Quiz: Lab Hash Experiments

Instructions: 20 multiple choice questions — Select the best answer — Answers revealed after each question

Quiz (1–5)

Q1. What Python library is used for computing SHA-256 hashes?
A) cryptolib B) hashlib C) sha256lib D) securelib

Quiz (1–5)

Q1. What Python library is used for computing SHA-256 hashes?
A) cryptolib B) hashlib C) sha256lib D) securelib
Answer: B – hashlib is the standard library for hash functions in Python.

Q2. What is the output length of a SHA-256 hash in hexadecimal characters?
A) 32 B) 64 C) 128 D) 256

Quiz (1–5)

Q1. What Python library is used for computing SHA-256 hashes?
A) cryptolib B) hashlib C) sha256lib D) securelib
Answer: B – hashlib is the standard library for hash functions in Python.

Q2. What is the output length of a SHA-256 hash in hexadecimal characters?
A) 32 B) 64 C) 128 D) 256
Answer: B – SHA-256 produces 256 bits = 32 bytes = 64 hex characters.

Q3. If you hash "Blockchain" and "blockchain", what will happen?
A) Same hash B) Similar hashes C) Completely different hashes D) Error

Quiz (1–5)

Q1. What Python library is used for computing SHA-256 hashes?
A) cryptolib B) hashlib C) sha256lib D) securelib
Answer: B – hashlib is the standard library for hash functions in Python.

Q2. What is the output length of a SHA-256 hash in hexadecimal characters?
A) 32 B) 64 C) 128 D) 256
Answer: B – SHA-256 produces 256 bits = 32 bytes = 64 hex characters.

Q3. If you hash "Blockchain" and "blockchain", what will happen?
A) Same hash B) Similar hashes C) Completely different hashes D) Error
Answer: C – Hash functions are case-sensitive; any input change produces a completely different output.

Q4. Which property ensures that finding two inputs with the same hash is computationally infeasible?
A) Determinism B) Collision resistance C) Preimage resistance D) Avalanche effect

Quiz (1–5)

Q1. What Python library is used for computing SHA-256 hashes?
A) cryptolib B) hashlib C) sha256lib D) securelib
Answer: B – hashlib is the standard library for hash functions in Python.

Q2. What is the output length of a SHA-256 hash in hexadecimal characters?
A) 32 B) 64 C) 128 D) 256
Answer: B – SHA-256 produces 256 bits = 32 bytes = 64 hex characters.

Q3. If you hash "Blockchain" and "blockchain", what will happen?
A) Same hash B) Similar hashes C) Completely different hashes D) Error
Answer: C – Hash functions are case-sensitive; any input change produces a completely different output.

Q4. Which property ensures that finding two inputs with the same hash is computationally infeasible?
A) Determinism B) Collision resistance C) Preimage resistance D) Avalanche effect
Answer: B – Collision resistance means it's practically impossible to find two different inputs producing the same hash.

Q5. What happens if you add a single space to the end of a string before hashing?
A) Hash unchanged B) Hash slightly modified C) Completely different hash D) Hashing fails

Quiz (1–5)

Q1. What Python library is used for computing SHA-256 hashes?
A) cryptolib B) hashlib C) sha256lib D) securelib
Answer: B – hashlib is the standard library for hash functions in Python.

Q2. What is the output length of a SHA-256 hash in hexadecimal characters?
A) 32 B) 64 C) 128 D) 256
Answer: B – SHA-256 produces 256 bits = 32 bytes = 64 hex characters.

Q3. If you hash "Blockchain" and "blockchain", what will happen?
A) Same hash B) Similar hashes C) Completely different hashes D) Error
Answer: C – Hash functions are case-sensitive; any input change produces a completely different output.

Q4. Which property ensures that finding two inputs with the same hash is computationally infeasible?
A) Determinism B) Collision resistance C) Preimage resistance D) Avalanche effect
Answer: B – Collision resistance means it's practically impossible to find two different inputs producing the same hash.

Q5. What happens if you add a single space to the end of a string before hashing?
A) Hash unchanged B) Hash slightly modified C) Completely different hash D) Hashing fails
Answer: C – Even tiny changes (one space) produce completely different hashes due to avalanche effect.

