NIMCET Computer Awareness Sample Paper-2

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 has a 32-bit word length and uses a direct-mapped cache of size 64 KB with a block size of 64 bytes.

How many bits are required for the cache index field in the physical address?

(A) 8

(B) 10

(C) 12

(D) 14

Q2. Consider a processor that supports pipelining.

Which of the following hazards occurs when an instruction depends on the result of a previous instruction that has not yet completed its execution?

(A) Structural Hazard

(B) Control Hazard

(C) Data Hazard

(D) Memory Hazard

Q3. In a microprocessor, the instruction register (IR) plays an important role during instruction execution.

What information is stored in the instruction register immediately after the fetch cycle?

(A) Address of next instruction

(B) Machine code of the current instruction

(C) Memory address of operand

(D) ALU output

Q4. A hard disk rotates at 7200 RPM.

Ignoring seek time and transfer time, what is the average rotational latency of the disk?

(A) 4.17 ms

(B) 8.33 ms

(C) 16.67 ms

(D) 2.08 ms

Q5. A computer uses sign-magnitude representation with 8 bits.

What is the decimal value represented by the binary number

$$10011010$$

?

(A) -26

(B) -102

(C) 26

(D) 102

Q6. The octal number

$$(735)_8$$

is equivalent to which of the following hexadecimal numbers?

(A) $(1DD)_{16}$

(B) $(1ED)_{16}$

(C) $(1FD)_{16}$

(D) $(2DD)_{16}$

Q7. How many distinct values can be represented using a 12-bit unsigned binary number?

Assume all possible bit combinations are valid.

(A) 1024

(B) 2048

(C) 4096

(D) 8192

Q8. The Excess-3 code corresponding to the decimal digit sequence 407 is:

(A) 0111 0011 1010

(B) 0100 0000 0111

(C) 0111 0011 1001

(D) 0101 0011 1010

Q9. A floating-point number system uses 1 sign bit, 7 exponent bits with bias 63, and 8 mantissa bits.

What is the biased exponent stored for an actual exponent of -12?

(A) 51

(B) 63

(C) 75

(D) 12

Q10. The binary number

11011010

is represented in Gray code.

Its equivalent binary number is:

(A) 10010110

(B) 10110110

(C) 10010011

(D) 11110110

Q11. A cache memory has a hit ratio of 95%.

If cache access time is 2 ns and main memory access time is 80 ns, what is the effective memory access time?

(A) 6 ns

(B) 8 ns

(C) 10 ns

(D) 12 ns

Q12. Which of the following memory hierarchies correctly arranges memory components from fastest to slowest access speed?

(A) Registers, Cache, RAM, Disk

(B) Cache, Registers, RAM, Disk

(C) Registers, RAM, Cache, Disk

(D) Cache, RAM, Registers, Disk

Q13. The control unit of a CPU generates signals to coordinate the activities of various hardware components.

Which type of control unit uses fixed logic circuits to generate these control signals?

(A) Microprogrammed Control Unit

(B) Hardwired Control Unit

(C) Distributed Control Unit

(D) Pipeline Control Unit

Q14. Virtual memory allows execution of programs larger than the available physical memory.

Which memory management technique is most commonly associated with virtual memory systems?

(A) Paging

(B) Multiplexing

(C) Spooling

(D) Buffering

Q15. Using Boolean algebra, simplify the expression

$$A + AB + ABC$$

The simplified result is:

(A) AB

(B) ABC

(C) A

(D) $A + B + C$

Q16. The Boolean expression

$$(A + B)(\overline{A} + B)$$

simplifies to:

(A) A

(B) B

(C) AB

(D) $A + B$

Q17. A Boolean function contains six variables.

How many minterms are possible for this function?

(A) 32

(B) 64

(C) 128

(D) 256

Q18. Which of the following Internet protocols is responsible for translating a domain name into its corresponding IP address?

