NIMCET Computer Awareness Sample Paper-16

Duration: 15 Minutes **Maximum Marks: 120**

Instructions

- This paper contains **20** Multiple Choice Questions (Single Correct).
- Each correct answer carries **+6 marks**.
- Each incorrect answer carries: **−1.5** marks.
- Unattempted questions carry **0** marks.
- Only one option is correct for each question.
- Use of mobile phones, smartwatches, calculators, or any electronic gadgets is strictly prohibited.

Q1. A computer system contains a 2 MB cache that is 8-way set associative. The cache block size is 64 bytes and the physical address size is 48 bits.

How many cache sets are present in the cache?

(A) 2^{10}

(B) 2^{11}

(C) 2^{12}

(D) 2^{13}

Q2. A processor uses dynamic branch prediction. Out of 50,000 branch instructions executed, 47,500 are predicted correctly.

What is the branch prediction accuracy of the processor?

(A) 90%

(B) 92%

(C) 95%

(D) 97%

Q3. A superscalar processor can issue up to 4 instructions per clock cycle.

Ignoring all hazards and dependencies, what is the minimum number of clock cycles required to execute 10,000 instructions?

(A) 1000

(B) 2000

(C) 2500

(D) 4000

Q4. A DMA controller transfers a block of data directly between memory and an I/O device.

Which of the following is the primary advantage of DMA over programmed I/O?

(A) Reduces memory size

(B) Eliminates cache misses

(C) Frees the CPU from continuous data transfer operations

(D) Increases disk capacity

Q5. The decimal number

$$2047$$

is represented using binary notation.

How many bits are required in its minimum unsigned binary representation?

(A) 10

(B) 11

(C) 12

(D) 13

Q6. The hexadecimal number

$$(FFFF)_{16}$$

is interpreted as an unsigned integer.

What is its decimal equivalent?

(A) 32767

(B) 65535

(C) 65534

(D) 131071

Q7. A cyclic redundancy check (CRC) generator polynomial has degree 5.

How many CRC check bits are appended to each message before transmission?

(A) 4

(B) 5

(C) 6

(D) 8

Q8. A communication system transmits data at a rate of 10 Mbps.

Ignoring all overheads, how much time is required to transmit a file of size 5 MB?

(A) 2 s

(B) 4 s

(C) 8 s

(D) 10 s

Q9. A floating-point system uses a biased exponent representation with a bias value of 127.

If the stored exponent is 140, what is the actual exponent represented?

(A) 11

(B) 12

(C) 13

(D) 14

Q10. A character encoding scheme uses exactly 21 bits to represent each code point.

Ignoring reserved values, how many distinct code points can theoretically be represented?

(A) 2^{16}

(B) 2^{20}

(C) 2^{21}

(D) 2^{32}

Q11. A demand-paging system experiences a page-fault rate of 0.002. Memory access time is 100 ns, while page-fault service time is 8 ms.

What is the approximate effective access time?

(A) 100 ns

(B) 1.6 μs

(C) 16 μs

(D) 160 μs

Q12. A process spends 30% of its execution time performing I/O operations.

According to Amdahl's Law, what is the maximum possible speedup if all CPU computation time becomes infinitely fast?

(A) 2.33

(B) 3.33

(C) 4.33

(D) Infinite

Q13. In a multiprogramming environment, the degree of multiprogramming is defined as:

Which of the following quantities?

(A) Number of CPUs in the system

(B) Number of processes residing in memory

(C) Number of files stored on disk

(D) Number of I/O devices connected

Q14. A magnetic tape and a hard disk are both secondary storage devices.

Which characteristic primarily makes hard disks superior for random-access workloads?

(A) Lower storage density

(B) Sequential access mechanism

(C) Direct access capability

(D) Lower transfer rate

Q15. The Boolean function

$$F(A, B, C) = \Sigma m(1, 3, 5, 7)$$

simplifies to which of the following expressions?

(A) A

(B) B

(C) C

(D) AB

Q16. For a Boolean function with n variables, the number of distinct Boolean functions possible is:

Choose the correct expression.

(A) 2^n

(B) 2^{2n}

(C) 2^{2^n}

(D) $n!$

Q17. A logic circuit is built entirely using NAND gates.

Which of the following properties of NAND gates makes this possible?

(A) Associative property

(B) Universal gate property

(C) Commutative property

(D) Distributive property

Q18. A DNS resolver receives a Time-To-Live (TTL) value of 3600 seconds for a cached record.

