Project Questions

Jordan Tewnion

Topic 7

What are the benefits of computers using the Binary Number System?

A:

1. The Binary Number System is an exact copy of the Decimal Number System, making it easy for both humans and computers to read the same number.
2. Binary numbers can be perfectly stored and reproduced by computers.
3. Binary numbers are perfect for computers who are easily able to store three different bit states for each binary digit.
4. Binary numbers are better for computers than decimal numbers as computers have difficulty determining value of numbers by digit position.

Answer: Binary numbers can be perfectly stored and reproduced by computers.

Explanation: Computers are very good at storing two distinct states (in this case 0 and 1). Binary numbers are able to be perfectly stored and retrieved by computers without loss, it is much more difficult for them to store decimal numbers.

What is the purpose of the least significant bit in a binary number?

A:

1. The least significant bit determines if the binary number is positive or negative.
2. The least significant bit determines the length of the binary representation of a number.
3. The least significant bit is a placeholder as the leftmost bit of a binary number used for internal formatting.
4. The least significant bit is a relic from the days of early computers and serves no purpose in modern 64-bit computing.

Answer: The least significant bit determines if the binary number is positive or negative.

Explanation: The least significant bit is used to determine if a binary number is positive or negative. Regardless of the number of bits used to represent a number, the least significant bit is always the leftmost bit.

How many bits are in a byte?

A:

1. There are 2 bits in a byte.
2. There is only one bit in a byte.
3. There are 4 bits in a byte.
4. There are 8 bits in a byte.

Answer: There are 8 bits in a byte.

Explanation: The number of bits in a byte is 8. This is also known as an octet. 4 bits is known as a nybble, half of a byte.

What are the range of symbols used in Hexadecimal Numbers?

A:

1. Hexadecimal Numbers use symbols ranging from 1-16.
2. Hexadecimal Numbers use symbols ranging from 1-10 and A-F.
3. Hexadecimal Numbers use symbols ranging from 0-9 and A-F.
4. Hexadecimal Numbers use symbols ranging from A-P.

Answer: Hexadecimal Numbers use symbols ranging from 0-9 and A-F.

Explanation: The numerical values represented by hexadecimal numbers are the digits 0-9 A common mistake people make is thinking that the digits represented start from 1, but computers start counting at 0 which leads to the mistake. For letters, A-F represent the values 10-15.

What is a real-world equivalent to the unary number system?

A:

1. There is no difference between unary numbers and decimal numbers.
2. Tally marks are a real-world application of the unary number system.
3. Computers often use the unary number system in applications to make counting more readable to users.
4. Unary numbers are used as an easy way to make binary numbers more readable.

Answer: Tally marks are a real-world application of the unary number system.

Explanation: There aren’t that many practical applications of the unary number system. Computers are bad at storing the unary number system in a practical way, and the representation of a unary number is inefficient to display. Tally marks are one of the few uses of the unary number system in the real world.

How many binary digits are required to represent a single hexadecimal digit?

A:

1. 4 bits are required to represent a single hexadecimal digit.
2. 16 bits are required to represent a single hexadecimal digit.
3. 6 bits are required to represent a single hexadecimal digit.
4. You only need one bit to represent a single hexadecimal digit.

Answer: 4 bits are required to represent a single hexadecimal digit.

Explanation: To properly represent a hexadecimal digit, 4 bits are required. The range of values represented by a 4-bit number are 0-15, which is the minimum required to represent the number completely.

What determines the length of a binary string?

A:

1. The length of a binary string is the amount of storage space the string requires to store on a hard drive.
2. The length of a binary string is the number of bits inside the string (excluding leading 0s).
3. The length of a binary string is the number of bits inside the string (including leading 0s).
4. The length of a binary string is exclusively the number of 1s that are stored inside the string.

Answer: The length of a bit string is the number of bits inside a string (excluding leading 0s).

Explanation: The length of a bit string is determined by the total number of bits inside a string, subtracting leading 0s from the count. Removing leading 0s allows for the number to be the smallest possible without losing any accuracy in the value of the number.

What is the hamming weight of a binary string?

A:

1. The hamming weight of a binary string is the number of 0s in the string.
2. The hamming weight of a binary string is equal to the length of the string (excluding leading 0s).
3. The hamming weight of a binary string is the number of 1s in the string.
4. The hamming weight of a binary string is the number of leading 0s in the string.

Answer: The hamming weight of a binary string is the number of 1s in the string.

Explanation: The hamming weight of a binary string is equal to the number of 1s in the string. 0s are excluded from the count, regardless of their position (leading or not), resulting in the weight of the string.

What happens when two strings are concatenated?