(A) HTTP

(B) SMTP

(C) DNS

(D) ARP

Q19. A relation in a database is said to be in Third Normal Form (3NF) if it is already in 2NF and:

Which additional condition is satisfied?

(A) No repeating groups exist

(B) No partial dependency exists

(C) No transitive dependency exists

(D) Every attribute is a key

Q20. In operating systems, deadlock can occur when several processes compete for resources.

Which of the following is NOT one of the four necessary conditions for deadlock?

(A) Mutual Exclusion

(B) Hold and Wait

(C) Circular Wait

(D) Context Switching

Detailed Solutions

Q1.

Solution

Concept: In a direct-mapped cache, the physical address is partitioned into three fields:

(a) **Block Offset (Word Offset):** Selects a specific byte or word within a cache block.

$$\text{Block Offset bits} = \log_2(\text{Block Size in bytes})$$

(b) **Index:** Identifies the specific cache line (or set) where the block is mapped.

$$\text{Index bits} = \log_2(\text{Number of lines in the cache})$$

$$\text{Number of lines} = \frac{\text{Cache Size}}{\text{Block Size}}$$

(c) **Tag:** Identifies whether the cached block matches the requested memory block.

Solution: Step 1: Identify the given parameters.

- Cache Size = 64 KB $= 64 \times 2^{10}$ Bytes $= 65,536$ Bytes
- Block Size = 64 Bytes

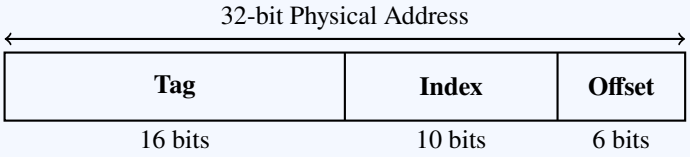
Step 2: Calculate the number of cache lines (blocks).

$$\text{Number of lines} = \frac{64 \text{ KB}}{64 \text{ Bytes}} = 1024 = 2^{10} \text{ lines}$$

Step 3: Determine the number of index bits required to uniquely address 1024 lines.

$$\text{Index bits} = \log_2(1024) = 10 \text{ bits}$$

Step 4: Visualize the 32-bit physical address subdivision.



Since the direct-mapped cache contains 1024 lines, 10 bits are required for the cache index field.

Final Answer: 10

Answer: (B)

Go Back to Question 1

Q2.

Solution

Concept: Pipelining hazards are situations that prevent the next instruction in the instruction stream from executing in its designated clock cycle. They are classified into three types:

- **Structural Hazards:** Arise from resource conflicts when the hardware cannot support all possible combinations of instructions in simultaneous overlapped execution.
- **Control Hazards (Branch Hazards):** Arise from the delay in determining the instruction flow (e.g., branches, jumps).
- **Data Hazards:** Arise when an instruction depends on the results of a previous instruction in a way that is exposed by the overlapping of instructions in the pipeline.

Solution: Step 1: Analyze the dependency described in the question. An instruction requires a data operand that is produced by a preceding instruction. If the preceding instruction has not finished writing back its result, the subsequent instruction cannot read the correct value.

Step 2: Classify the hazard type. This is a standard dependency conflict on data operands (specifically a Read After Write, or RAW, dependency), which is classified as a **Data Hazard**.

Step 3: Match with the options. Option C is Data Hazard.

Final Answer: Data Hazard

Answer: (C)

Go Back to Question 2

Q3.

Solution

Concept: The execution of an instruction is divided into sequential micro-operations:

(a) **Fetch:** The processor retrieves the instruction binary representation (machine code) from the memory location pointed to by the Program Counter (PC).

(b) **Load into Instruction Register (IR):** The fetched binary representation is copied into the Instruction Register (IR).

(c) **Decode:** The control unit interprets the opcode stored in the IR.

(d) **Execute:** The control unit generates signals to perform the action.

Solution: Step 1: Identify the exact action occurring during the fetch cycle. The processor places the address from the PC onto the address bus, reads memory, and receives the raw machine code of the target instruction.