What does this value indicate?

(A) Maximum packet size

(B) Time before the cached record expires

(C) Time required to establish a connection

(D) DNS query timeout interval

Q19. A transaction in a DBMS satisfies the ACID properties.

Which ACID property guarantees that the effects of a committed transaction survive system failures?

(A) Atomicity

(B) Consistency

(C) Isolation

(D) Durability

Q20. In an operating system, starvation refers to a situation in which:

Which of the following occurs?

(A) Two processes wait indefinitely for each other

(B) A process never receives the resources it requires to proceed

(C) A process terminates unexpectedly

(D) Main memory becomes completely full

Detailed Solutions

Q1.

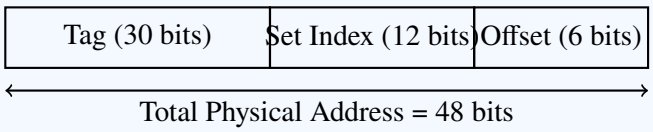
Solution

Concept: For a k-way set-associative cache,

$$S = \frac{\text{Cache Capacity}}{\text{Block Size} \times \text{Associativity}}$$

Address fields:

- Offset Bits = $\log_2(\text{Block Size})$
- Set Index Bits = $\log_2(S)$
- Tag Bits = Address Size − (Index Bits + Offset Bits)



Solution: Step 1: Express the given parameters in powers of 2:

- Cache Size = 2 MB = 2^{21} Bytes
- Block Size = 64 Bytes = 2^6 Bytes
- Associativity = 8 = 2^3

Step 2: Find the number of sets:

$$\begin{aligned} S &= \frac{\text{Cache Size}}{\text{Block Size} \times \text{Associativity}} \\ &= \frac{2^{21}}{2^6 \times 2^3} \\ &= 2^{12} \end{aligned}$$

Thus, the cache contains $2^{12} = 4096$ sets.

Step 3: Verification:

$$\text{Set Index Bits} = \log_2(2^{12}) = 12$$

Final Answer: $\boxed{2^{12}}$

Answer: (C)

Go Back to Question 1

Q2.

Solution

Concept: Branch prediction is used to reduce control hazards in a pipeline.

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Branches}} \times 100\%$$

$$\text{Misprediction Rate} = 100\% - \text{Accuracy}$$

Solution: Step 1: Extract the experimental metrics from the problem:

- **Total branch instructions executed (N_{total}):** $50,000$
- **Correctly predicted branches (N_{correct}):** $47,500$

Step 2: Apply the values directly to the branch prediction accuracy formula:

$$\begin{aligned}\text{Accuracy} &= \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100\% \\ &= \frac{47,500}{50,000} \times 100\%\end{aligned}$$

Step 3: Perform the algebraic simplification: Divide the numerator and denominator by 100 to simplify the trailing zeros:

$$\text{Accuracy} = \frac{475}{500} \times 100\%$$

Further dividing by 500:

$$\begin{aligned}\text{Accuracy} &= \frac{475}{5} \times \frac{100}{100}\% \\ &= 95\%\end{aligned}$$

Step 4: Verify the result using the misprediction rate:

$$\begin{aligned}\text{Mispredicted Branches} &= 50,000 - 47,500 = 2,500 \\ \text{Misprediction Rate} &= \frac{2,500}{50,000} \times 100\% = \frac{2.5}{50}\% = 5\% \\ \text{Accuracy} &= 100\% - 5\% = 95\%\end{aligned}$$

Both approaches yield 95%, confirming that Option C is correct.

Final Answer: $\boxed{95\%}$

Answer: (C)

Go Back to Question 2

Q3.

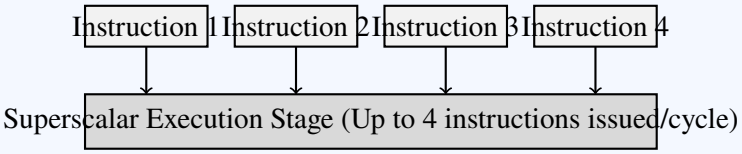
Solution

Concept: A superscalar processor can issue up to W instructions per clock cycle, where W is the issue width.

