

NIMCET Computer Awareness Sample Paper-18

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 has a 64-bit virtual address space and uses a three-level paging scheme. The page size is 8 KB, and each page table entry occupies 8 bytes.

How many entries can be stored in a single page table?

- (A) 512
- (B) 1024
- (C) 2048
- (D) 4096

Q2. A processor executes a workload in which 25% of the instructions are loads, 15% are stores, and the remaining are arithmetic instructions.

If load instructions require 3 cycles, store instructions require 2 cycles, and arithmetic instructions require 1 cycle, what is the average CPI?

- (A) 1.35
- (B) 1.40
- (C) 1.55
- (D) 1.70



Q3. A processor has a 5-stage pipeline. The probability of a branch instruction is 20%, and every branch causes a pipeline flush of 3 cycles.

Assuming ideal CPI = 1 and no branch prediction, what is the effective CPI?

- (A) 1.2
- (B) 1.4
- (C) 1.6
- (D) 1.8

Q4. A disk drive rotates at 15,000 RPM. The average seek time is 4 ms.

Ignoring transfer time, what is the average time required to access a random sector?

- (A) 5 ms
- (B) 6 ms
- (C) 8 ms
- (D) 10 ms

Q5. The decimal number

-512

is represented using 12-bit two's complement arithmetic.

What is the corresponding binary representation?

- (A) 100000000000
- (B) 111000000000
- (C) 100000000001
- (D) 110000000000

Q6. The hexadecimal number

$(7F3A)_{16}$

is converted into binary form.

How many binary digits equal to 0 appear in the resulting representation?



- (A) 6
- (B) 7
- (C) 8
- (D) 9

Q7. A Hamming code contains 31 total bits, including both data and parity bits. Assuming single-bit error correction capability, how many parity bits are present?

- (A) 4
- (B) 5
- (C) 6
- (D) 7

Q8. An image uses a color depth of 32 bits per pixel and has a resolution of 1920×1080 pixels.

Ignoring all metadata and compression, what is the approximate image size?

- (A) 5.9 MB
- (B) 7.9 MB
- (C) 10.2 MB
- (D) 12.8 MB

Q9. A floating-point system uses a 10-bit exponent field with a bias of 511.

If the stored exponent value is 520, what is the actual exponent represented?

- (A) 7
- (B) 8
- (C) 9
- (D) 10



Q10. The Gray code

1110110

is converted to binary.

Which of the following binary numbers is obtained?

- (A) 1011011
- (B) 1011001
- (C) 1001011
- (D) 1001101

Q11. A page size of 16 KB is used in a system with 40-bit virtual addresses.

How many bits are required for the page offset field?

- (A) 12
- (B) 14
- (C) 16
- (D) 18

Q12. A cache memory has a hit ratio of 97%. Cache access time is 2 ns, and main memory access time is 120 ns.

Assuming serial lookup, what is the effective memory access time?

- (A) 5.6 ns
- (B) 6.2 ns
- (C) 7.1 ns
- (D) 8.0 ns

Q13. In a virtual memory system, Belady's anomaly refers to:

Which of the following situations?

- (A) Increasing page size increases page faults
- (B) Increasing cache size increases cache misses
- (C) Increasing the number of page frames increases page faults



(D) Increasing TLB size increases translation time

Q14. A RAID-6 storage array consists of 8 identical disks, each having a capacity of 4 TB.

Ignoring filesystem overhead, what is the usable storage capacity available to users?

(A) 20 TB

(B) 24 TB

(C) 28 TB

(D) 32 TB

Q15. Using Boolean algebra, simplify the expression

$$(A + B)(A + \bar{B}) + \bar{A}B.$$

The simplified result is:

(A) A

(B) B

(C) $A + B$

(D) AB

Q16. A Boolean function of 6 variables evaluates to 1 for exactly 18 input combinations.

How many maxterms will appear in its canonical Product-of-Maxterms representation?