Quiz (6–10)

Q6. What percentage of bits should change when one input bit is flipped (avalanche effect)?
A) 10% B) 25% C) 50% D) 100%

Quiz (6–10)

Q6. What percentage of bits should change when one input bit is flipped (avalanche effect)?
A) 10% B) 25% C) 50% D) 100%
Answer: C – The avalanche effect means approximately 50% of output bits should change.

Q7. In the lab exercise, changing "dog" to "dof" in the input resulted in what?
A) No change B) Small change C) About 50% bit change D) Complete reversal

Quiz (6–10)

Q6. What percentage of bits should change when one input bit is flipped (avalanche effect)?
A) 10% B) 25% C) 50% D) 100%
Answer: C – The avalanche effect means approximately 50% of output bits should change.

Q7. In the lab exercise, changing "dog" to "dof" in the input resulted in what?
A) No change B) Small change C) About 50% bit change D) Complete reversal
Answer: C – This demonstrates the avalanche effect with approximately 50% of bits differing.

Q8. What is a "nonce" in proof-of-work mining?
A) A timestamp B) A counter incremented to find valid hash C) The block data D) The previous hash

Quiz (6–10)

Q6. What percentage of bits should change when one input bit is flipped (avalanche effect)?
A) 10% B) 25% C) 50% D) 100%
Answer: C – The avalanche effect means approximately 50% of output bits should change.

Q7. In the lab exercise, changing "dog" to "dof" in the input resulted in what?
A) No change B) Small change C) About 50% bit change D) Complete reversal
Answer: C – This demonstrates the avalanche effect with approximately 50% of bits differing.

Q8. What is a "nonce" in proof-of-work mining?
A) A timestamp B) A counter incremented to find valid hash C) The block data D) The previous hash
Answer: B – A nonce (number used once) is incremented until the hash meets the difficulty target.

Q9. If mining difficulty increases from 3 to 4 leading zeros, how much harder does it become?
A) 2x B) 4x C) 10x D) 16x

Quiz (6–10)

Q6. What percentage of bits should change when one input bit is flipped (avalanche effect)?
A) 10% B) 25% C) 50% D) 100%
Answer: C – The avalanche effect means approximately 50% of output bits should change.

Q7. In the lab exercise, changing "dog" to "dof" in the input resulted in what?
A) No change B) Small change C) About 50% bit change D) Complete reversal
Answer: C – This demonstrates the avalanche effect with approximately 50% of bits differing.

Q8. What is a "nonce" in proof-of-work mining?
A) A timestamp B) A counter incremented to find valid hash C) The block data D) The previous hash
Answer: B – A nonce (number used once) is incremented until the hash meets the difficulty target.

Q9. If mining difficulty increases from 3 to 4 leading zeros, how much harder does it become?
A) 2x B) 4x C) 10x D) 16x
Answer: D – Each additional hex zero increases difficulty by 16x (one hex digit has 16 possible values).

Q10. What is the only way to find a valid nonce in proof-of-work?
A) Mathematical formula B) Brute-force trial and error C) Reverse the hash D) Use quantum computing

Quiz (6–10)

Q6. What percentage of bits should change when one input bit is flipped (avalanche effect)?
A) 10% B) 25% C) 50% D) 100%
Answer: C – The avalanche effect means approximately 50% of output bits should change.

Q7. In the lab exercise, changing "dog" to "dof" in the input resulted in what?
A) No change B) Small change C) About 50% bit change D) Complete reversal
Answer: C – This demonstrates the avalanche effect with approximately 50% of bits differing.

Q8. What is a "nonce" in proof-of-work mining?
A) A timestamp B) A counter incremented to find valid hash C) The block data D) The previous hash
Answer: B – A nonce (number used once) is incremented until the hash meets the difficulty target.

Q9. If mining difficulty increases from 3 to 4 leading zeros, how much harder does it become?
A) 2x B) 4x C) 10x D) 16x
Answer: D – Each additional hex zero increases difficulty by 16x (one hex digit has 16 possible values).

Q10. What is the only way to find a valid nonce in proof-of-work?
A) Mathematical formula B) Brute-force trial and error C) Reverse the hash D) Use quantum computing
Answer: B – No shortcut exists; miners must try nonces sequentially until finding a valid hash.

Quiz (11–15)

Q11. What is the “previous hash” field in a block used for?
A) Encrypt data B) Link blocks into chain C) Store transaction data D) Calculate difficulty

Quiz (11–15)

Q11. What is the "previous hash" field in a block used for?