Under ideal conditions (no data, structural, or control hazards), the minimum number of clock cycles required for N instructions is:

$$C = \left\lceil \frac{N}{W} \right\rceil$$

$$C = \left\lceil \frac{N}{W} \right\rceil$$



Solution: Step 1: Extract the execution parameters:

- **Total instructions** (N)**:** $10,000$
- **Max issue width** (W)**:** 4 instructions per clock cycle

Step 2: Calculate the minimum number of clock cycles (C): Assuming a hazard-free pipeline where the processor sustains its peak execution throughput of 4 instructions per cycle:

$$\begin{aligned} C &= \frac{10,000 \text{ instructions}}{4 \text{ instructions/cycle}} \\ &= 2500 \text{ cycles} \end{aligned}$$

Step 3: Analyze the implications of the ceiling function: The ceiling function is critical for non-divisible instruction counts. For example, if there were $10,001$ instructions, the execution would require:

$$\left\lceil \frac{10,001}{4} \right\rceil = \lceil 2500.25 \rceil = 2501 \text{ cycles}$$

Since $10,000$ is perfectly divisible by 4, it requires exactly 2500 cycles. Any physical structural or data hazards would increase this cycle count, meaning 2500 is the absolute theoretical minimum. This matches Option C.

Final Answer: 2500

Answer: (C)

Go Back to Question 3

Q4.

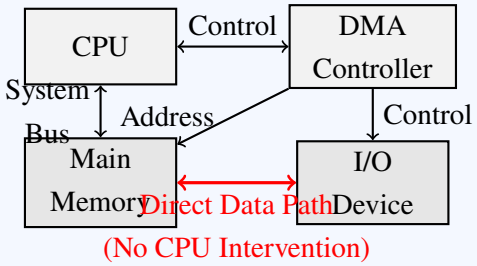
Solution

Concept: Managing data transfers between main memory and peripheral devices is a key task in computer systems. Three main methods are used:

(a) **Programmed I/O (PIO):** The CPU is fully involved in the transfer. It runs a busy-waiting loop, polling the device status registers, and moving data one word at a time through CPU registers.

(b) **Interrupt-Driven I/O:** The CPU initiates the transfer, then works on other tasks. The device sends an interrupt when it is ready, but the CPU must still perform the actual data transfer.

(c) **Direct Memory Access (DMA):** A dedicated hardware DMA controller takes over the system bus to transfer blocks of data directly between the peripheral device and main memory. The CPU is only involved at the start and end of the block transfer.



Solution: Step 1: Evaluate Option A (Reduces memory size): DMA acts as an alternative data path. It does not alter the physical storage capacity of RAM. Thus, Option A is incorrect.

Step 2: Evaluate Option B (Eliminates cache misses): DMA can cause cache coherence issues because it modifies main memory directly without updating the CPU cache. Fixing this requires cache invalidations, which can actually increase cache misses. Thus, Option B is incorrect.

Step 3: Evaluate Option C (Frees the CPU from continuous data transfer operations): In Programmed I/O, the CPU must run a loop for every byte transferred, which uses up processor cycles. With DMA, the CPU only programs the starting address and transfer size into the DMA controller, then switches to other processing tasks. The DMA controller manages the data transfer and interrupts the CPU only when the entire block is transferred. This frees the CPU from continuous data transfer operations. Thus, Option C is correct.

Step 4: Evaluate Option D (Increases disk capacity): DMA is a transfer method and has no effect on the physical capacity of storage disks. Thus, Option D is incorrect.

Final Answer: C

Answer: (C)

Go Back to Question 4

Q5.

Solution

Concept: An unsigned b-bit binary number can represent values from 0 to $2^b - 1$. The minimum number of bits required to represent a positive integer N is the smallest b such that:

$$N < 2^b$$

Equivalently,

$$b = \lfloor \log_2(N) \rfloor + 1$$

Solution: Step 1: Determine the range of values representable with nearby bit lengths:

$$2^{10} - 1 = 1023$$
$$2^{11} - 1 = 2047$$

Step 2: Check whether 10 bits are sufficient: An unsigned 10-bit number can represent values from 0 to 1023. Since

$$2047 > 1023,$$

10 bits are not sufficient.
Step 3: Check whether 11 bits are sufficient: An unsigned 11-bit number can represent values from 0 to 2047. Since

$$2047 \leq 2047,$$

11 bits are sufficient.
Step 4: Verification:

$$2047_{10} = 2^{10} + 2^9 + 2^8 + \cdots + 2^1 + 2^0$$
$$= 11111111111_2$$

Final Answer: $\boxed{11}$

Answer: (B)

Go Back to Question 5

Q6.