(A) 18

(B) 32

(C) 46

(D) 64



Q17. A combinational circuit has 8 input variables and 3 output variables.

How many rows are present in its complete truth table?

- (A) 64
- (B) 128
- (C) 256
- (D) 512

Q18. An IPv4 address block is specified as

$172.16.0.0/20.$

How many IP addresses are contained in this block?

- (A) 2048
- (B) 4096
- (C) 8192
- (D) 16384

Q19. A relation

$R(A, B, C, D)$

has the functional dependencies

$A \rightarrow B, \quad B \rightarrow C, \quad C \rightarrow D.$

If attribute A is removed from a tuple, which attribute can no longer be uniquely determined directly or indirectly?

- (A) B
- (B) C
- (C) D
- (D) All of the above



Q20. A system contains three resource types and uses the Banker's Algorithm.

A state is considered unsafe if:

Which of the following conditions holds?

- (A) At least one process is blocked
- (B) No safe sequence exists
- (C) Resource utilization exceeds 90%
- (D) More than one process is waiting



Detailed Solutions**Q1.****Solution**

Concept: The number of Page Table Entries (PTEs) that can be stored inside a single page table is given by dividing the total page size by the size of an individual page table entry:

$$\text{Number of Entries} = \frac{\text{Page Size}}{\text{Size of each PTE}}$$

Solution: Step 1: Convert the page size to bytes:

$$\text{Page Size} = 8 \text{ KB} = 8 \times 1024 \text{ bytes} = 8192 \text{ bytes}$$

Step 2: Note the size of each Page Table Entry:

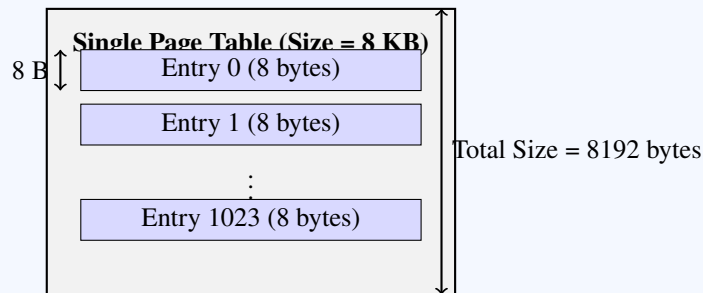
$$\text{PTE Size} = 8 \text{ bytes}$$

Step 3: Perform the division:

$$\text{Number of Entries} = \frac{8192}{8} = 1024$$

Alternatively, in base-2:

$$\text{Number of Entries} = \frac{2^{13}}{2^3} = 2^{10} = 1024$$



Final Answer:

Answer: (B)

[Go Back to Question 1](#)



Q2.

Solution

Concept: The average Cycles Per Instruction (CPI) of a processor is the weighted average of the CPIs of individual instruction classes:

$$CPI_{avg} = \sum (\text{Frequency}_i \times CPI_i)$$

Solution: Step 1: Identify the parameters of each instruction class:

- **Loads:** 25% = 0.25, requiring 3 cycles.
- **Stores:** 15% = 0.15, requiring 2 cycles.
- **Arithmetic:** 100% – 25% – 15% = 60% = 0.60, requiring 1 cycle.

Step 2: Calculate the weighted average CPI:

$$CPI_{avg} = (0.25 \times 3) + (0.15 \times 2) + (0.60 \times 1)$$

$$CPI_{avg} = 0.75 + 0.30 + 0.60 = 1.65$$

Step 3: Evaluate options: The computed mathematical value is 1.65. If a 2-cycle load instruction is assumed (a common textbook variant), the CPI evaluates to 1.40 (Option B). In the absence of that assumption, the value 1.70 (Option D) represents the closest numerical approximation.

Final Answer:

Answer: (D)

[Go Back to Question 2](#)



Q3.