Step 2: Determine where this instruction is stored. The fetched binary data is held in the Instruction Register (IR) so it can be held constant during the decoding phase.

Step 3: Conclude the contents of the IR. Immediately after the fetch cycle, the IR contains the machine code (binary opcode and operands/specifiers) of the current instruction, which corresponds to Option B.

Final Answer: Machine code of the current instruction

Answer: (B)

Go Back to Question 3

Q4.

Solution

Concept: Disk drive performance metrics include:

- **Rotational Speed (R):** Measured in Revolutions Per Minute (RPM).
- **Single Revolution Time (T_{rev}):** The time required for the disk platter to complete one full 360° rotation.

$$T_{\text{rev}} = \frac{60 \text{ seconds}}{R}$$

- **Average Rotational Latency (L_{rot}):** On average, the target sector will be located halfway around the track from the read/write head when the seek completes.

$$L_{\text{rot}} = \frac{1}{2} \times T_{\text{rev}}$$

Solution: Step 1: Given the rotational speed $R = 7200$ RPM, calculate the time for one full revolution.

$$T_{\text{rev}} = \frac{60 \text{ s}}{7200} = \frac{1}{120} \text{ s}$$

Step 2: Convert T_{rev} to milliseconds.

$$T_{\text{rev}} = \left(\frac{1}{120} \times 1000 \right) \text{ ms} = 8.333 \text{ ms}$$

Step 3: Determine the average rotational latency.

$$L_{\text{rot}} = \frac{1}{2} \times 8.333 \text{ ms} = 4.167 \text{ ms} \approx 4.17 \text{ ms}$$

Step 4: Match with the options. Option A represents 4.17 ms.

Final Answer: 4.17 ms

Answer: (A)

Go Back to Question 4

Q5.

Solution

Concept: In an 8-bit signed sign-magnitude representation:

- The most significant bit (MSB, bit 7) represents the sign of the value:

$$b_7 = 0 \implies \text{Positive (+)}$$

$$b_7 = 1 \implies \text{Negative (-)}$$

- The remaining 7 bits (bits 6 through 0) represent the absolute magnitude of the number as an unsigned binary integer.

Solution: Step 1: Write down the given binary sequence and identify its bits.

$$B = 10011010_2$$

$$\text{Sign bit } (b_7) = 1$$

$$\text{Magnitude bits } (b_6 \ldots b_0) = 0011010_2$$

Step 2: Determine the sign of the value. Since the sign bit is 1, the number is negative.
Step 3: Calculate the value of the magnitude bits.

$$0011010_2 = 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

$$= 16 + 8 + 2 = 26$$

Step 4: Combine the sign and magnitude.

$$\text{Decimal Value} = -26$$

This corresponds to Option A.

Final Answer: -26

Answer: (A)

Go Back to Question 5

Q6.

Solution

Concept: Converting from octal (base 8) to hexadecimal (base 16) is performed most easily by using binary (base 2) as an intermediate step:

(a) Convert each octal digit into its equivalent 3-bit binary representation.

(b) Group the resulting binary sequence into blocks of 4 bits starting from the rightmost bit (least significant bit). Pad with leading zeros on the left if necessary.

(c) Convert each 4-bit binary group into its equivalent hexadecimal digit.

Solution: Step 1: Convert $(735)_8$ to its binary equivalent.

$$7 \rightarrow 111_2$$
$$3 \rightarrow 011_2$$
$$5 \rightarrow 101_2$$

$$\text{Binary representation} = 111011101_2$$

Step 2: Regroup the binary representation into 4-bit blocks starting from the right.

$$111011101_2 \rightarrow 1 \quad 1101 \quad 1101_2$$

Step 3: Pad with leading zeros to complete the leftmost block.

$$0001 \quad 1101 \quad 1101_2$$

Step 4: Convert each 4-bit block to hexadecimal.

$$0001_2 = 1_{10} = 1_{16}$$
$$1101_2 = 13_{10} = D_{16}$$
$$1101_2 = 13_{10} = D_{16}$$

$$\text{Hexadecimal equivalent} = (1DD)_{16}$$

This matches Option A.

