NIMCET Computer Awareness Sample Paper-4

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 uses a 40-bit physical address space and a direct-mapped cache of size 256 KB. The cache block size is 64 bytes.

How many bits of the physical address are used as the tag field?

(A) 20

(B) 22

(C) 24

(D) 26

Q2. A processor executes a program containing 20% branch instructions. The branch predictor has an accuracy of 90%, and every misprediction incurs a penalty of 4 clock cycles.

Assuming an ideal CPI of 1, what is the effective CPI of the processor?

(A) 1.04

(B) 1.08

(C) 1.12

(D) 1.16

Q3. A microprocessor contains a 16-bit Program Counter (PC).

Ignoring segmentation and paging, what is the maximum number of distinct memory locations that can be directly addressed by the PC?

(A) 2^8

(B) 2^{12}

(C) 2^{16}

(D) 2^{32}

Q4. In a non-pipelined processor, each instruction requires 5 clock cycles to complete.

After introducing an ideal 5-stage pipeline with no hazards, what is the theoretical maximum speedup for a very large number of instructions?

(A) 2

(B) 3

(C) 4

(D) 5

Q5. The decimal number -73 is represented using 8-bit two's complement notation.

What is its binary representation?

(A) 10110111

(B) 10110110

(C) 11010111

(D) 11011001

Q6. The hexadecimal number

$$(7A3F)_{16}$$

is converted into binary form.

How many binary digits equal to 1 appear in the resulting binary representation?

(A) 8

(B) 9

(C) 10

(D) 11

Q7. A communication system uses a 7-bit character code.

Assuming all bit combinations are valid, how many unique characters can be represented using this code?

(A) 64

(B) 96

(C) 128

(D) 256

Q8. The binary number

10101110

is transmitted over a channel using odd parity.

What parity bit should be appended to ensure odd parity?

(A) 0

(B) 1

(C) Either 0 or 1

(D) Cannot be determined

Q9. An IEEE-754 single precision floating-point number uses 1 sign bit, 8 exponent bits and 23 fraction bits.

Which of the following quantities determines the range of representable values?

(A) Fraction field only

(B) Sign bit only

(C) Exponent field primarily

(D) Mantissa field only

Q10. The Excess-3 code corresponding to the decimal number 582 is represented using 12 bits.

Which of the following bit patterns is correct?

(A) 1000 1011 0101

(B) 1000 1011 0110

(C) 1001 1010 0101

(D) 1000 1100 0101

Q11. A page size of 4 KB is used in a virtual memory system.

If the logical address space consists of 32 bits, how many bits are required for the page number field?

(A) 12

(B) 18

(C) 20

(D) 22

Q12. A memory chip is organized as 4096×8.

What is the storage capacity of the chip?

(A) 4 KB

(B) 8 KB

(C) 16 KB

(D) 32 KB

Q13. In a multicore processor, each core has a private L1 cache while sharing an L3 cache.

What is the primary purpose of cache coherence protocols?

(A) Increase clock speed

(B) Reduce instruction length

(C) Maintain consistency of shared data

(D) Reduce RAM size

Q14. A hard disk has 1024 cylinders, 16 heads and 63 sectors per track. Each sector stores 512 bytes.

What is the approximate storage capacity of the disk?

(A) 504 MB

(B) 528 MB

(C) 1 GB

(D) 256 MB

Q15. Using Boolean algebra, simplify the expression

$$AB + \overline{A}B + A\overline{B}$$

The simplified result is:

(A) $A + B$

(B) AB

(C) A

(D) B

Q16. If a Boolean function of four variables contains all possible minterms except one, then the function evaluates to 0 for how many input combinations?

(A) 0

(B) 1

(C) 4

(D) 15

Q17. A combinational circuit has 5 input variables.

How many rows will be present in its complete truth table?

(A) 16

(B) 25

(C) 32

(D) 64

Q18. An IPv4 address consists of 32 bits.

How many distinct IPv4 addresses are theoretically possible?

(A) 2^{16}

(B) 2^{24}

(C) 2^{32}

(D) 2^{64}

Q19. In a relational database, a candidate key is defined as a minimal set of attributes that uniquely identifies each tuple.

Which of the following statements is always true regarding a primary key?

(A) Every primary key is a candidate key

(B) Every candidate key is a primary key

(C) A primary key may contain duplicate values

(D) A primary key can be NULL

Q20. A process enters a deadlock state because four necessary conditions are simultaneously satisfied.

Which one of the following conditions, if eliminated, guarantees that deadlock can never occur?