Solution

Concept: In hexadecimal (base 16), the symbols A–F represent values 10–15. A hexadecimal number is converted to decimal using:

$$\text{Value} = \sum d_i \, 16^i$$

Also, a k-digit number consisting entirely of the maximum digit $(R-1)$ in base R has value:

$$R^k - 1$$

Solution: Step 1: Write down the given hexadecimal number:

$$(FFFF)_{16}$$

This number has $k = 4$ consecutive digits, all equal to the maximum digit value F = 15.

Step 2: Method 1 — Apply the base R shortcut formula for $R = 16$ and $k = 4$:

$$(FFFF)_{16} = 16^4 - 1$$

Calculate 16^4:

$$\begin{aligned} 16^2 &= 256 \\ 16^3 &= 256 \times 16 = 4096 \\ 16^4 &= 4096 \times 16 = 65,536 \\ (FFFF)_{16} &= 65,536 - 1 = 65,535 \end{aligned}$$

Step 3: Method 2 — Verify by direct polynomial expansion:

$$\begin{aligned} (FFFF)_{16} &= 15 \cdot 16^3 + 15 \cdot 16^2 + 15 \cdot 16^1 + 15 \cdot 16^0 \\ &= 15 \cdot 4096 + 15 \cdot 256 + 15 \cdot 16 + 15 \cdot 1 \\ &= 61,440 + 3840 + 240 + 15 \\ &= 65,280 + 240 + 15 \\ &= 65,520 + 15 = 65,535 \end{aligned}$$

Both methods yield 65, 535, which corresponds to Option B.

Final Answer: 65535

Answer: (B)

Go Back to Question 6

Q7.

Solution

Concept: The Cyclic Redundancy Check (CRC) is an error-detecting block code based on polynomial division over the Galois Field of 2, GF(2).

- A binary message is represented as a polynomial, $M(x)$.
- A fixed generator polynomial, $G(x)$, of degree r is used as the divisor.
- To calculate the CRC, the sender shifts the message polynomial by r positions (appending r zero bits): $M(x) \cdot x^r$.
- This shifted polynomial is divided by $G(x)$ using modulo-2 arithmetic.
- The remainder polynomial, $R(x)$, must have a degree strictly less than r (i.e., $\deg(R(x)) \leq r - 1$).
- This r-bit remainder is appended to the message as the CRC check bits.

Solution: Step 1: Identify the degree of the generator polynomial:

$$\text{Degree } (r) = 5$$

Step 2: Determine the divisor characteristics: A generator polynomial of degree $r = 5$ (such as $G(x) = x^5 + x^2 + 1$) is represented by $r + 1 = 6$ coefficients (the binary divisor `100101`).

Step 3: Determine the remainder characteristics: When dividing any polynomial by a polynomial of degree 5, the remainder polynomial $R(x)$ must have a degree of at most 4:

$$R(x) = c_4x^4 + c_3x^3 + c_2x^2 + c_1x^1 + c_0x^0$$

Since there are 5 coefficients $(c_4, c_3, c_2, c_1, c_0)$, the remainder is represented by exactly 5 bits.

Thus, exactly 5 CRC check bits are appended to the message. This corresponds to Option B.

Final Answer: $\boxed{5}$

Answer: (B)

Go Back to Question 7

Q8.

Solution

Concept: Transmission time is given by:

$$T = \frac{S_{\text{bits}}}{R}$$

where S_{bits} is the file size in bits and R is the transmission rate.

Useful conversions:

$$1 \text{ MB} = 10^6 \text{ Bytes}, \qquad 1 \text{ Byte} = 8 \text{ bits}, \qquad 1 \text{ Mbps} = 10^6 \text{ bits/s}$$

Solution: Step 1: Convert the file size from Megabytes (MB) to bits using decimal metrics [8]:

$$\begin{aligned} S_{\text{Bytes}} &= 5 \text{ MB} = 5 \times 10^6 \text{ Bytes} \\ S_{\text{bits}} &= S_{\text{Bytes}} \times 8 \text{ bits/Byte} \\ &= (5 \times 10^6) \times 8 = 40 \times 10^6 \text{ bits} \end{aligned}$$