A:

1. All the bits in each string are replaced with 0s.
2. The first string is replaced by the second string, effectively copying the value of the second string.
3. The second string is appended to the end of the first string.
4. The length of the second string is appended to the end of the first string, as trailing 0s.

Answer: The second string is appended to the end of the first string.

Explanation: The value of the second string is added to the end of the first string. The value of the first string is unchanged, and the total value of the second string is added in full.

How many bits are used to represent a character in ASCII?

A:

1. An ASCII character is represented using 8 bits.
2. An ASCII character is represented using 4 bits.
3. An ASCII character is represented using 7 bits.
4. An ASCII character is represented using 16 bits.

Answer: An ASCII character is represented using 8 bits.

Explanation: While it is true that an ASCII character is represented using 7 bits, an 8th bit added for parity. It’s rather impressive that all the ASCII characters can be represented by 128 values.

What is an aspect shared between the Binary and Decimal Number systems?

A:

1. They are perfect copies of each other, making them simple to use in computer-based arithmetic.
2. Both number systems are position based, where the position of a digit relates to its value.
3. They cover the same number of symbols for the representation of digits.
4. Both number systems give all digits the same value, the position of each digit is irrelevant.

Answer: Both number systems are position based, where the position of a digit relates to its value.

Explanation: Both number systems are based on the position of the digits. While decimal numbers give a value of 10^x to the position of each digit, binary numbers give a value of 2^x to the position of each digit.

What is the prefix used to represent binary numbers?

A:

1. The prefix for binary numbers is 0x.
2. The prefix for binary numbers is 1b.
3. The prefix for binary numbers is 0n.
4. The prefix for binary numbers if 0b.

Answer: The prefix for binary numbers is 0b.

Explanation: 0b is the prefix used for binary numbers. 1b and 0n are incorrect prefixes to depict binary numbers, and 0x is the prefix used for hexadecimal numbers.

What is the length of an empty string?

A:

1. The length of an empty string is equal to the amount of storage space required to store it on a hard drive.
2. The length of an empty string is undefined.
3. The length of an empty string is always equal to 1.
4. The length of an empty string is defined as 0.

Answer: The length of an empty string is defined as 0.

Explanation: The length of an empty string is always defined as 0. The amount of storage space required to store a string is irrelevant to the actual length of the string.

What happens to a bit string when a bitwise not operation is applied?

A:

1. All the bits in the string are set to a value of 0.
2. The length of the bit string is set to 0, effectively deleting the string.
3. The value of each bit is flipped, each 1 is set to a 0, and every 0 is set to a 1.
4. All the bits in the string are set to a random value, chosen between 0 or 1.

Answer: The value of each bit is flipped, each 1 is set to a 0, and every 0 is set to a 1.

Explanation: The value of each bit is reversed in a bitwise not operation. This does not mean the string is deleted or that the value of each bit is random, it simply means that every 0 is set to a 1, and each 1 is set to a 0.

What happens when a binary string is truncated?

A:

1. A specified number of bits are discarded starting from either the start or end of the bitstring.
2. A specified number of bits are added to the rightmost bit of the string.
3. The string has a bitwise and operation applied to it, effectively deleting the string.
4. The length of the bit is set to 1, with a random value of either 0 or 1.

Answer: A specified number of bits are discarded starting from either the start or end of the bitstring.

Explanation: The truncation operation on a bitstring removes a specified number of bits from either the left or the right side of the string. The format used for the operation is lsb^n(x) and msb^n(x), removing n bits from the bitstring x.

How many hexadecimal characters are required to represent an ASCII character?

A:

1. You only need one hexadecimal character to represent an ASCII character.
2. You only need two hexadecimal characters to represent an ASCII character.
3. You are unable to represent the value of an ASCII character with hexadecimal characters.
4. You only need eight hexadecimal characters to represent an ASCII character, the same as the number of binary digits required.

Answer: You only need two hexadecimal characters to represent an ASCII character.

Explanation: Only two hexadecimal characters are required to represent a singular ASCII character. Each ASCII character is represented by 8 bits (7 bits to represent the character, and 1 bit for parity), and a single hexadecimal character represents 4 bits, leading to a minimum of two characters to represent a single ASCII character.

How many characters are represented by the ASCII character set?

A:

1. The ASCII character set represents 128 unique characters.
2. The ASCII character set represents 64 unique characters.
3. The ASCII character set represents 32 unique characters.
4. The ASCII character set represents 127 unique characters.

Answer: The ASCII character set represents 127 unique characters.

Explanation: While it is true that the number of possible ASCII characters that can be represented by the 7-bit integer limit is 128 characters, the 128th character is reserved for the delete operation. As such, only 127 characters are represented by the ASCII character set.

