
customer3 receivedLoan 100.00000 expected 100.00000
customer3 receivedSavings 100.00000 expected 100.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 100.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 1000.00000
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
     ✓ #23 change hat test (1426ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mint 100 to customer1 with a hat benefiting customer2(100%): started
rToken.mint 100 to customer1 with a hat benefiting customer2(100%): done, gas used 606291, gas price 20 Gwei
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 100.00000 expected 100.00000
customer2 receivedSavings 100.00000 expected 100.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 100.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 1000.00000
rToken.createHat for customer1 with a hat benefiting customer3(100%): started
rToken.createHat for customer1 with a hat benefiting customer3(100%): done, gas used 443963, gas price 20 Gwei
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
customer3 tokenBalance 0.00000 expected 0.00000
customer3 receivedLoan 90.00000 expected 90.00000
customer3 receivedSavings 90.00000 expected 90.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 90.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 900.00000
customer4 tokenBalance 0.00000 expected 0.00000
customer4 receivedLoan 10.00000 expected 10.00000
customer4 receivedSavings 10.00000 expected 10.00000
customer4 interestPayable 0.00000 expected 0.00000
customer4 cumulativeInterest 0.00000 expected 0.00000
customer4 lDebt 10.00000
customer4 rInterest 0.00000
customer4 sInternalAmount 100.00000
     ✓ #23 change hat test w/ 2 recipients (1262ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
```

```
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381024, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950114, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
rToken.createHat for customer1 benefiting admin and customer3 10/90: started
rToken.createHat for customer1 benefiting admin and customer3 10/90: done, gas used 209663, gas price 20 Gwei
token.approve 1000 by customer1: started
token.approve 1000 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mintWithSelectedHat 1000 to customer1 with the first hat: started
rToken.mintWithSelectedHat 1000 to customer1 with the first hat: done, gas used 525751, gas price 20 Gwei
admin tokenBalance 0.00000 expected 0.00000
admin receivedLoan 100.00000 expected 100.00000
admin receivedSavings 100.00000 expected 100.00000
admin interestPayable 0.00000 expected 0.00000
admin cumulativeInterest 0.00000 expected 0.00000
admin lDebt 100.00000
admin rInterest 0.00000
admin sInternalAmount 1000.00000
customer1 tokenBalance 1000.00000 expected 1000.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
customer3 tokenBalance 0.00000 expected 0.00000
customer3 receivedLoan 900.00000 expected 900.00000
customer3 receivedSavings 900.00000 expected 900.00000
customer3 interestPayable 0.00000 expected 0.00000
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 900.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 9000.00000
customer4 tokenBalance 0.00000 expected 0.00000
customer4 receivedLoan 0.00000 expected 0.00000
customer4 receivedSavings 0.00000 expected 0.00000
customer4 interestPayable 0.00000 expected 0.00000
customer4 cumulativeInterest 0.00000 expected 0.00000
customer4 lDebt 0.00000
customer4 rInterest 0.00000
customer4 sInternalAmount 0.00000
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 167630, gas price 20 Gwei
Wait for 142 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
admin tokenBalance 0.00000 expected 0.00000
admin receivedLoan 100.00000 expected 100.00000
admin receivedSavings 100.00014 expected 100.00014
admin interestPayable 0.00014 expected 0.00014
admin cumulativeInterest 0.00000 expected 0.00000
admin lDebt 100.00000
admin rInterest 0.00000
admin sInternalAmount 1000.00000
customer1 tokenBalance 1000.00000 expected 1000.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
customer3 tokenBalance 0.00000 expected 0.00000
customer3 receivedLoan 900.00000 expected 900.00000
customer3 receivedSavings 900.00128 expected 900.00128
customer3 interestPayable 0.00128 expected 0.00128
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 900.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 9000.00000
customer4 tokenBalance 0.00000 expected 0.00000
customer4 receivedLoan 0.00000 expected 0.00000
customer4 receivedSavings 0.00000 expected 0.00000
customer4 interestPayable 0.00000 expected 0.00000
customer4 cumulativeInterest 0.00000 expected 0.00000
customer4 lDebt 0.00000
customer4 rInterest 0.00000
customer4 sInternalAmount 0.00000
rToken.createHat for customer1 benefiting customer2 and customer4 49/51: started
rToken.createHat for customer1 benefiting customer2 and customer4 49/51: done, gas used 503620, gas price 20 Gwei
Before binge borrowing: 1 cToken = 0.10000 Token
cToken.borrow 10 to bingeBorrower: started
cToken.borrow 10 to bingeBorrower: done, gas used 110405, gas price 20 Gwei
Wait for 298 blocks...
cToken.accrueInterest: started
cToken.accrueInterest: done, gas used 56687, gas price 20 Gwei
After binge borrowing: 1 cToken = 0.10000 Token
admin tokenBalance 0.00000 expected 0.00000
admin receivedLoan 0.00000 expected 0.00000
admin receivedSavings 0.00014 expected 0.00014
admin interestPayable 0.00014 expected 0.00014
admin cumulativeInterest 0.00000 expected 0.00000
admin lDebt 0.00000
admin rInterest 0.00000
admin sInternalAmount 0.00143
customer1 tokenBalance 1000.00000 expected 1000.00000
customer1 receivedLoan 0.00000 expected 0.00000
customer1 receivedSavings 0.00000 expected 0.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 0.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 0.00000
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 490.00000 expected 490.00000
customer2 receivedSavings 490.00293 expected 490.00293
customer2 interestPayable 0.00293 expected 0.00293
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 490.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 4899.99299
customer3 tokenBalance 0.00000 expected 0.00000
customer3 receivedLoan 0.00000 expected 0.00000
customer3 receivedSavings 0.00129 expected 0.00129
customer3 interestPayable 0.00129 expected 0.00129
customer3 cumulativeInterest 0.00000 expected 0.00000
customer3 lDebt 0.00000
customer3 rInterest 0.00000
customer3 sInternalAmount 0.01287
customer4 tokenBalance 0.00000 expected 0.00000
customer4 receivedLoan 510.00000 expected 510.00000
customer4 receivedSavings 510.00305 expected 510.00305
customer4 interestPayable 0.00304 expected 0.00304
customer4 cumulativeInterest 0.00000 expected 0.00000
customer4 lDebt 510.00000
customer4 rInterest 0.00000
customer4 sInternalAmount 5099.99271
     ✓ #24 complex functional test (5631ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
```