Step 2: Convert the transmission rate from Megabits per second (Mbps) to bits per second:

$$R = 10 \text{ Mbps} = 10 \times 10^6 \text{ bits per second}$$

Step 3: Calculate the transmission time (T):

$$\begin{aligned} T &= \frac{S_{\text{bits}}}{R} \\ &= \frac{40 \times 10^6 \text{ bits}}{10 \times 10^6 \text{ bits/second}} \\ &= 4 \text{ seconds} \end{aligned}$$

Step 4: Alternative calculation using binary prefixes: If binary prefixes were used (1 MiB = 2^{20} Bytes = $1,048,576$ Bytes):

$$\begin{aligned} S_{\text{bits}} &= 5 \times 1,048,576 \times 8 = 41,943,040 \text{ bits} \\ T &= \frac{41,943,040 \text{ bits}}{10,000,000 \text{ bits/second}} = 4.194 \text{ seconds} \end{aligned}$$

In standard networking problems, the metric conversion is preferred, which yields exactly 4 seconds. This corresponds to Option B.

Final Answer: $\boxed{4 \text{ s}}$

Answer: (B)

Go Back to Question 8

Q9.

Solution

Concept: Floating-point representations (such as the IEEE-754 standard) use biased exponents to store positive and negative exponents as non-negative integers. This allows exponent comparisons to be performed using simple, unsigned integer comparison hardware.

To convert between the actual exponent (E) and the stored biased exponent (E_{biased}), we use the relation:

$$E_{\text{biased}} = E + \text{Bias}$$

Rearranging to find the actual exponent represented:

$$E = E_{\text{biased}} - \text{Bias}$$

Solution: Step 1: Identify the given biased system values:

- Stored exponent (E_{biased}) = 140
- Bias value = 127

Step 2: Subtract the bias value from the stored exponent:

$$\begin{aligned} E &= 140 - 127 \\ &= 13 \end{aligned}$$

Step 3: Verify the bounds: For single-precision float representation, the biased exponent field is 8 bits wide (0 to 255). The value 140 is within the valid normalized range $[1, 254]$.

The actual exponent represented is 13. This corresponds to Option C.

Final Answer: $\boxed{13}$

Answer: (C)

Go Back to Question 9

Q10.

Solution

Concept: In character encoding schemes (such as Unicode or UTF-32), each distinct character is mapped to a unique integer value called a code point.

For a fixed-width character encoding scheme that allocates exactly n bits for each code point:

- Every bit can exist in one of two binary states (0 or 1).
- The total number of unique binary patterns (or combinations) that can be constructed using n bits is 2^n.
- Thus, the theoretical maximum number of distinct characters or code points that can be represented by the scheme is 2^n.

Solution: Step 1: Identify the bit width of the code points:

$$n = 21 \text{ bits}$$

Step 2: Calculate the theoretical maximum number of distinct code points: Using the combination formula for n bits:

$$\text{Total Code Points} = 2^{21}$$

Step 3: Evaluate the combinatorial range: 2^{21} equates to exactly $2,097,152$ unique values. (Note: This 21-bit limit is the basis of the Unicode code space, which is capped at U+10FFFF to restrict encoding to $1,114,112$ code points).

The theoretical capacity is 2^{21}, which corresponds to Option C.

Final Answer: $\boxed{2^{21}}$

Answer: (C)

Go Back to Question 10

Q11.

Solution

Concept: In demand paging, a page fault occurs when a required page is not present in main memory. The Effective Access Time (EAT) is:

$$\text{EAT} = (1 - p)\, T_{\text{ma}} + p\, T_{\text{pf}}$$

where p is the page-fault rate, T_{ma} is the memory access time, and T_{pf} is the page-fault service time.

Solution: Step 1: Identify the given system parameters and convert them to consistent units:

- Page-fault rate (p) = 0.002
- Normal memory access time (T_{ma}) = 100 ns = 100×10^{-9} s
- Page-fault service time (T_{pf}) = 8 ms = 8×10^{-3} s = 8,000,000 ns

Step 2: Substitute these values into the EAT formula:

$$\text{EAT} = (1 - 0.002) \cdot 100 \text{ ns} + 0.002 \cdot 8,000,000 \text{ ns}$$

Step 3: Compute each component of the sum:

$$\begin{aligned} \text{EAT} &= 0.998 \cdot 100 \text{ ns} + 16,000 \text{ ns} \\ &= 99.8 \text{ ns} + 16,000 \text{ ns} \\ &= 16,099.8 \text{ ns} \end{aligned}$$

Step 4: Convert the result into microseconds (μs): Recall that 1 μs = 1000 ns.

$$\begin{aligned} \text{EAT} &= \frac{16,099.8}{1000}\ \mu\text{s} \\ &\approx 16.1\ \mu\text{s} \end{aligned}$$

The approximate effective access time is 16 μs. This corresponds to Option C.