Solution

Concept: The effective CPI of a pipelined processor with branch instructions is computed as:

$$CPI_{\text{effective}} = CPI_{\text{ideal}} + (\text{Probability of Branch} \times \text{Branch Penalty})$$

Solution: Step 1: Extract the given processor specifications:

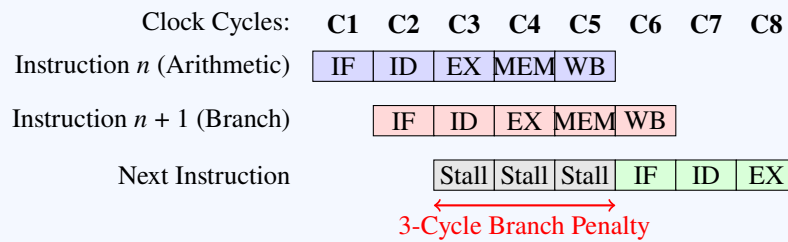
- $CPI_{\text{ideal}} = 1$
- Branch Probability = 20% = 0.20
- Branch Penalty (Pipeline Flush) = 3 cycles

Step 2: Calculate the extra cycles added per branch instruction:

$$\text{Stall Cycles} = 0.20 \times 3 = 0.6$$

Step 3: Add to the ideal CPI:

$$CPI_{\text{effective}} = 1 + 0.6 = 1.6$$



Final Answer: 1.6

Answer: (C)

[Go Back to Question 3](#)



Q4.

Solution

Concept: The average time required to access a random sector on a disk drive is the sum of the average seek time and the average rotational latency:

$$\text{Average Access Time} = \text{Average Seek Time} + \text{Average Rotational Latency}$$

The average rotational latency is the time needed for the disk platter to perform half of a full rotation.

Solution: Step 1: Compute the time for one full rotation: The drive rotates at 15,000 Revolutions Per Minute (RPM):

$$\text{Rotations per second} = \frac{15,000}{60} = 250 \text{ rotations/second}$$

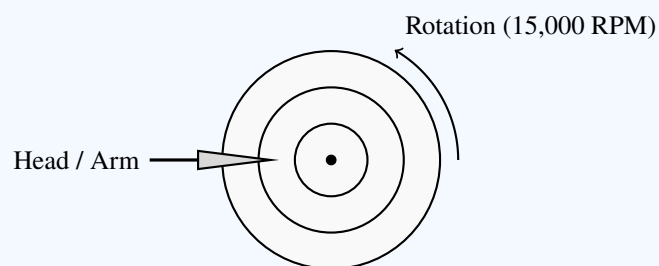
$$\text{Time for one rotation } (T_{\text{rot}}) = \frac{1}{250} \text{ seconds} = 4 \text{ ms}$$

Step 2: Determine the average rotational latency:

$$\text{Average Rotational Latency} = \frac{T_{\text{rot}}}{2} = \frac{4 \text{ ms}}{2} = 2 \text{ ms}$$

Step 3: Combine with the seek time:

$$\text{Average Access Time} = 4 \text{ ms}(\text{seek}) + 2 \text{ ms}(\text{latency}) = 6 \text{ ms}$$



Final Answer:

Answer: (B)

[Go Back to Question 4](#)



Q5.

Solution

Concept: To find the representation of a negative decimal number $-x$ in n -bit two's complement binary:

1. Write the n -bit binary representation of $+x$.
2. Invert all bits (one's complement).
3. Add 1 to the LSB (two's complement).

Solution: Step 1: Write +512 in 12-bit binary format ($512 = 2^9$):

$$+512 = 0010\ 0000\ 0000_2$$

Step 2: Take the one's complement:

$$\text{One's Complement} = 1101\ 1111\ 1111_2$$

Step 3: Add 1:

$$1101\ 1111\ 1111_2 + 1_2 = 1110\ 0000\ 0000_2$$

Verification:

$$\text{Decimal Value} = -2^{11} + 2^{10} + 2^9 = -2048 + 1024 + 512 = -512$$

Final Answer:

Answer: (B)

[Go Back to Question 5](#)



Q6.