(A) Mutual Exclusion

(B) Hold and Wait

(C) Circular Wait

(D) Any one of the above

Detailed Solutions

Q1.

Solution

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

$$\text{Physical Address} = \text{Tag} + \text{Index} + \text{Block Offset}$$

where the field bit-widths are calculated as:

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

$$\text{Index bits} = \log_2\left(\frac{\text{Cache Size}}{\text{Block Size}}\right)$$

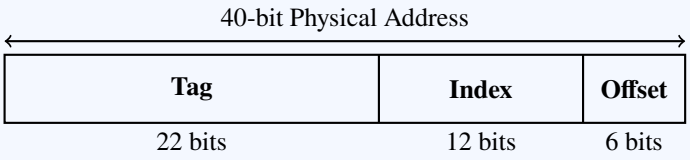
Solution: Step 1: Calculate the block offset and index bits from the given values:

- Block Size = 64 B = 2^6 B $\implies$ Block Offset = 6 bits
- Number of lines = $\frac{\text{Cache Size}}{\text{Block Size}} = \frac{256\text{ KB}}{64\text{ B}} = \frac{2^{18}\text{ B}}{2^6\text{ B}} = 2^{12} \implies$ Index = 12 bits

Step 2: Calculate the remaining tag bits by subtraction from the 40-bit address:

$$\text{Tag bits} = 40 - (\text{Index bits} + \text{Block Offset bits})$$

$$\text{Tag bits} = 40 - (12 + 6) = 22 \text{ bits}$$



Final Answer: 22

Answer: (B)

Go Back to Question 1

Q2.

Solution

Concept: The effective Cycles Per Instruction (CPI) of a processor is the sum of its ideal baseline CPI and any penalty cycles introduced by instruction stalls or mispredictions:

$$\text{Effective CPI} = \text{Ideal CPI} + \text{Stall cycles per instruction}$$

The average stall cycles added per instruction is calculated as:

$$\text{Stall cycles per instruction} = \text{Instruction Frequency} \times \text{Condition Frequency} \times \text{Penalty}$$

Solution: Step 1: Identify the parameters of the program execution.

- Ideal baseline CPI = 1
- Branch instruction frequency = 20% = 0.20
- Branch prediction accuracy = 90% = 0.90
- Branch misprediction rate = 100% − 90% = 10% = 0.10
- Misprediction penalty = 4 clock cycles

Step 2: Calculate the stall cycles introduced on average by mispredicted branch instructions.

$$\text{Stall cycles} = \text{Branch Frequency} \times \text{Misprediction Rate} \times \text{Misprediction Penalty}$$

$$\text{Stall cycles} = 0.20 \times 0.10 \times 4 = 0.08 \text{ cycles}$$

Step 3: Calculate the effective CPI.

$$\text{Effective CPI} = 1 + 0.08 = 1.08$$

This matches Option B.

Final Answer: 1.08

Answer: (B)

Go Back to Question 2

Q3.

Solution

Concept: The Program Counter (PC) stores the memory address of the next instruction to be fetched. The maximum physical addressing capability of a processor register (without additional segmentation or paging mapping mechanisms) is determined directly by the width N of the register:

$$\text{Maximum directly addressable locations} = 2^N$$

Solution: Step 1: Identify the given PC register width.

$$N = 16 \text{ bits}$$

Step 2: Apply the binary addressing formula.

$$\text{Total addresses} = 2^{16}$$

Step 3: Evaluate. With a 16-bit register width, the program counter can generate unique binary patterns ranging from `0000000000000000` to `1111111111111111`, allowing it to directly address exactly 2^{16} (or $65,536$) distinct memory locations.
This matches Option C.

Final Answer: $\boxed{2^{16}}$

Answer: (C)

Go Back to Question 3

Q4.

Solution

Concept: The speedup (S) achieved by a pipelined processor over a non-pipelined processor is defined as:

$$S = \frac{\text{Time}_{\text{non-pipelined}}}{\text{Time}_{\text{pipelined}}}$$

For executing a large number of instructions N under ideal conditions (with no hazard stalls and matching clock period τ):

- Non-pipelined execution time: $\text{Time}_{\text{non-pipelined}} = N \times k \times \tau$
- Pipelined execution time: $\text{Time}_{\text{pipelined}} = (k + N - 1) \times \tau$

where k is the number of pipeline stages.

Solution: Step 1: Identify the parameters of the system.