Final Answer: $\boxed{16\ \mu\text{s}}$

Answer: (C)

Go Back to Question 11

Q12.

Solution

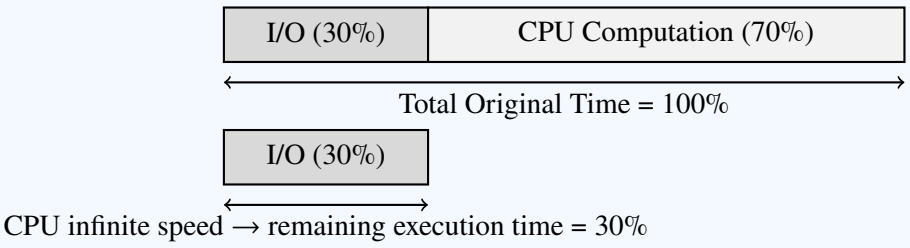
Concept: According to Amdahl's Law,

$$S = \frac{1}{(1-f) + \frac{f}{k}}$$

where f is the fraction improved and k is its speedup factor.

For $k \to \infty$,

$$S_{\max} = \frac{1}{1-f}$$



Solution: Step 1: Identify the sequential and parallelizable components:

- The process spends 30% of its time on I/O, which cannot be enhanced by speeding up the CPU. Hence, the non-enhanced portion is $1 - f = 0.30$.
- The remaining 70% is spent on CPU computation, which is the enhanced portion. Hence, $f = 0.70$.

Step 2: Apply the infinite enhancement condition ($k \to \infty$) to the speedup formula:

$$\begin{aligned} S_{\max} &= \frac{1}{1-f} \\ &= \frac{1}{0.30} \end{aligned}$$

Step 3: Simplify the expression:

$$\begin{aligned} S_{\max} &= \frac{10}{3} \\ &\approx 3.33 \end{aligned}$$

According to Amdahl's Law, even if CPU computation takes zero time, the overall execution time is bounded by the I/O bottleneck, yielding a maximum speedup of 3.33. This corresponds to Option B.

Final Answer: 3.33

Answer: (B)

Go Back to Question 12

Q13.

Solution

Concept: Multiprogramming is a technique designed to increase CPU utilization. Because standard CPU operations are much faster than I/O transfers, a single program running alone would leave the CPU idle during I/O operations. Multiprogramming addresses this by keeping multiple active processes in physical memory (RAM) simultaneously. When the currently executing process waits for an I/O operation to complete, the operating system context-switches the CPU to execute another memory-resident process.

The degree of multiprogramming is a standard operating systems metric defined specifically as:

Degree of Multiprogramming = Number of processes currently residing in main memory

Managing this degree is critical: if it is too low, the CPU may sit idle; if it is too high, the system may experience thrashing due to frequent page faults.

Solution: Step 1: Evaluate each of the provided options:

- **Option A (Number of CPUs in the system):** This defines the degree of multiprocessing, not multiprogramming.
- **Option B (Number of processes residing in memory):** This is the exact definition of the degree of multiprogramming.
- **Option C (Number of files stored on disk):** This is a characteristic of the file system capacity.
- **Option D (Number of I/O devices connected):** This relates to the peripheral hardware configuration.

Thus, Option B is the correct answer.

Final Answer: B

Answer: (B)

Go Back to Question 13

Q14.

Solution

Concept: Secondary storage devices are classified based on their hardware architecture and physical access methods:

- **Sequential Access (e.g., Magnetic Tape):** Data is stored sequentially along the length of a tape. To access a specific record, the tape drive must physically wind through all preceding records, making the access time highly variable and dependent on current tape position.
- **Direct/Random Access (e.g., Hard Disk):** Data is organized on rotating circular platters divided into tracks and sectors. A movable read/write head can go directly to any target track (seek time) and wait for the target sector to rotate underneath (rotational latency).

Because a hard disk can transition to any addressable data block with a relatively low, bounded latency, it possesses direct access capability.

Solution: Step 1: Analyze the needs of a random-access workload: A random-access workload consists of read/write requests to unpredictable addresses across the storage media. To handle this efficiently, the hardware must minimize the time spent traversing unrelated data.

