**Proof of Work Consensus Simulation and Competition**

**Pre-Instructions**

**Objective:** In this simulation, you will each act as a miner keeping your own record of the blockchain. You will store your copy of the blockchain in a Microsoft Word document and you will use the Hash Tool to mine blocks. When you have found a valid block, add it to the end of your Word document and attach that document to an email message sent to all the students in the class using Reply All to the email I send to you at the start of the simulation. Your objective is to beat all your classmates by mining more blocks than anyone else in 20 minutes or by the time the blockchain is 26 blocks long, whichever comes first. In case of a tie, the winner will be the student collecting the most in fees. The winner will receive an award.

**Block format:** Each block will consist of a header, the hash of the header, and a single transaction. As with bitcoin, the header will contain the block number, the hash of the previous block, a block validation, and a nonce. With bitcoin, the block validation consists of a Merkle tree of the transactions in the block, but in our simulation, we will replace the Merkle tree with the hash of the single transaction. Also, with bitcoin the address of the miner will appear in the first (coinbase) transaction. Since we will not have a coinbase transaction, the miners’ full name will also be included in the header.

**Header format:** The header will consist of the following items in this order with commas between them and no spaces (except, optionally, in the miner’s name): Block number, hash of previous block, hash of transaction in this block, miner’s name, and nonce. Here is an example of a header:  
3,006b3c8718f3ec171b62a086ab2d063ef31e1f01ab4eaa68aae1a56cc4a18c56, 7f8276bedb87d4489547909266569c8ae14669b283d38abd71bc206ccb6729ea,Steven Gordon,37  
Word breaks header into two lines, but for purposes of this simulation you should treat it as one line. To find the header’s hash, copy the header into the hash tool and backspace before 7f to make it one line. The hash should be 21681b7b9f15b3dcae61c4d2ebcbc596ccffe604db3f13e56229cca13278b03a. Check it out to make sure you know how to do this.

**Nonce:** You will need to select a nonce such that the hash of the header meets a certain condition. For bitcoin, the condition is, roughly, that the hash starts with a certain number of zeros. In the example above, the condition is that the third letter of the hash is a “6” (note that the hash of the previous block and the hash of the current block both meet this condition). I will not reveal the condition that results in a valid header until the time of the simulation to keep savvy users from writing a computer program to help them mine more quickly. You will have to find a nonce that creates a valid header by trial and error.

**Transactions:** You must select transactions from the mempool, the list of transactions proposed by your blockchain’s users along with the fee they are willing to pay to include that transaction in a block. In practice, users continually update the mempool as they create new transactions. For our simulation, the mempool is fixed, as given below:

Alligator1, Bear2, Camel3, Dolphin4, Elephant1, Frog2, Gorilla3, Horse4, Iguana1, Jackal2, Kangaroo3, Lion4, Mouse1, Narwhal2, Octopus3, Parrot4, Quail1, Robbin2, Skunk3, Turkey4, Unicorn1, Vulture2, Walrus3, Yak4, Zebra5, Ant5, Bunny5, Caribou5, Dog6, Eagle6, Fish6

Note that all mempool transactions begin with a capital letter. The numbers following the animal names represent the fee the user is willing to pay. The transaction itself is meaningless, but in the real world, miners do not really care what a transaction contains. They do have to determine that a transaction is valid, but we will be skipping that task. Of course, once a transaction appears on the blockchain, you must remove it from the mempool. Any block that includes a transaction already on the blockchain is invalid.

**A Sample Block:** Following is a sample of a valid block for the condition that the third letter of its hash is a “6”:

3,006b3c8718f3ec171b62a086ab2d063ef31e1f01ab4eaa68aae1a56cc4a18c56, 7f8276bedb87d4489547909266569c8ae14669b283d38abd71bc206ccb6729ea,Steven Gordon,37  
21681b7b9f15b3dcae61c4d2ebcbc596ccffe604db3f13e56229cca13278b03a  
Camel3

Note that Camel3 is an item from the mempool, that its hash (starting with 7f and ending in ea) appears just before my name, that the previous hash is valid according to the sample rule, and that the hash of the header is valid according to this rule. This is, presumably, block #3.

**Genesis Block:** I will not reveal the first block in the blockchain until the simulation starts.

**Consensus:** The default rule is to add blocks to the longest chain. When you receive a message containing a new chain, this will be a longer than the chain you had previously used, so the hash of the previous block will have changed. This means you will need to start mining anew. Of course, you are free to ignore expectations and continue to mine on the old, shorter blockchain. However, when the simulation is over, I will consider the longest chain to be the valid one as long its last four blocks are valid. So, if you continue to build on a short chain while others are building on the longer chain, any work you’ve done to add blocks to that short chain will be worthless unless you can make your chain longer than theirs.

In the real world, when two miners create new blocks at about the same time, some miners will see one block first and some will see the other. Miners will not necessarily know which blockchain to mine on. To simulate this condition, the email list I send each student will reach only 1/3 of the class and me. When I receive an email with a blockchain, I will rebroadcast it to the entire class after a short delay. When you receive a message from me and from someone else, if they are the same, no problem. If they are different, you will need to decide which chain to build on. Remember that ultimately, the longest chain wins. So, if you are still building on a chain with block 10 as its last block when you receive a message containing blocks 11 and 12, you probably should give up on building block 11.

**Bluffing:** Sneaking in a bad block is easier and quicker than finding a nonce that gives you a valid block. So, miners have a strong incentive to bluff. If you receive a block and fail to check its validity, you will be contributing to the bluffer winning. On the other hand, it takes time to check the validity of blocks you receive, and you might decide to let others check the block while you continue to mine. It’s your choice. If you find a bad block, you should warn the other miners by yelling, “The new block sent by … is invalid.” Hopefully, they will not build on the invalid chain and your chain will survive. Beware, however, of competitors yelling that a block is invalid even when it is valid. You are advised to check for yourself. If any of the last four blocks is invalid, I will remove the invalid block and all blocks following it before I determine the winner.

**Validation:** To be valid, a block must follow these rules:

* The line after the header must be the SHA 256 hash of the header and must meet rule (to be revealed at the start of the simulation) of a valid header hash
* Block number must be one more than the block number of the previous block
* The header must follow the format laid out in the section above called “Header format.”
* The transaction must be selected from the mempool and must not have been used in a previous block

**Pooling:** You may team up with another member of the class. If you do, use both your names in the header with only a space between them. The pool will be twice as likely as an individual is to find a valid block, but if the pool wins, each member will receive only half the award. I can only split the award into two parts, so pools of three or more are prohibited.

**Practice:** At the start of the simulation, we will have two practice rounds, each adding one block to the blockchain. These blocks will not be counted in determining the winner of the competition. Only blocks 3 and beyond will count.

**Conclusion:** After the simulation is over, post at BB’s Assignments link/Consensus Simulation your copy of the blockchain followed by a short paragraph about what you learned from the simulation.