- Number of cycles per non-pipelined instruction = $k = 5$
- Number of pipeline stages = 5

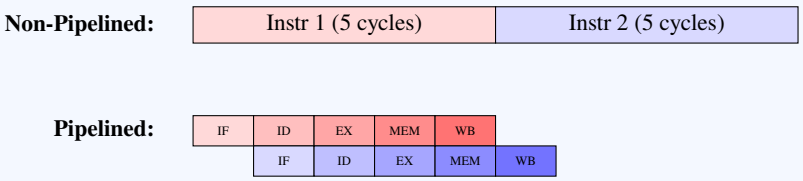
Step 2: Express the theoretical speedup ratio.

$$S = \frac{N \times 5 \times \tau}{(5 + N - 1) \times \tau} = \frac{5N}{N + 4}$$

Step 3: Calculate the limit as the number of instructions N approaches infinity (a very large number of instructions).

$$\lim_{N \to \infty} S = \lim_{N \to \infty} \frac{5N}{N + 4} = 5$$

Step 4: Visualize the throughput improvement.



The speedup asymptotically approaches the number of pipeline stages, which is 5. This corresponds to Option D.

Final Answer: 5

Answer: (D)

Go Back to Question 4

Q5.

Solution

Concept: To find the 8-bit two's complement representation of a negative decimal number $-X$:

(a) Convert the absolute magnitude $+X$ to an 8-bit unsigned binary number.

(b) Invert all the bits to obtain the one's complement.

(c) Add 1 to the one's complement.

Solution: Step 1: Convert the positive magnitude +73 into an 8-bit binary representation.

$$73 = 64 + 8 + 1 = 2^6 + 2^3 + 2^0 \implies 01001001_2$$

Step 2: Invert all bits to get the one's complement.

$$\text{One's Complement} = 10110110_2$$

Step 3: Add 1 to get the two's complement representation.

$$10110110_2 + 1 = 10110111_2$$

Step 4: Verify the result using powers of two weights:

$$-128 + 32 + 16 + 4 + 2 + 1 = -128 + 55 = -73$$

The correct binary representation is 10110111, which corresponds to Option A.

Final Answer: 10110111

Answer: (A)

Go Back to Question 5

Q6.

Solution

Concept: To convert a hexadecimal number to its binary equivalent, each hexadecimal digit is represented as a unique 4-bit binary sequence. The hexadecimal digit assignments are:

$$0 \ldots 9 \rightarrow \text{standard values}, \quad A = 10, \quad B = 11, \quad C = 12, \quad D = 13, \quad E = 14, \quad F = 15$$

Solution: Step 1: Convert each digit of $(7A3F)_{16}$ individually into its 4-bit binary group:

$$7 \rightarrow 0111_2$$
$$A \rightarrow 1010_2$$
$$3 \rightarrow 0011_2$$
$$F \rightarrow 1111_2$$

Step 2: Combine the groups to construct the complete binary representation.

$$\text{Binary equivalent} = 0111101000111111_2$$

Step 3: Count the number of binary digits that are equal to 1:

- From 0111_2: Three 1s (1, 1, 1)
- From 1010_2: Two 1s (1, 1)
- From 0011_2: Two 1s (1, 1)
- From 1111_2: Four 1s (1, 1, 1, 1)

$$\text{Total number of 1s} = 3 + 2 + 2 + 4 = 11$$

This matches Option D.

Final Answer: $\boxed{11}$

Answer: (D)

Go Back to Question 6

Q7.

Solution

Concept: An N-bit binary code can represent a total of 2^N unique states. In a character encoding scheme where every possible bit combination is defined as a valid character, the total number of distinct characters that can be represented is equal to the total number of unique states.

Solution: Step 1: Identify the code width.

$$N = 7 \text{ bits}$$

Step 2: Calculate the number of unique combinations.

$$\text{Unique characters} = 2^7$$

Step 3: Evaluate.

$$2^7 = 128$$

With a 7-bit character code (such as standard ASCII), exactly 128 unique characters can be represented. This corresponds to Option C.

Final Answer: $\boxed{128}$

Answer: (C)

Go Back to Question 7

Q8.

Solution

Concept: Parity checks are used to detect errors during data transmission:

- **Odd Parity:** The total number of 1s in the entire codeword (data bits plus the appended parity bit) must be an **odd** number.
- **Even Parity:** The total number of 1s in the entire codeword must be an **even** number.

Solution: Step 1: Count the number of 1s in the given binary message data 10101110.

$$\text{Data} = 1, 0, 1, 0, 1, 1, 1, 0$$

$$\text{Number of 1s in the data} = 5$$

Step 2: Check if the current count matches the target parity. The number 5 is already an odd number.