A) Encrypt data B) Link blocks into chain C) Store transaction data D) Calculate difficulty

Answer: B – Previous hash links the current block to its predecessor, creating the chain structure.

Q12. What value does the genesis block (first block) use for previous_hash?

A) Empty string B) "0" or all zeros C) Random value D) Hash of timestamp

Quiz (11–15)

Q11. What is the "previous hash" field in a block used for?
A) Encrypt data B) Link blocks into chain C) Store transaction data D) Calculate difficulty
Answer: B – Previous hash links the current block to its predecessor, creating the chain structure.

Q12. What value does the genesis block (first block) use for previous_hash?
A) Empty string B) "0" or all zeros C) Random value D) Hash of timestamp
Answer: B – Genesis block has no predecessor, so previous_hash is typically "0" or all zeros.

Q13. If you tamper with block 3 in a 10-block chain, how many blocks are invalidated?
A) Only block 3 B) Blocks 3-10 C) All 10 blocks D) None

Quiz (11–15)

Q11. What is the "previous hash" field in a block used for?
A) Encrypt data B) Link blocks into chain C) Store transaction data D) Calculate difficulty
Answer: B – Previous hash links the current block to its predecessor, creating the chain structure.

Q12. What value does the genesis block (first block) use for previous_hash?
A) Empty string B) "0" or all zeros C) Random value D) Hash of timestamp
Answer: B – Genesis block has no predecessor, so previous_hash is typically "0" or all zeros.

Q13. If you tamper with block 3 in a 10-block chain, how many blocks are invalidated?
A) Only block 3 B) Blocks 3-10 C) All 10 blocks D) None
Answer: B – Tampering invalidates block 3 and all subsequent blocks (4-10) since their previous hashes no longer match.

Q14. What does the chain verification function check for each block?
A) Only the nonce B) Hash validity and previous hash link C) Transaction signatures D) Timestamp order

Quiz (11–15)

Q11. What is the "previous hash" field in a block used for?
A) Encrypt data B) Link blocks into chain C) Store transaction data D) Calculate difficulty
Answer: B – Previous hash links the current block to its predecessor, creating the chain structure.

Q12. What value does the genesis block (first block) use for previous_hash?
A) Empty string B) "0" or all zeros C) Random value D) Hash of timestamp
Answer: B – Genesis block has no predecessor, so previous_hash is typically "0" or all zeros.

Q13. If you tamper with block 3 in a 10-block chain, how many blocks are invalidated?
A) Only block 3 B) Blocks 3-10 C) All 10 blocks D) None
Answer: B – Tampering invalidates block 3 and all subsequent blocks (4-10) since their previous hashes no longer match.

Q14. What does the chain verification function check for each block?
A) Only the nonce B) Hash validity and previous hash link C) Transaction signatures D) Timestamp order
Answer: B – Verification checks that each block's hash is valid and correctly references the previous block's hash.

Q15. Why can't an attacker simply modify one block without detection?
A) Encryption prevents it B) Hash links break C) Network rejects it D) Signatures fail

Quiz (11–15)

Q11. What is the "previous hash" field in a block used for?
A) Encrypt data B) Link blocks into chain C) Store transaction data D) Calculate difficulty
Answer: B – Previous hash links the current block to its predecessor, creating the chain structure.

Q12. What value does the genesis block (first block) use for previous_hash?
A) Empty string B) "0" or all zeros C) Random value D) Hash of timestamp
Answer: B – Genesis block has no predecessor, so previous_hash is typically "0" or all zeros.

Q13. If you tamper with block 3 in a 10-block chain, how many blocks are invalidated?
A) Only block 3 B) Blocks 3-10 C) All 10 blocks D) None
Answer: B – Tampering invalidates block 3 and all subsequent blocks (4-10) since their previous hashes no longer match.

Q14. What does the chain verification function check for each block?
A) Only the nonce B) Hash validity and previous hash link C) Transaction signatures D) Timestamp order
Answer: B – Verification checks that each block's hash is valid and correctly references the previous block's hash.

Q15. Why can't an attacker simply modify one block without detection?
A) Encryption prevents it B) Hash links break C) Network rejects it D) Signatures fail
Answer: B – Modifying block data changes its hash, breaking the link to the next block and invalidating the chain.