How many binary digits are represented by a word?

A:

1. A word represents 16 binary digits.
2. A word represents 32 binary digits.
3. A word represents 64 binary digits.
4. A word represents a variable number of binary digits that differs based on the design of the processor used on the hardware.

Answer: A word represents a variable number of binary digits that differs based on the design of the processor used on the hardware.

Explanation: While it is true that a word could represent 32, 16, or 64 binary digits (or other values 2^x), the number of bits in a word is determined by the processor used in the hardware. Most modern computers use either 32 or 64 bits in a word.

What is the range of symbols used in the decimal number system?

A:

1. The symbols used in decimal numbers range from 0-10.
2. The symbols used in decimal numbers range from 0-9.
3. The symbols used in decimal numbers range from 0-F.
4. The symbols used in decimal numbers range from 1-10.

Answer: The symbols used in decimal numbers range from 0-9.

Explanation: The range of symbols for decimal numbers is from 0-9. The range 0-F is the range of symbols used to represent binary numbers, and the number 10 is not a unique symbol, thus it is invalid.

What numeral system is the decimal number system based on?

A:

1. The decimal number system is based off of the Indo-Arab numeral system.
2. The decimal number system is based off of the Korean numeral system.
3. The decimal number system is based off of the Chinese rod numeral system.
4. The decimal number system is based off of the Burmese numeral system.

Answer: The decimal number system is based off of the Indo-Arab numeral system.

Explanation: The digits 0-9 are based off of the Indo-Arab numeral system. The Korean, Chinese, and Burmese number systems are region numeral systems and are not as widely used as the Indo-Arab numeral system.

Topic 8

Which of the following does not use cryptography in some way?

A:

1. Protecting unencrypted files against malware attacks.
2. Encrypting the SIM card used in a mobile phone.
3. Encrypting and hashing a password table.
4. Sending encrypted information over a network to another user.

Answer: Protecting unencrypted files against malware attacks.

Explanation: Unencrypted files do not use cryptography. As the file is stored in plaintext it has not been encoded in any way, thus cryptography is not involved. All the other answers involve encryption and cryptography in some way.

What are the origins of the word “Cryptography”?

A:

1. The word is based of a phrase coined in the journal of Alan Turing.
2. The word originated in modern English, and it has no basis in historical languages.
3. The word is based off the Ancient Greek kryptós and graphein.
4. The word is based off of a Latin phrase which has since become untranslatable.

Answer: The word is based off the Ancient Greek kryptós and graphein.

Explanation: While it would be interesting to see the word originate in modern English or as a phrase coined by a famous computer scientist, the origins of the word cryptography are based in Ancient Greek. It is a concatenation of kryptós and graphein.

Which element of a system should be a secret?

A:

1. The algorithm should be kept secret, to ensure that the data it is protecting is not accessed by an unauthorised third party.
2. The cryptography key should be kept a secret.
3. The identity of the sender should be kept a secret, to ensure anonymity.
4. The identity of the receiver should be kept a secret, to ensure their key does not get accessed.

Answer: The cryptography key should be kept a secret.

Explanation: Ultimately, the identity of the sender and receiver are irrelevant to an encryption algorithm. Encryption algorithms are usually public, as the cryptography key should be unique and private, so the message is impossible to decrypt by an unauthorised recipient.

What is the difference between plaintext and ciphertext?

A:

1. There is no difference between plaintext and ciphertext, they are simply different words for the same thing.
2. Plaintext is an encrypted message, and ciphertext is the key used to decode the plaintext.
3. Plaintext is a gibberish message generated by an algorithm contained in the ciphertext message.
4. Plaintext is the original readable message, and ciphertext is the unreadable encrypted message.

Answer: Plaintext is the original readable message, and ciphertext is the unreadable encrypted message.

Explanation: As can be reasoned out with a bit of thinking, plaintext is the original form of a message, and ciphertext is the encrypted (and thus unreadable) message.

What is the difference between Symmetric-key and Asymmetric-key encryption?

A:

1. In symmetric encryption, the keys for encryption and decryption are different, but in asymmetric encryption the keys for encryption and decryption are the same.
2. In symmetric encryption the encryption key is public, but in asymmetric encryption both the public key and the private keys are secret.
3. In asymmetric encryption you use only one key for encryption and decryption, but for symmetric encryption you use different keys for encryption and decryption.
4. In symmetric encryption the key for encryption and decryption is the same, but in asymmetric encryption the keys for encryption and decryption are different.

Answer: In symmetric encryption the key for encryption and decryption is the same, but in asymmetric encryption the keys for encryption and decryption are different.