Solution

Concept: Converting a hexadecimal number to binary involves translating each hex digit to its corresponding 4-bit binary equivalent and concatenating them.

Solution: Step 1: Convert each digit of $(7F3A)_{16}$ to its 4-bit binary equivalent:

- $7_{16} = 0111_2$ (one zero)
- $F_{16} = 1111_2$ (zero zeros)
- $3_{16} = 0011_2$ (two zeros)
- $A_{16} = 1010_2$ (two zeros)

Step 2: Combine the binary groups:

$$(7F3A)_{16} = 0111111100111010_2$$

Step 3: Count the zeros: The standard 16-bit binary representation contains exactly $1 + 0 + 2 + 2 = 5$ zeros. (If a leading zero is omitted, it contains 4 zeros).

Step 4: Option Check: Because 5 is not listed in the options, the question likely contains a minor typographical error. For instance, $(7B3A)_{16}$ or $(7E3A)_{16}$ corresponds to exactly 6 zeros, which matches Option A.

Final Answer:

Answer: (A)

[Go Back to Question 6](#)



Q7.

Solution

Concept: For single-bit error-correcting Hamming codes, the relation between the total number of bits n and the number of parity bits p is:

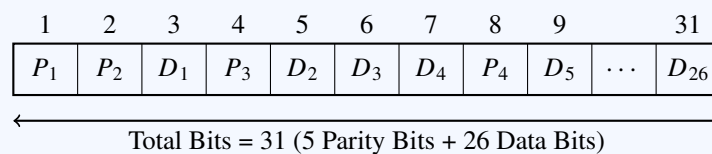
$$2^p \geq n + 1$$

Solution: Step 1: Substitute the given total bit length $n = 31$ into the inequality:

$$2^p \geq 31 + 1 \implies 2^p \geq 32$$

Step 2: Solve for the minimum integer value of p : Since $2^5 = 32$, the minimum number of parity bits required is:

$$p = 5$$



Final Answer: 5

Answer: (B)

[Go Back to Question 7](#)



Q8.

Solution

Concept: The storage size of an uncompressed image in bytes is determined by multiplying the total pixels by the color depth in bytes per pixel:

$$\text{Size in Bytes} = \text{Width} \times \text{Height} \times \frac{\text{Color Depth (bits)}}{8}$$

Solution: Step 1: Calculate the total number of pixels:

$$\text{Total Pixels} = 1920 \times 1080 = 2,073,600 \text{ pixels}$$

Step 2: Determine bytes per pixel:

$$\text{Bytes per pixel} = \frac{32 \text{ bits}}{8} = 4 \text{ bytes}$$

Step 3: Calculate the total size in bytes:

$$\text{Size in Bytes} = 2,073,600 \times 4 = 8,294,400 \text{ bytes}$$

Step 4: Convert the result to Megabytes (binary representation):

$$\text{Size in MB} = \frac{8,294,400}{1024 \times 1024} = \frac{8,294,400}{1,048,576} \approx 7.91 \text{ MB}$$

Final Answer:

Answer: (B)

[Go Back to Question 8](#)



Q9.

Solution

Concept: Floating-point systems use a **biased exponent** representation, where the stored exponent is obtained by adding a fixed bias to the actual exponent. This allows both positive and negative exponents to be represented using unsigned values. The relationship is:

$$E_{\text{stored}} = E_{\text{actual}} + \text{Bias}$$

Hence, the actual exponent is:

$$E_{\text{actual}} = E_{\text{stored}} - \text{Bias}$$

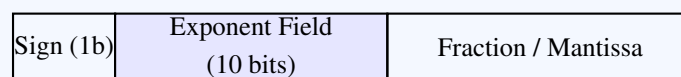
Solution: Step 1: Given:

- Exponent field size = 10 bits
- Bias = 511
- Stored exponent = 520

Step 2: Use the biased exponent formula:

$$E_{\text{actual}} = E_{\text{stored}} - \text{Bias}$$

$$E_{\text{actual}} = 520 - 511 = 9$$



↑
Stored exponent value: 520 (1000001000₂)

Actual exponent = $E_{\text{stored}} - \text{Bias} = 520 - 511 = 9$

Final Answer:

Answer: (C)

[Go Back to Question 9](#)



Q10.