Quiz (16–20)

Q16. Which hash algorithm is considered cryptographically broken and should NOT be used?
A) SHA-256 B) SHA-3 C) MD5 D) BLAKE2

Quiz (16–20)

Q16. Which hash algorithm is considered cryptographically broken and should NOT be used?
A) SHA-256 B) SHA-3 C) MD5 D) BLAKE2
Answer: C – MD5 has known collision vulnerabilities and is no longer secure for cryptographic use.

Q17. How does Git version control use hash functions?
A) Encrypt repositories B) Generate commit IDs C) Compress files D) Authenticate users

Quiz (16–20)

Q16. Which hash algorithm is considered cryptographically broken and should NOT be used?
A) SHA-256 B) SHA-3 C) MD5 D) BLAKE2
Answer: C – MD5 has known collision vulnerabilities and is no longer secure for cryptographic use.

Q17. How does Git version control use hash functions?
A) Encrypt repositories B) Generate commit IDs C) Compress files D) Authenticate users
Answer: B – Git uses SHA-1 (moving to SHA-256) to generate unique commit identifiers based on content.

Q18. What computational property makes blockchain immutability expensive to break?
A) Encryption strength B) Network size C) Re-mining cost D) Disk space

Quiz (16–20)

Q16. Which hash algorithm is considered cryptographically broken and should NOT be used?
A) SHA-256 B) SHA-3 C) MD5 D) BLAKE2
Answer: C – MD5 has known collision vulnerabilities and is no longer secure for cryptographic use.

Q17. How does Git version control use hash functions?
A) Encrypt repositories B) Generate commit IDs C) Compress files D) Authenticate users
Answer: B – Git uses SHA-1 (moving to SHA-256) to generate unique commit identifiers based on content.

Q18. What computational property makes blockchain immutability expensive to break?
A) Encryption strength B) Network size C) Re-mining cost D) Disk space
Answer: C – To tamper with history, attackers must re-mine all subsequent blocks faster than the network.

Q19. In the lab, you computed file hashes. What's the practical use of this?
A) File compression B) Integrity verification C) File encryption D) Access control

Quiz (16–20)

Q16. Which hash algorithm is considered cryptographically broken and should NOT be used?
A) SHA-256 B) SHA-3 C) MD5 D) BLAKE2
Answer: C – MD5 has known collision vulnerabilities and is no longer secure for cryptographic use.

Q17. How does Git version control use hash functions?
A) Encrypt repositories B) Generate commit IDs C) Compress files D) Authenticate users
Answer: B – Git uses SHA-1 (moving to SHA-256) to generate unique commit identifiers based on content.

Q18. What computational property makes blockchain immutability expensive to break?
A) Encryption strength B) Network size C) Re-mining cost D) Disk space
Answer: C – To tamper with history, attackers must re-mine all subsequent blocks faster than the network.

Q19. In the lab, you computed file hashes. What's the practical use of this?
A) File compression B) Integrity verification C) File encryption D) Access control
Answer: B – File hashing verifies integrity; if the hash changes, the file was modified or corrupted.

Q20. If a hash function were NOT deterministic, what would fail?
A) Mining would be faster B) Verification would be impossible C) Security would improve D) Nothing

Quiz (16–20)

Q16. Which hash algorithm is considered cryptographically broken and should NOT be used?
A) SHA-256 B) SHA-3 C) MD5 D) BLAKE2
Answer: C – MD5 has known collision vulnerabilities and is no longer secure for cryptographic use.

Q17. How does Git version control use hash functions?
A) Encrypt repositories B) Generate commit IDs C) Compress files D) Authenticate users
Answer: B – Git uses SHA-1 (moving to SHA-256) to generate unique commit identifiers based on content.

Q18. What computational property makes blockchain immutability expensive to break?
A) Encryption strength B) Network size C) Re-mining cost D) Disk space
Answer: C – To tamper with history, attackers must re-mine all subsequent blocks faster than the network.

Q19. In the lab, you computed file hashes. What's the practical use of this?
A) File compression B) Integrity verification C) File encryption D) Access control
Answer: B – File hashing verifies integrity; if the hash changes, the file was modified or corrupted.

Q20. If a hash function were NOT deterministic, what would fail?
A) Mining would be faster B) Verification would be impossible C) Security would improve D) Nothing
Answer: B – Determinism (same input = same output) is essential; otherwise you couldn't verify by recomputing the hash.