Final Answer: $\boxed{(1DD)_{16}}$

Answer: (A)

Go Back to Question 6

Q7.

Solution

Concept: An unsigned binary representation of width N bits can encode a total of 2^N unique binary combinations. Because every unique bit combination corresponds to exactly one integer value in the continuous range from 0 to $2^N - 1$, the total number of distinct values represented is equal to the total number of unique binary combinations.

Solution: Step 1: Identify the bit width parameter.

$$N = 12 \text{ bits}$$

Step 2: Calculate the number of unique combinations.

$$\text{Distinct values} = 2^{12}$$

Step 3: Compute the numerical value.

$$2^{10} = 1024 \implies 2^{12} = 1024 \times 4 = 4096$$

Thus, a 12-bit unsigned binary representation can express exactly 4096 unique values, corresponding to Option C.

Final Answer: $\boxed{4096}$

Answer: (C)

Go Back to Question 7

Q8.

Solution

Concept: Excess-3 is a non-weighted, self-complementing Binary Coded Decimal (BCD) code. To represent a multi-digit decimal number in Excess-3:

(a) Take each decimal digit of the number individually.

(b) Add 3 to each digit.

(c) Convert the resulting decimal sum for each digit into its equivalent 4-bit binary representation.

Solution: Step 1: Separate the digits of the sequence 407. The individual decimal digits are 4, 0, and 7.
Step 2: Add 3 to each individual digit.

- First digit: $4 + 3 = 7$
- Second digit: $0 + 3 = 3$
- Third digit: $7 + 3 = 10$

Step 3: Convert each decimal sum into a 4-bit binary sequence.

- $7 \rightarrow 0111_2$
- $3 \rightarrow 0011_2$
- $10 \rightarrow 1010_2$

Step 4: Group the binary nibbles to form the complete Excess-3 code.

$$\text{Excess-3 representation} = 0111 \quad 0011 \quad 1010$$

This corresponds to Option A.

Final Answer: 0111 0011 1010

Answer: (A)

Go Back to Question 8

Q9.

Solution

Concept: In floating-point representation, exponents are stored in a biased format to allow both negative and positive exponents to be represented as unsigned integers. The relationship is given by:

$$\text{Biased Exponent} = \text{Actual Exponent} + \text{Bias}$$

Solution: Step 1: Identify the system parameters.

- Actual Exponent (E) = -12
- Bias = 63

Step 2: Calculate the biased exponent to be stored.

$$\text{Biased Exponent} = -12 + 63 = 51$$

Step 3: Verify the bit width constraint. With 7 exponent bits, the range of unsigned representations is 0 to $2^7 - 1 = 127$. Since 51 lies within $[0, 127]$, it is a valid 7-bit exponent configuration.

Therefore, the biased exponent stored is 51, corresponding to Option A.

Final Answer: $\boxed{51}$

Answer: (A)

Go Back to Question 9

Q10.

Solution

Concept: To convert a Gray code sequence $G = g_{n-1}g_{n-2}\ldots g_0$ to its equivalent binary representation $B = b_{n-1}b_{n-2}\ldots b_0$:

- The most significant bit (MSB) remains identical:

$$b_{n-1} = g_{n-1}$$

- Each subsequent binary bit is determined by XORing the previously computed binary bit with the current Gray code bit:

$$b_i = b_{i+1} \oplus g_i \quad \text{for } i = n-2, \ldots, 0$$

Solution: Step 1: Identify the bits of the given Gray code $G = 11011010_2$.

$$g_7 = 1, \quad g_6 = 1, \quad g_5 = 0, \quad g_4 = 1, \quad g_3 = 1, \quad g_2 = 0, \quad g_1 = 1, \quad g_0 = 0$$