Explanation: Symmetric encryption uses the same key for both encryption and decryption, and the key must be shared between the sender and the recipient in order for the message to be decoded. In asymmetric encryption, the encryption key is public, but the decryption key is private and must be shared between the sender and the recipient.

How is a message decrypted in Symmetric-key encryption?

A:

1. In symmetric encryption, the secret key must be securely shared through a key exchange protocol in order to decrypt a message.
2. In symmetric encryption, users only need a key to encrypt a message, not to decrypt a message.
3. In symmetric encryption, users need a secret key to decrypt a message, but not to encrypt a message.
4. In symmetric encryption, a user needs two distinct secret keys in order to encrypt and decrypt a message.

Answer: In symmetric encryption, the secret key must be securely shared through a key exchange protocol in order to decrypt a message.

Explanation: Symmetric encryption requires a secret key to be shared between a sender and a recipient in order for a message to be decrypted. A key exchange program is often used in order to prevent the key from being leaked to an unauthorised third party.

Who is credited with inventing Information Theory?

A:

1. Information Theory was invented by Claude Shannon.
2. Information Theory was invented by Steve Jobs.
3. Information Theory was invented by Alan Turing.
4. Information Theory was invented by James Gosling.

Answer: Information Theory was invented by Claude Shannon.

Explanation: The invention of Information Theory is often credited to Claude Shannon (1916-2001). He was also credited with formalizing secrecy with the definition of perfect secrecy.

What type of algorithm is considered “efficient” in regard to cryptography?

A:

1. An efficient algorithm is an algorithm that runs quickly on hardware that is at least five years old.
2. An efficient algorithm is an algorithm that can be calculated by the strongest computer that exists today.
3. An efficient algorithm is an algorithm that can be calculated by any computer that exists today.
4. An efficient algorithm is an algorithm that will eventually end.

Answer: An efficient algorithm is an algorithm that can be calculated by the strongest computer that exists today.

Explanation: In regard to cryptography, an algorithm is considered efficient if the computation is possible by the strongest computers that exist today for a given input size. An efficient algorithm may not be possible for most computers available for purchase on the market to calculate, but it is still satisfied by the definition.

What is a flaw in the definition of security?

A:

1. It is too general, and does not cover every possible security system or defensive tactic.
2. It is too simple, as it does not cover things an attacker can do in response to a secure system.
3. It is too broad, almost every defensive tactic is covered under this definition.
4. It is too strict, as it does not allow security to be mathematically proved.

Answer: It is too simple, as it does not cover things an attacker can do in response to a secure system.

Explanation: The definition of security is strong in the sense that it allows security to be mathematically proved. However, it does not consider any action an attacker can take to attempt to damage a secure system.

Who is credited with inventing one-time-pad encryption?

A:

1. One-time-pad encryption was invented by Gilbert Vernam.
2. One-time-pad encryption was invented by Guido van Rossum.
3. One-time-pad encryption was invented by Alex Medvendnikov.
4. One-time-pad encryption was invented by Dennis Ritchie.

Answer: One-time-pad encryption was invented by Gilbert Vernam.

Explanation: The invention of one-time-pad encryption is credited to Gilbert Vernam. He was an engineer at AT&T at the time, and he also is credited with the invention of the stream cipher.

Which bitwise operation is used in the mechanical one-time-pad machine.

A:

1. The bitwise operation used is a mechanical implementation of the exclusive or operation.
2. The bitwise operation used is a mechanical implementation of the and operation.
3. The bitwise operation used is a mechanical implementation of the inclusive or operation.
4. The bitwise operation used is a mechanical implementation of the not operation.

Answer: The bitwise operation used is a mechanic implementation of the exclusive or operation.

Explanation: In the machine invented by Gilbert Vernam, the one-time-pad was a slightly more rudimentary. In the machine, the message is encrypted by mechanically exclusive or-ing a message with a secret key.

What must be true about a one-time-pad in order for it to have “perfect” secrecy.

A:

1. A one-time-pad has perfect secrecy so long as it is never attacked by a third party.
2. A one-time-pad has perfect secrecy so long as all of the conditions are satisfied.
3. A one-time-pad has perfect secrecy so long as it is implemented correctly.
4. A one-time-pad has perfect secrecy so long as it is never leaked by the user.

Answer: A one-time-pad has perfect secrecy so long as all of the conditions are satisfied.

Explanation: All the elements of perfect secrecy must be completely satisfied for a one-time-pad to have perfect secrecy. It is ultimately irrelevant if the pad is attacked by a third party or is implemented incorrectly.

What is not a possible risk that a one-time-pad could fail?