Step 3: Determine the required parity bit value. Since the total number of 1s in the transmitted codeword must be odd, we should not add another 1. Therefore, the parity bit to be appended is 0.

$$\text{Codeword} = 101011100 \implies 5 \text{ ones (odd)}$$

This corresponds to Option A.

Final Answer: $\boxed{0}$

Answer: (A)

Go Back to Question 8

Q9.

Solution

Concept: The standard IEEE-754 single-precision floating-point format partitions its 32 bits into three fields:

(a) **Sign Bit (1 bit):** Determines the sign (positive or negative) of the value.

(b) **Exponent (8 bits):** Determines the dynamic scale or **range** of the values that can be represented (from smallest magnitude to largest magnitude).

(c) **Fraction/Mantissa (23 bits):** Determines the precision or accuracy (number of significant digits) of the representation.

Solution: Step 1: Analyze the question. We need to identify which field defines the limits of representation magnitude (the range).

Step 2: Compare field purposes.

- The fraction determines precision.
- The sign determines direction on the number line.
- The exponent scales the base value ($2^{\text{exponent}-\text{bias}}$) and is the primary factor in determining how large or small of a number can be represented.

Step 3: Conclude that the exponent field primarily determines the range of representable values, corresponding to Option C.

Final Answer: Exponent field primarily

Answer: (C)

Go Back to Question 9

Q10.

Solution

Concept: To represent a multi-digit decimal number in Excess-3:

(a) Treat each decimal digit of the sequence individually.

(b) Add 3 to each digit.

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

Solution: Step 1: Separate the digits of the decimal value 582. The individual digits are 5, 8, and 2.

Step 2: Add 3 to each of the individual digits.

- First digit: $5 + 3 = 8$
- Second digit: $8 + 3 = 11$
- Third digit: $2 + 3 = 5$

Step 3: Convert each decimal sum into its 4-bit binary group.

- $8 \rightarrow 1000_2$
- $11 \rightarrow 1011_2$
- $5 \rightarrow 0101_2$

Step 4: Combine the binary nibbles sequentially to obtain the 12-bit pattern.

$$\text{Excess-3 representation} = 1000 \quad 1011 \quad 0101$$

This corresponds to Option A.

Final Answer: 1000 1011 0101

Answer: (A)

Go Back to Question 10

Q11.

Solution

Concept: In a virtual memory system with paging, the logical address generated by the CPU is divided into two parts:

(a) **Page Offset (d):** Identifies the specific byte within the page.

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

(b) **Page Number (p):** Identifies the specific page in virtual memory.

$$\text{Page Number bits} = \text{Logical Address width} - \text{Page Offset bits}$$

Solution: Step 1: Identify the given system parameters.

- Logical Address Space = 32 bits
- Page Size = 4 KB = 4×1024 Bytes = 4096 Bytes = 2^{12} Bytes

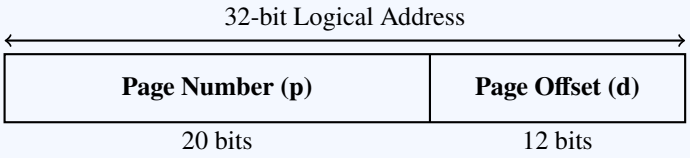
Step 2: Calculate the number of bits required for the page offset.

$$\text{Page Offset bits} = \log_2(2^{12}) = 12 \text{ bits}$$

Step 3: Calculate the number of bits required for the page number.

$$\text{Page Number bits} = 32 - 12 = 20 \text{ bits}$$

Step 4: Visualize the 32-bit logical address breakdown.



Therefore, 20 bits are required for the page number field, which corresponds to Option C.

Final Answer: 20

Answer: (C)

Go Back to Question 11

Q12.

Solution

Concept: A memory chip organized as $M \times N$ contains:

- M addressable locations (words).
- N bits stored in each location (word size).
- The total storage capacity is calculated as:

$$\text{Total Bits} = M \times N \text{ bits}$$

$$\text{Total Bytes} = \frac{M \times N}{8} \text{ Bytes}$$

We convert the byte value to standard memory units using: 1 KB = 1024 Bytes.

Solution: Step 1: Identify the given memory organization parameters.