```
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381012, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381024, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337290, gas price 20 Gwei
compoundAS2.transferOwnership: started
compoundAS2.transferOwnership: done, gas used 30953, gas price 20 Gwei
rToken.createHat for customer1 benefiting admin and customer3 10/90: started
rToken.createHat for customer1 benefiting admin and customer3 10/90: done, gas used 209663, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mintWithSelectedHat 100 to customer1 with the first hat: started
rToken.mintWithSelectedHat 100 to customer1 with the first hat: done, gas used 540751, gas price 20 Gwei
rToken.redeem 100 of customer1 to customer3: started
rToken.redeem 100 of customer1 to customer3: done, gas used 196377, gas price 20 Gwei
     ✓ #25 change allocation strategy multiple times (1413ms)
ERC20Mintable.new: started
ERC20Mintable.new: done, gas used 801360, gas price 20 Gwei
token.mint 1000 -> customer1: started
token.mint 1000 -> customer1: done, gas used 66943, gas price 20 Gwei
ComptrollerMock.new: started
ComptrollerMock.new: done, gas used 753255, gas price 20 Gwei
InterestRateModelMock.new: started
InterestRateModelMock.new: done, gas used 107427, gas price 20 Gwei
CErc20.new: started
CErc20.new: done, gas used 4381036, gas price 20 Gwei
CompoundAllocationStrategy.new: started
CompoundAllocationStrategy.new: done, gas used 950126, gas price 20 Gwei
RToken.new: started
RToken.new: done, gas used 4196930, gas price 20 Gwei
Proxy.new: started
Proxy.new: done, gas used 337266, gas price 20 Gwei
compoundAS.transferOwnership: started
compoundAS.transferOwnership: done, gas used 30953, gas price 20 Gwei
token.approve 100 by customer1: started
token.approve 100 by customer1: done, gas used 44061, gas price 20 Gwei
rToken.mintWithSelectedHat 100 to customer1 with the self hat: started
rToken.mintWithSelectedHat 100 to customer1 with the self hat: done, gas used 606336, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00000 expected 100.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 1000.00000
rToken.transfer all from customer1 to customer2: started
rToken.transfer all from customer1 to customer2: done, gas used 463469, gas price 20 Gwei
customer1 tokenBalance 80.00000 expected 80.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00000 expected 100.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 1000.00000
customer2 tokenBalance 20.00000 expected 20.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
rToken.transfer all from customer2 back to customer1: started
rToken.transfer all from customer2 back to customer1: done, gas used 257835, gas price 20 Gwei
customer1 tokenBalance 100.00000 expected 100.00000
customer1 receivedLoan 100.00000 expected 100.00000
customer1 receivedSavings 100.00000 expected 100.00000
customer1 interestPayable 0.00000 expected 0.00000
customer1 cumulativeInterest 0.00000 expected 0.00000
customer1 lDebt 100.00000
customer1 rInterest 0.00000
customer1 sInternalAmount 1000.00000
customer2 tokenBalance 0.00000 expected 0.00000
customer2 receivedLoan 0.00000 expected 0.00000
customer2 receivedSavings 0.00000 expected 0.00000
customer2 interestPayable 0.00000 expected 0.00000
customer2 cumulativeInterest 0.00000 expected 0.00000
customer2 lDebt 0.00000
customer2 rInterest 0.00000
customer2 sInternalAmount 0.00000
     ✓ #27 normal operations with self-recipient hatter (2020ms)