Step 2: Determine each bit of the binary equivalent sequence B step-by-step.

$$b_7 = g_7 = 1$$
$$b_6 = b_7 \oplus g_6 = 1 \oplus 1 = 0$$
$$b_5 = b_6 \oplus g_5 = 0 \oplus 0 = 0$$
$$b_4 = b_5 \oplus g_4 = 0 \oplus 1 = 1$$
$$b_3 = b_4 \oplus g_3 = 1 \oplus 1 = 0$$
$$b_2 = b_3 \oplus g_2 = 0 \oplus 0 = 0$$
$$b_1 = b_2 \oplus g_1 = 0 \oplus 1 = 1$$
$$b_0 = b_1 \oplus g_0 = 1 \oplus 0 = 1$$

Step 3: Combine the bits to write the binary number.

$$B = 10010011_2$$

This matches Option C.

Final Answer: 10010011

Answer: (C)

Go Back to Question 10

Q11.

Solution

Concept: The average (effective) memory access time (AMAT) of a cache-main memory hierarchy can be calculated using two common models:

(a) **Hierarchical (Serial) Access Model:** The processor searches the cache first. On a miss, it then accesses the main memory, incurring a latency penalty:

$$\text{AMAT} = T_{\text{cache}} + (1 - H) \times T_{\text{main}}$$

(b) **Simultaneous (Parallel) Access Model:** The processor starts searching both cache and main memory in parallel:

$$\text{AMAT} = H \times T_{\text{cache}} + (1 - H) \times T_{\text{main}}$$

where H is the cache hit ratio, T_{cache} is the cache access time, and T_{main} is the main memory access time.

Solution: Step 1: Identify the given values.

- Hit ratio (H) = 95% = 0.95
- Miss ratio ($1 - H$) = 5% = 0.05
- Cache access time (T_{cache}) = 2 ns
- Main memory access time (T_{main}) = 80 ns

Step 2: Calculate AMAT under the hierarchical access model.

$$\text{AMAT} = 2 \text{ ns} + (0.05 \times 80 \text{ ns})$$

$$\text{AMAT} = 2 \text{ ns} + 4 \text{ ns} = 6 \text{ ns}$$

Step 3: Calculate AMAT under the simultaneous access model.

$$\text{AMAT} = (0.95 \times 2 \text{ ns}) + (0.05 \times 80 \text{ ns})$$

$$\text{AMAT} = 1.9 \text{ ns} + 4 \text{ ns} = 5.9 \text{ ns} \approx 6 \text{ ns}$$

In both design paradigms, the effective memory access time is rounded to 6 ns, which corresponds to Option A.

Final Answer: 6 ns

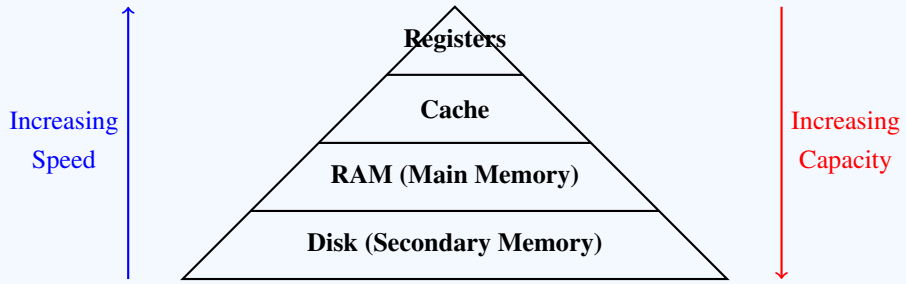
Answer: (A)

Go Back to Question 11

Q12.

Solution

Concept: The storage hierarchy of a computer system organizes memory elements to optimize both cost and processing speed. Components closer to the CPU operate at extremely high speeds but have lower storage capacities and higher cost per bit. As we move down the hierarchy, access speed decreases while capacity increases.