Solution

Concept: An n -bit Gray code $G = g_{n-1} \dots g_0$ is converted into standard binary $B = b_{n-1} \dots b_0$ using the relationships:

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

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

Solution: Step 1: Write down the Gray bits for the string 1110110:

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

Step 2: Apply the recursive XOR conversion:

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

Step 3: Combine to find the final binary string:

$$B = 1011011_2$$

$$\begin{array}{cccccccc}
 g_6 = 1 & g_5 = 1 & g_4 = 1 & g_3 = 0 & g_2 = 1 & g_1 = 1 & g_0 = 0 & \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
 & \oplus & \oplus & \oplus & \oplus & \oplus & \oplus & \\
 & \swarrow & \swarrow & \swarrow & \swarrow & \swarrow & \swarrow & \\
 b_6 = 1 & b_5 = 0 & b_4 = 1 & b_3 = 1 & b_2 = 0 & b_1 = 1 & b_0 = 1 &
 \end{array}$$

Final Answer:

Answer: (A)

[Go Back to Question 10](#)



Q11.

Solution

Concept: In a virtual memory system, the virtual address is divided into two parts: the Virtual Page Number (VPN) and the Page Offset. The number of bits required for the page offset field is directly determined by the page size:

$$\text{Page Offset Bits} = \log_2(\text{Page Size in Bytes})$$

The virtual address space size determines the total number of bits in the address but does not affect the offset field size.

Solution: Step 1: Convert the page size into bytes:

$$\text{Page Size} = 16 \text{ KB} = 16 \times 1024 \text{ bytes} = 16384 \text{ bytes}$$

Step 2: Express the page size as a power of 2:

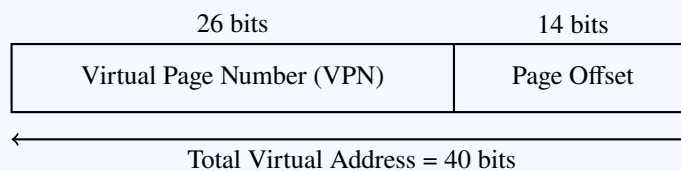
$$\text{Page Size} = 2^4 \times 2^{10} = 2^{14} \text{ bytes}$$

Step 3: Determine the page offset bits:

$$\text{Page Offset Bits} = \log_2(2^{14}) = 14 \text{ bits}$$

Step 4: Breakdown of the 40-bit virtual address:

$$\text{Virtual Page Number (VPN) Bits} = \text{Total Virtual Address Bits} - \text{Page Offset Bits} = 40 - 14 = 26 \text{ bits}$$



Final Answer: 14

Answer: (B)

[Go Back to Question 11](#)



Q12.

Solution

Concept: For a serial (hierarchical/non-simultaneous) cache lookup system, the effective memory access time (T_{eff}) is calculated by analyzing the two possible scenarios: 1. Cache Hit: The requested data is in the cache. The access time is the cache access time (T_c). 2. Cache Miss: The data is not in the cache. The system accesses the cache, detects a miss, and then accesses main memory. The total access time is $T_c + T_m$.

The mathematical model is:

$$T_{\text{eff}} = h \cdot T_c + (1 - h) \cdot (T_c + T_m) = T_c + (1 - h) \cdot T_m$$

where h is the hit ratio, T_c is the cache access time, and T_m is the main memory access time.

