Contract: Generate\_Credit\_Card\_Number

Getting started:

1. Go to remix.ethereum.org.

2. Click on the "+" button on the upper left corner of the screen.

3. Name the file name anything appropriate.

4. Copy-paste the code file's contents into the IDE.

Note: Make sure that the compiler version is set to "0.4.24+commit.e67f0147" before trying to run the code, else errors will come.

How to use:

1. On the “Deploy” button under the “Run” tab, you’ll notice how it has a text field beside it. Click on the “v” button for it to drop down.
2. Fill out the following fields:
   1. “\_commitPhaseLengthInSeconds” to 10 seconds or more. If it’s less than 10 seconds, the deployment won’t go through.
   2. “\_choice1” represents the first candidate. Fill it with a name.
   3. “\_choice2” represents the second candidate. Fill it with a name.
3. Once that’s filled out, click on the “Transact” button to deploy the contract.
4. Now, you’ll have to use keccak256 hashes in order to vote and keep track of it.
   1. Go to <https://emn178.github.io/online-tools/keccak_256.html>
   2. On the “Input” field, type in the first candidate by following this format, “1-[Insert first candidate name here]. Make sure it doesn’t contain any spaces if you typed in a full name. (Example: 1-Batman, 1-BruceWayne)
   3. If you’re voting for the second candidate, simply change the value to “2”. (Example: 2-Luthor, 2-LexLuthor)
   4. Below the field, it’ll automatically generate a hash. Copy that and place it in fields that require a “bytes32” input. Make sure to add “0x” in the beginning of the hash. (Example: 0x26c7c5c38eaeaa66c32b8621dd974e78783949751f893471c94b274d746b2357)
5. On the “CommitField” function, click on the “v” button and it’ll expand. In the “\_voteCommit” field, add the hash of the candidate of your choosing in the bytes32 field. Click on “Transact” to proceed. Note, this only commits the data during the committing period that you’ve set before.
6. On the “revealVote” function, click on the “v” button and it’ll expand. In the “\_vote” field, add the name of the candidate of your choosing under the same format as step 4b. For “\_voteCommit”, add the hash of the candidate of your choosing in the bytes32 field. Click on “Transact” to proceed. This will reveal the votes after the committing period is over.
7. Clicking on “choice1” displays the first candidate’s name.
8. Clicking on “choice2” displays the second candidate’s name.
9. Clicking on “commitPhaseEndTime” displays the time you have in long seconds.
10. Clicking on “getWinner” displays the winner with the most votes. It could also display a tie.
11. Clicking on “numberOfVotesCast” displays the overall number of votes placed between the candidates.
12. The function “voteCommits” verifies the vote by using the same hash that you used to vote for the candidate.
13. Clicking on “votesForChoice1” reveals the number of votes for the first candidate.
14. Clicking on “votesForChoice2” reveals the number of votes for the second candidate.

Context: The behavioral pattern commit-reveal is used to keep secret of private data during a period of time in the blockchain.