A:

1. The one-time-pad is not truly uniformly random.
2. A user reuses their pad for two or more encryption.
3. The one-time-pad is sent to multiple recipients.
4. A segment of the one-time-pad is leaked to a third party.

Answer: The one-time-pad is sent to multiple recipients.

Explanation: The risk that a one-time-pad is not truly uniformly random, is reused, or is leaked in part or full to a third party, are all legitimate risks that compromise the security of a one-time-pad. A one-time-pad being sent to multiple recipients is not a risk that it could fail, so long as all recipients are intended to receive the one-time-pad.

What is the foundation assumption of perfect secrecy?

A:

1. Perfect secrecy assumes that the one-time-pad is truly random.
2. Perfect secrecy assumes that the one-time-pad is not accessed by a third party.
3. Perfect secrecy assumes that the sender and the receiver are anonymous.
4. Perfect secrecy assumes that a message is below a certain length to be encrypted.

Answer: Perfect secrecy assumes that the one-time-pad is truly random.

Explanation: The foundation assumption of perfect secrecy is that the one-time-pad must be truly random in order for a message to have perfect secrecy. This can be difficult, as generating truly random pads can be difficult for computers to do quickly.

What is a flaw with perfect secrecy?

A:

1. Perfect secrecy has no flaws, so long as the definition of perfect security is satisfied.
2. A message could be tampered with as perfect secrecy says nothing about message integrity.
3. Perfect secrecy does not consider the size of a message, utilizing inefficient one-time-pads.
4. Perfect secrecy does not cover the identities of the sender and the receiver of a message.

Answer: A message could be tampered with as perfect secrecy says nothing about message integrity.

Explanation: Even if a message satisfies the definition of perfect secrecy, the content of the message could still be tampered with. Someone could tamper with the value of the one-time-pad to alter the message received by the recipient, ultimately compromising the integrity of the message.

What is not an element of a one-time-pad with perfect security?

A:

1. A pad must be truly uniformly random.
2. A pad must have a different length than the message.
3. A pad must be shared before a communication starts.
4. A pad must be used for only one message.

Answer: A pad must have a different length than the message.

Explanation: Unfortunately, the one-time-pad having a different length than the message is not an element of a one-time-pad with perfect security. A one-time-pad must be the same length as the message in order for it to have perfect security.

What is the key difference between block ciphers and stream ciphers?

A:

1. Block ciphers are broken into blocks, and stream ciphers are broken into strings before encryption.
2. Block ciphers are processed by a pseudorandom string, while stream ciphers are processed by an encryption algorithm.
3. Both block ciphers and stream ciphers use the same encryption algorithm to mask the content of the message.
4. Block ciphers are processed by an encryption algorithm, while stream ciphers are masked by a pseudorandom string.

Answer: Block ciphers are processed by an encryption algorithm, while stream ciphers are masked by a pseudorandom string.

Explanation: Both block ciphers and stream ciphers have their message broken into blocks before they are turned into ciphertext. However, block ciphers are processed by an encryption algorithm, while stream ciphers are masked by a pseudorandom string.

What is the typical block size of modern ciphers?

A:

1. The typical block size is 64,127 bits.
2. The typical block size is 64,128 bits.
3. The typical block size is 65,536 bits.
4. The typical block size is 32,767 bits.

Answer: The typical block size is 64,128 bits.

Explanation: The size of a typical block in a modern cipher is 64,128 bits. Messages larger than this are split into multiple blocks, and if the length of the last block is less than 64,128 bits, additional bits are added for padding.

What is the average size of a symmetric key?

A:

1. The average size of a symmetric key is usually 128 or 256 bits.
2. The average size of a symmetric key is usually 256 or 512 bits.
3. The average size of a symmetric key is 64 bits.
4. The average size of a symmetric key is 16 to 32 bits.

Answer: The average size of a symmetric key is usually 128 or 256 bits.

Explanation: For modern symmetric keys, the average key size is usually 128 or 256 bits. With the computing power of attackers growing exponentially, 128 or 256 bit keys are necessary to ensure that it would be infeasible to conduct a brute force attack to crack the key.

What is an example of a classical cipher?

A:

1. An example of a classical cipher is a substitution cipher.
2. An example of a classical cipher is a symmetric-key cipher.
3. An example of a classical cipher is an asymmetric-key cipher.
4. An example of a classical cipher is a block cipher.

Answer: An example of a classical cipher is a substitution cipher.

Explanation: The substitution cipher is a prime example of a classical cipher. It was useful in historical times as a brute force attack would take an astronomical amount of time to calculate by hand, and it was used by Julius Caesar to communicate with his generals.