Step 2: Evaluate the options:

- **Option A (Lower storage density):** Lower density is a disadvantage, not a superior feature.
- **Option B (Sequential access mechanism):** This is the mechanism used by magnetic tape, which makes it slow for random access.
- **Option C (Direct access capability):** Hard disks allow the read/write head to move directly to any track/sector, which makes them much faster than tape drives for random-access workloads.
- **Option D (Lower transfer rate):** A lower transfer rate is a disadvantage.

Thus, Option C is correct.

Final Answer: C

Answer: (C)

Go Back to Question 14

Q15.

Solution

Concept: A Boolean function in canonical SOP form is expressed as:

$$F(A, B, C) = \Sigma m(1, 3, 5, 7)$$

Each minterm corresponds to a binary combination of the variables. The function can be simplified using Boolean algebra or a Karnaugh map (K-map).

Solution: Step 1: Write out the algebraic sum of the active minterms:

$$F = \overline{A}\,\overline{B}C + \overline{A}BC + A\overline{B}C + ABC$$

Step 2: Factor out the common variable C from all terms:

$$F = C\left(\overline{A}\,\overline{B} + \overline{A}B + A\overline{B} + AB\right)$$

Step 3: Group and simplify the terms inside the parentheses:

$$\overline{A}\,\overline{B} + \overline{A}B + A\overline{B} + AB = \overline{A}(\overline{B} + B) + A(\overline{B} + B)$$

Since $\overline{B} + B = 1$ by the complement law:

$$= \overline{A}(1) + A(1)$$
$$= \overline{A} + A$$

Since $\overline{A} + A = 1$ by the complement law:

$$= 1$$

Step 4: Substitute this back into the factored expression:

$$F = C \cdot (1)$$
$$= C$$

Step 5: Verify using the K-map: On the K-map, the four minterms form a single group of 4 adjacent cells across the columns where the variable C is 1 ($BC = 01$ and $BC = 11$), while A and B vary. This group simplifies directly to C.

Thus, Option C is correct.

Final Answer: C

Answer: (C)

Go Back to Question 15

Q16.

Solution

Concept: The number of possible distinct Boolean functions of n variables is determined by looking at the structure of its truth table:

- A truth table with n input variables has exactly 2^n rows (unique input combinations).
- For each row, a Boolean function must output either a '0' or a '1'.
- Therefore, a specific Boolean function is defined by a unique combination of 2^n binary outputs.
- The total number of distinct Boolean functions is the number of possible binary output sequences of length 2^n:

$$\begin{aligned}\text{Total Functions} &= 2^{\text{Number of Rows}} \\ &= 2^{2^n}\end{aligned}$$

Solution: Step 1: Determine the number of rows in the truth table for n variables:

$$\text{Rows} = 2^n$$

Step 2: Calculate the number of distinct output assignments: Since each of the 2^n rows can be independently assigned an output value of 0 or 1, the total number of unique functions is:

$$2 \times 2 \times \cdots \times 2 \quad (2^n \text{ times}) = 2^{2^n}$$

Step 3: Evaluate for small values of n:

- For $n = 1$: $2^{2^1} = 2^2 = 4$ functions (Constant 0, Constant 1, Identity A, and Complement $\overline{A}$).
- For $n = 2$: $2^{2^2} = 2^4 = 16$ functions (including AND, OR, NAND, NOR, XOR, etc.).

The expression 2^{2^n} represents the correct scaling behavior, which corresponds to Option C.

Final Answer: $\boxed{2^{2^n}}$

Answer: (C)

Go Back to Question 16

Q17.

Solution

Concept: In digital logic design, a logic gate is classified as a universal gate if any arbitrary Boolean function can be implemented using only combinations of that single gate type.

There are two primary universal gates:

- **NAND Gate** (AND followed by NOT)
- **NOR Gate** (OR followed by NOT)

By combining NAND gates in specific structures, one can construct the fundamental logic operations: NOT, AND, and OR. Because any Boolean expression can be written using only NOT, AND, and OR, it follows that any circuit can be built using only NAND gates.

Solution: Step 1: Evaluate the options:

- **Option A (Associative property):** This is an algebraic property (e.g., $(x \cdot y) \cdot z = x \cdot (y \cdot z)$) which, in fact, the NAND operator does not satisfy.
- **Option B (Universal gate property):** This property defines a gate that can implement all other logic operations (NOT, AND, OR, etc.) without requiring other gate types.
- **Option C (Commutative property):** While the NAND gate is commutative (i.e., A NAND $B = B$ NAND A), commutativity alone does not allow it to implement other logic gates.
- **Option D (Distributive property):** This is another algebraic property and is not the reason a circuit can be built entirely using NAND gates.