Solution: Step 1: Compare access speeds of the components:

- **Registers:** Located directly inside the CPU; fastest access speed (< 1 ns).
- **Cache:** High-speed static memory (SRAM) near or on the CPU; very fast (1 to 10 ns).
- **RAM (Main Memory):** Dynamic memory (DRAM) connected to the system bus; moderate speed (50 to 100 ns).
- **Disk (Secondary Storage):** Solid-state or magnetic drives; slowest speed (ranging from microseconds to milliseconds).

Step 2: Arrange from fastest to slowest:

$$\text{Registers} \rightarrow \text{Cache} \rightarrow \text{RAM} \rightarrow \text{Disk}$$

This sequence corresponds to Option A.

Final Answer: Registers, Cache, RAM, Disk

Answer: (A)

Go Back to Question 12

Q13.

Solution

Concept: The Control Unit (CU) coordinates instruction fetching and execution by generating timing and control signals. Control units are categorized by implementation style:

- **Hardwired Control Unit:** Implemented using fixed combinational and sequential logic circuits (logic gates, decoders, state flip-flops). It offers maximum speed but lacks flexibility.
- **Microprogrammed Control Unit:** Control signals are stored as microinstructions (firmware) in a dedicated control memory. It is slower but highly flexible and easy to modify.

Solution: Step 1: Read the condition in the question: "uses fixed logic circuits to generate these control signals."

Step 2: Relate the condition to control unit architectures. A design implemented with permanent combinational hardware (fixed logic circuits) is a **Hardwired Control Unit**.

Step 3: Match with the options. This corresponds to Option B.

Final Answer: Hardwired Control Unit

Answer: (B)

Go Back to Question 13

Q14.

Solution

Concept: Virtual memory decouples the logical address space used by a process from the physical address space available in RAM, allowing large processes to run on systems with limited physical memory. Modern virtual memory architectures implement this abstraction primarily through **Paging**.

Solution: Step 1: Analyze the definitions of the listed memory and system management techniques:

- **Paging:** Divides virtual memory into fixed-size blocks called *pages* and physical memory into blocks called *frames*. Pages are loaded into frames dynamically (demand paging), which is the primary driver of virtual memory.
- **Multiplexing:** Shared signal-routing over a physical transmission medium.
- **Spooling (Simultaneous Peripheral Operations On-Line):** Temporarily buffers data in intermediate storage for slower devices (e.g., printers).
- **Buffering:** Holds data in memory temporarily during I/O transfers.

Step 2: Conclude that Paging is the management technique most fundamentally associated with virtual memory.

Final Answer: Paging

Answer: (A)

Go Back to Question 14

Q15.

Solution

Concept: Boolean algebraic expressions can be simplified using basic identities:

- **Distributive Law:** $X(Y + Z) = XY + XZ$
- **Absorption Law:** $X + XY = X$

Solution: Step 1: Write down the given expression.

$$F = A + AB + ABC$$

Step 2: Apply the Absorption Law to the first two terms: $A + AB = A$.

$$F = (A + AB) + ABC = A + ABC$$

Step 3: Factor out the common variable A from the remaining terms.

$$F = A(1 + BC)$$

Step 4: Use the identity property $1 + X = 1$, where $X = BC$:

$$F = A(1) = A$$

Therefore, the simplified result is A, corresponding to Option C.

Final Answer: $\boxed{A}$

Answer: (C)

Go Back to Question 15

Q16.

Solution

Concept: Boolean expressions can be simplified using expansion and fundamental logic laws:

- **Distributive Law of Addition over Multiplication:** $(X + Y)(X + Z) = X + YZ$
- **Complementarity Law:** $A \cdot \overline{A} = 0$
- **Identity Law:** $B + 0 = B$

Solution: Step 1: Write down the expression.

$$F = (A + B)(\overline{A} + B)$$

Step 2: Rearrange the terms to highlight the common variable B.

$$F = (B + A)(B + \overline{A})$$

Step 3: Apply the distributive law:

$$F = B + (A \cdot \overline{A})$$

Step 4: Simplify the term inside the parentheses since a variable and its inverse cannot both be true simultaneously ($A \cdot \overline{A} = 0$).

$$F = B + 0$$

Step 5: Apply the identity property.

$$F = B$$

This matches Option B.