  Contract: RTokenStorage
RTokenStorageLayoutTester.new: started
RTokenStorageLayoutTester.new: done, gas used 4202998, gas price 20 Gwei
     ✓ #0 validate immutable storage layout (114ms)


  29 passing (2m)


Compiled 20 contracts successfully


  Contract: gDai Staking
    deploying
       ✓ Reverts when reward token is zero (51ms)
       ✓ Reverts when rewards duration is zero
       ✓ Reverts when start is zero
       ✓ Reverts when staking rewards is zero
    start
       ✓ sets required reward rate on notify (154ms)
       ✓ reverts if called twice (118ms)
       ✓ reverts if start time is still in the future
       ✓ reverts if no tokens sent
      with real (not mock contract)
         ✓ can start issuing rewards (61ms)
    stake
       ✓ can stake and get rewards (101ms)
    stake and withdraw then claim rewards
       ✓ reverts if withdrawing nothing (38ms)
       ✓ reverts if staking for zero address
       ✓ reverts if trying to withdraw and have no stake (61ms)
       ✓ reverts if not called by the staking controller
       ✓ can stake and get rewards with a single withdrawal (100ms)
       ✓ can stake and get rewards over 2 withdrawals (210ms)
    lastTimeRewardApplicable()
       ✓ should return 0
      when updated
         ✓ should equal current timestamp
    getReward()
       ✓ should do nothing if no stake (44ms)
    getRewardForDuration()
       ✓ should increase rewards token balance (52ms)
    rewardPerToken()
       ✓ should return 0
       ✓ should be > 0 (90ms)


  22 passing (4s)
```

# Appendix

## File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

### Contracts

```
8a949ccac9a01f18a989deb63eaca62404eb1425d8891fa027348a35b7d484ca ./rDAI.sol
c10e33db65c732b9a655f06019a0e346b6eb81a9d1bf7b7c24e110fec56f0d32 ./rSAI.sol
9c5342d9d1e02d810dd1be669e3a469bc1ec0e7955f3432ae713b884c6ff83e1 ./CompoundAllocationStrategy.sol
ad66f56aff590f1443bf3180a0b40003a3a6305722dd3e3505a884a06a14bc7e ./CompoundAllocationStrategyV2.sol
4426a024417a5deb34e282d7101e156f463701c87c67fe5b744a0b386719b548 ./IAllocationStrategy.sol
2ccf735421be4b352716c61d0174700201d852258dd8cb9ea33ab43a3ff46d76 ./IRToken.sol
9ce24d139b88ce72247a0abb21e976697ae49250a8c70462aa068a75db6d8c02 ./IRTokenAdmin.sol
2e169692eed6eac3cf884addbdbca4471f2b42c5ab304f4ec8ee57cee6c2a3ed ./LibraryLock.sol
4f3a34b5894241ea4fd35db6f3038fbd4a23b0747f7115cfd8b53acddb9e9d33 ./Ownable.sol
9f1bc1f6baffbad577a93f0a0fbdc37374427d88fa8e7d44dbf459959f5d806e ./Proxiable.sol
847dcb160caa6cff695e63f6e2615e560cac0d009b2dfaf3210e70cfc565c6d3 ./Proxy.sol
8aa5b1bf84ee78ceb4f39bc0ffd51dc52f18752050bbe62d32d0ad4406cfcd6f ./ReentrancyGuard.sol
bde8f7637d013ad0927b531f3bb74426cfb3a7a9e6fae1da56c6043d4ed3f616 ./RToken.sol
c039919084d0eab5ddbef679d2bddb20dffa2777bef1dc8dd08a816a298c6494 ./RTokenStorage.sol
9985e5a95b8d5e64c26de7c1de4dbcf2eea64409d810ed1508b83bd44de87e84 ./RTokenStructs.sol
0a75893d9e09e005db4fd88850c3ea5751f9f2fd0968965c3f5c0b08ea23d511 ./contracts/gDaiStaking.sol
```

# Changelog

- 2020-10-20 - Initial report
- 2020-12-11 - Re-audit report [core: `2d94480`, rtoken-monorepo: `a30cb60`]

## About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure blockchain platforms at scale using computer-aided reasoning tools, with a mission to help boost the adoption of this exponentially growing technology.

With over 1000 Google scholar citations and numerous published papers, Quantstamp's team has decades of combined experience in formal verification, static analysis, and software verification. Quantstamp has also developed a protocol to help smart contract developers and projects worldwide to perform cost-effective smart contract security scans.

To date, Quantstamp has protected $5B in digital asset risk from hackers and assisted dozens of blockchain projects globally through its white glove security assessment services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Quantstamp's collaborations with leading academic institutions such as the National University of Singapore and MIT (Massachusetts Institute of Technology) reflect our commitment to research, development, and enabling world-class blockchain security.