- Number of locations (M) = 4096
- Bits per location (N) = 8 bits (which is exactly 1 Byte)

Step 2: Calculate the total storage capacity in bytes.

$$\text{Capacity} = 4096 \times 8 \text{ bits} = 4096 \text{ Bytes}$$

Step 3: Convert the capacity to Kilobytes (KB).

$$\text{Capacity in KB} = \frac{4096 \text{ Bytes}}{1024 \text{ Bytes/KB}} = 4 \text{ KB}$$

This matches Option A.

Final Answer: 4 KB

Answer: (A)

Go Back to Question 12

Q13.

Solution

Concept: In symmetric multiprocessing (SMP) systems with multicore processors, each processor core typically contains its own private high-speed cache memory (such as L1). When multiple cores cache local copies of the same shared physical memory block, modifications to one copy will make the other copies stale.

Cache coherence protocols (e.g., MESI, MSI, snooping-based, or directory-based protocols) are hardware mechanisms designed to track the state of cached blocks and synchronize updates to prevent processors from reading outdated or inconsistent data.

Solution: Step 1: Analyze the definitions of the choices:

- **Option A (Increase clock speed):** Clock speed is determined by physical transistor properties and pipeline depth.
- **Option B (Reduce instruction length):** Instruction length is fixed by the instruction set architecture (ISA).
- **Option C (Maintain consistency of shared data):** Coherence protocols monitor the read/write actions of all cores on shared variables to ensure that all caches see a consistent view of memory.
- **Option D (Reduce RAM size):** Physical memory size is independent of caching protocols.

Step 2: Identify the primary purpose of cache coherence protocols as maintaining the consistency of shared data. This corresponds to Option C.

Final Answer: Maintain consistency of shared data

Answer: (C)

Go Back to Question 13

Q14.

Solution

Concept: The storage capacity of a hard disk drive using CHS (Cylinder-Head-Sector) geometry is calculated by multiplying the physical coordinates of the drive and the size of a single sector:

$$\text{Capacity in Bytes} = \text{Cylinders} \times \text{Heads} \times \text{Sectors per Track} \times \text{Bytes per Sector}$$

We can express the capacity in Megabytes (MB) using either:

- **Binary definition (MiB):** $1\ \text{MB} = 2^{20}\ \text{Bytes} = 1,048,576\ \text{Bytes}$
- **Decimal definition (MB):** $1\ \text{MB} = 10^{6}\ \text{Bytes} = 1,000,000\ \text{Bytes}$

Solution: Step 1: Identify the disk parameters.

- Cylinders = 1024
- Heads = 16
- Sectors per track = 63
- Bytes per sector = 512

Step 2: Calculate the total capacity in bytes.

$$\text{Capacity} = 1024 \times 16 \times 63 \times 512 \text{ bytes}$$

$$\text{Capacity} = 528,482,304 \text{ bytes}$$

Step 3: Convert the byte capacity to megabytes.

- Using the binary definition (commonly used by operating systems like MS-DOS/Windows):

$$\text{Capacity} = \frac{528,482,304 \text{ bytes}}{1,048,576 \text{ bytes/MB}} = 504 \text{ MB}$$

- Using the decimal definition (commonly used by hardware manufacturers):

$$\text{Capacity} = \frac{528,482,304 \text{ bytes}}{1,000,000 \text{ bytes/MB}} = 528.48 \text{ MB} \approx 528 \text{ MB}$$

This dual calculation represents the famous 504 MB (binary) / 528 MB (decimal) BIOS addressing barrier of early PC systems. Because both options are provided, the binary interpretation of 504 MB represents the standard software programming metric (Option A), while 528 MB represents the hardware decimal metric (Option B).

Final Answer: 504 MB

Answer: (A)

Go Back to Question 14

Q15.

Solution

Concept: Boolean expressions can be simplified using basic algebraic laws, such as:

- **Distributive Law:** $XY + XZ = X(Y + Z)$
- **Inverse/Complementarity Law:** $A + \overline{A} = 1$
- **Identity Law:** $X \cdot 1 = X$

Solution: Step 1: Write down the given expression.

$$F = AB + \overline{A}B + A\overline{B}$$

Step 2: Group the first two terms and factor out the common variable B.

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

Step 3: Apply the inverse identity $A + \overline{A} = 1$:

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

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

Step 4: Apply the distributive law of addition over multiplication $(X + YZ = (X + Y)(X + Z))$ with $X = B, Y = A, Z = \overline{B}$:

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

Step 5: Simplify using the inverse identity $B + \overline{B} = 1$:

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

$$F = A + B$$

The simplified result is $A + B$, which corresponds to Option A.

Final Answer: $\boxed{A + B}$

Answer: (A)

Go Back to Question 15

Q16.