Thus, Option B is correct.

Final Answer: Universal gate property

Answer: (B)

Go Back to Question 17

Q18.

Solution

Concept: The Domain Name System (DNS) translates human-readable domain names into machine-readable IP addresses. To improve performance and reduce network traffic, DNS resolvers store resolved queries in a temporary cache.

Each DNS record returned by an authoritative name server includes a metadata field called Time-To-Live (TTL):

- TTL is an integer value expressed in seconds.
- It specifies the duration for which a DNS resolver is allowed to keep the record in its cache.
- Once the TTL timer expires, the resolver must discard the cached record and query the authoritative DNS server again for any subsequent requests.

Solution: Step 1: Evaluate the definition of TTL in DNS: TTL specifies the lifetime of a DNS record within a cache.

Step 2: Contrast this with the options:

- **Option A (Maximum packet size):** This is defined by the Maximum Transmission Unit (MTU) of network interfaces, not DNS TTL.
- **Option B (Time before the cached record expires):** This matches the definition of TTL in DNS.
- **Option C (Time required to establish a connection):** This relates to network latency or TCP handshake times.
- **Option D (DNS query timeout interval):** This is the period a client waits for a response before retrying, which is independent of the TTL.

A TTL of 3600 seconds means the resolver can cache and use the record for exactly 1 hour (3600 seconds) before it expires and must be refreshed. This corresponds to Option B.

Final Answer: B

Answer: (B)

Go Back to Question 18

Q19.

Solution

Concept: To ensure database integrity, Database Management Systems (DBMS) enforce the ACID properties for all transaction executions:

- **Atomicity:** Guarantees that a transaction is treated as a single, indivisible unit of work; either all changes are applied, or none are.
- **Consistency:** Ensures that a transaction brings the database from one valid state to another, maintaining all schema constraints and rules.
- **Isolation:** Guarantees that concurrent execution of transactions results in a system state equivalent to executing them sequentially.
- **Durability:** Guarantees that once a transaction commits, its updates are permanently recorded in non-volatile storage (such as a hard disk) and will not be lost even in the event of a system crash or power failure.

Solution: Step 1: Match the given condition to the correct ACID property: The question asks which property guarantees that the effects of a committed transaction survive system failures.

Step 2: Analyze the durability property: Durability ensures that committed transactions are written to persistent storage. If a system failure occurs immediately after a commit, the recovery manager uses transaction logs to restore or maintain these updates.

Thus, Durability is the property responsible for this guarantee, which corresponds to Option D.

Final Answer: Durability

Answer: (D)

Go Back to Question 19

Q20.

Solution

Concept: In multitasking operating systems, process scheduling algorithms allocate system resources (such as CPU time, memory, or peripheral access) to active processes.

A scheduling algorithm can suffer from starvation (also known as indefinite blocking):

- Starvation occurs when a runnable process is perpetually denied the resources it needs to execute because other processes are repeatedly given higher priority.
- For example, in a strict priority-based scheduling algorithm, a low-priority process may wait indefinitely if high-priority processes are constantly added to the ready queue.
- Starvation is often resolved using techniques like aging, which gradually increases the priority of a process the longer it waits in the queue.

Solution: Step 1: Evaluate each option to identify the definition of starvation:

- **Option A (Two processes wait indefinitely for each other):** This describes a deadlock state, which is distinct from starvation.
- **Option B (A process never receives the resources it requires to proceed):** This is the exact definition of starvation, where a process is ready to execute but is indefinitely bypassed during resource allocation.
- **Option C (A process terminates unexpectedly):** This describes a process crash or abnormal termination.
- **Option D (Main memory becomes completely full):** This is a memory exhaustion state, which can lead to thrashing, but is not starvation.

Thus, Option B is correct.

Final Answer: B

Answer: (B)

Go Back to Question 20

Answer Key

Q	Ans	Q	Ans	Q	Ans	Q	Ans	Q	Ans
1	C	2	C	3	C	4	C	5	B
6	B	7	B	8	B	9	C	10	C
11	C	12	B	13	B	14	C	15	C
16	C	17	B	18	B	19	D	20	B