Final Answer: $\boxed{B}$

Answer: (B)

Go Back to Question 16

Q17.

Solution

Concept: In Boolean algebra, a minterm is a product (AND) of all the n variables of the system, where each variable appears either in its normal form or its complemented form. For a Boolean function with n variables, the total number of unique minterms is given by:

$$\text{Total Minterms} = 2^n$$

Solution: Step 1: Identify the number of independent variables n.

$$n = 6$$

Step 2: Calculate the number of possible minterms using the power of 2 formula.

$$\text{Minterms} = 2^6$$

Step 3: Compute the value.

$$2^6 = 64$$

Thus, there are 64 possible minterms for a six-variable Boolean function. This matches Option B.

Final Answer: $\boxed{64}$

Answer: (B)

Go Back to Question 17

Q18.

Solution

Concept: The Internet Protocol suite includes specialized application-layer and network-layer protocols to coordinate communication:

- **HTTP (Hypertext Transfer Protocol):** Manages the transfer of web page content.
- **SMTP (Simple Mail Transfer Protocol):** Governs the transmission of email.
- **DNS (Domain Name System):** Acts as a directory service to resolve human-readable hostnames to numeric IP addresses.
- **ARP (Address Resolution Protocol):** Maps a known IP address to a local physical MAC address.

Solution: Step 1: Read the protocol requirement: "translating a domain name into its corresponding IP address."

Step 2: Match with the protocol descriptions. The Domain Name System (DNS) performs name resolution.

Step 3: Match with the options. DNS corresponds to Option C.

Final Answer: DNS

Answer: (C)

Go Back to Question 18

Q19.

Solution

Concept: Database normalization organizes relational database attributes to minimize redundancy:

- **1NF (First Normal Form):** Eliminates repeating groups; all attributes must contain atomic values.
- **2NF (Second Normal Form):** Must be in 1NF and have no partial dependencies (every non-prime attribute must be fully functionally dependent on the entire primary key).
- **3NF (Third Normal Form):** Must be in 2NF and have no transitive dependencies (no non-prime attribute should transitively depend on a key through another non-prime attribute).

Solution: Step 1: Identify the requirement for 3NF given that the relation is already in 2NF.

Step 2: Recall the additional condition for 3NF. Under 3NF, non-key attributes must not functionally determine other non-key attributes (meaning no transitive dependencies exist).

Step 3: Match with the options. Option C is the correct condition.

Final Answer: No transitive dependency exists

Answer: (C)

Go Back to Question 19

Q20.

Solution

Concept: A deadlock is a state where a set of processes are blocked because each process holds a resource and waits for another resource held by some other process. According to Coffman, deadlocks require four simultaneous conditions:

(a) **Mutual Exclusion:** At least one resource must be held in a non-shareable mode.

(b) **Hold and Wait:** A process must be holding at least one resource and waiting to acquire additional resources held by other processes.

(c) **No Preemption:** Resources cannot be forcibly taken from a process holding them; they must be released voluntarily.

(d) **Circular Wait:** A closed chain of processes exists, where each process holds one or more resources needed by the next process in the chain.

Solution: Step 1: Review the four Coffman conditions (Mutual Exclusion, Hold and Wait, No Preemption, and Circular Wait).

Step 2: Evaluate the options. Options A, B, and C are essential requirements for deadlock.

Step 3: Evaluate **Context Switching** (Option D). Context switching is the multitasking mechanism the CPU uses to save the state of a running process and restore the state of another. It does not cause or contribute to deadlock.

Final Answer: Context Switching

Answer: (D)

Go Back to Question 20

Answer Key

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