Solution

Concept: In Boolean algebra, a truth table has a row for every possible input combination of the variables. For a function with n variables:

- There are 2^n unique input combinations.
- There are 2^n possible minterms.
- Each unique minterm evaluates to 1 for exactly one specific input combination and 0 for all other combinations.

Solution: Step 1: Determine the total number of possible input combinations for a four-variable system ($n = 4$).

$$\text{Total Combinations} = 2^4 = 16$$

Step 2: Relate the minterms to the function outputs. If a function contains a particular minterm, it evaluates to 1 for that combination. If it contains all possible minterms except one, it evaluates to 1 for 15 combinations and 0 for the remaining combination.
Step 3: Count the combinations where the function evaluates to 0.

$$\text{Combinations evaluating to } 0 = 16 - 15 = 1$$

Thus, the function evaluates to 0 for exactly 1 input combination, which corresponds to Option B.

Final Answer: 1

Answer: (B)

Go Back to Question 16

Q17.

Solution

Concept: A truth table systematically lists all possible input combinations of a logical circuit and the corresponding output values. For a combinational circuit with n binary input variables, the number of distinct input combinations is given by:

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

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

$$n = 5$$

Step 2: Apply the input combinations formula to find the number of rows.

$$\text{Number of Rows} = 2^5$$

Step 3: Compute the value.

$$2^5 = 32$$

Thus, the complete truth table will contain exactly 32 rows, corresponding to Option C.

Final Answer: $\boxed{32}$

Answer: (C)

Go Back to Question 17

Q18.

Solution

Concept: An IP address is a binary number that uniquely identifies a network interface. Since every bit of an IP address can independently hold a value of either 0 or 1, a binary address of length N bits can theoretically produce a total of 2^N unique address allocations.

Solution: Step 1: Identify the bit length of an IPv4 address.

$$N = 32 \text{ bits}$$

Step 2: Apply the binary configuration formula.

$$\text{Theoretical distinct addresses} = 2^{32}$$

With 32 bits, there are exactly 2^{32} (or approximately 4.29 billion) unique IPv4 addresses possible. This corresponds to Option C.

Final Answer: $\boxed{2^{32}}$

Answer: (C)

Go Back to Question 18

Q19.

Solution

Concept: Key constraints in relational database design:

- **Candidate Key:** A minimal set of attributes that uniquely identifies each tuple in a relation.
- **Primary Key:** The specific candidate key chosen by the database designer to uniquely identify tuples in a table. It must contain unique values and cannot contain NULL values.

Solution: Step 1: Evaluate each statement based on relational database theory.

- **Option A (Every primary key is a candidate key):** True. The primary key is chosen directly from the set of candidate keys, which makes it a candidate key by definition.
- **Option B (Every candidate key is a primary key):** False. Only one candidate key is chosen as the primary key. The remaining candidate keys are alternate keys.
- **Option C (A primary key may contain duplicate values):** False. By definition, a primary key must maintain uniqueness to identify tuples without ambiguity.
- **Option D (A primary key can be NULL):** False. Entity integrity constraints state that no primary key attribute can contain a NULL value.

Step 2: Conclude that Option A is always true.

Final Answer: Every primary key is a candidate key

Answer: (A)

Go Back to Question 19

Q20.

Solution

Concept: According to Coffman, a deadlock state can occur if and only if the following four conditions are simultaneously satisfied in a system:

(a) **Mutual Exclusion:** Resources cannot be shared.

(b) **Hold and Wait:** Processes hold allocated resources while waiting for new ones.

(c) **No Preemption:** Resources cannot be forcibly stripped from a process.

(d) **Circular Wait:** A closed loop of processes exists where each process waits for a resource held by the next.

Because these conditions are all necessary, a deadlock state cannot exist if even one of these conditions is eliminated.

Solution: Step 1: Understand the core design principles of deadlock prevention. To prevent deadlocks, the system designer must construct protocols that ensure at least one of the four necessary conditions can never be met.

Step 2: Analyze the given choices. Eliminating Mutual Exclusion (e.g., spooling), eliminating Hold and Wait (e.g., requesting all resources upfront), or eliminating Circular Wait (e.g., enforcing a resource ordering) will each independently guarantee that a deadlock cannot occur.

Step 3: Select the correct option. Since eliminating *any one* of the conditions is sufficient to prevent deadlocks, the correct choice is Option D (Any one of the above).

Final Answer: Any one of the above

Answer: (D)

Go Back to Question 20

Answer Key

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