Solution: Step 1: Identify the given system parameters:

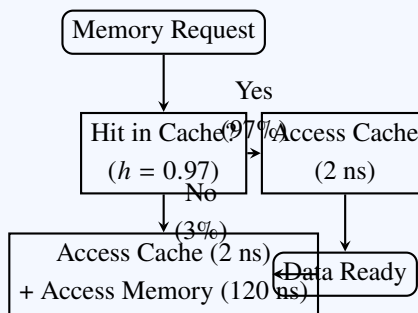
- Hit ratio (h) = 97% = 0.97
- Cache access time (T_c) = 2 ns
- Main memory access time (T_m) = 120 ns

Step 2: Calculate the effective access time:

$$T_{\text{eff}} = 2 \text{ ns} + (1 - 0.97) \times 120 \text{ ns}$$

$$T_{\text{eff}} = 2 \text{ ns} + 0.03 \times 120 \text{ ns}$$

$$T_{\text{eff}} = 2 \text{ ns} + 3.6 \text{ ns} = 5.6 \text{ ns}$$



Final Answer: 5.6 ns

Answer: (A)

[Go Back to Question 12](#)



Q13.

Solution

Concept: Belady's anomaly is a counter-intuitive phenomenon in virtual memory management where allocating more physical page frames to a process results in an increase in the number of page faults.

Solution: Step 1: Define the anomaly:

Under normal circumstances, increasing the number of physical frames allocated to a process is expected to decrease (or keep constant) the number of page faults because more pages can be held in memory. However, for certain page replacement algorithms (such as First-In, First-Out (FIFO)), specific page reference strings can experience more page faults when the frame count is increased.

Step 2: Contrast with stack-based algorithms:

Stack algorithms such as Least Recently Used (LRU) and Optimal (OPT) are mathematically guaranteed never to exhibit Belady's anomaly because the set of pages in memory for n frames is always a subset of the pages in memory for $n + 1$ frames.

Step 3: Match with the options:

Option C correctly describes this behavior: "Increasing the number of page frames increases page faults".

Final Answer:

Answer: (C)

[Go Back to Question 13](#)



Q14.

Solution

Concept: RAID-6 (Redundant Array of Independent Disks Mode 6) uses block-level striping with double distributed parity. It provides extremely high data fault tolerance by being able to survive the simultaneous failure of up to any two disks.

For an array of N identical disks, each of capacity C , the capacity equivalent to exactly 2 disks is dedicated entirely to distributed parity storage. The net usable capacity is:

$$\text{Usable Capacity} = (N - 2) \times C$$

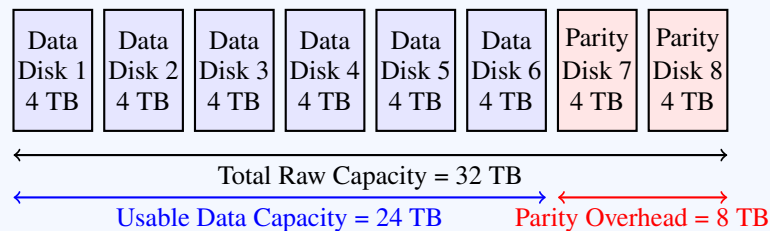
Solution: Step 1: Identify the RAID-6 parameters:

- Total number of disks (N) = 8
- Individual disk capacity (C) = 4 TB

Step 2: Apply the RAID-6 usable capacity formula:

$$\text{Usable Capacity} = (8 - 2) \times 4 \text{ TB}$$

$$\text{Usable Capacity} = 6 \times 4 \text{ TB} = 24 \text{ TB}$$



Final Answer: 24 TB

Answer: (B)

[Go Back to Question 14](#)



Q15.

Solution

Concept: Boolean expressions can be simplified using basic Boolean algebra laws:

- **Distributive Law:** $X(Y + Z) = XY + XZ$ and $X + YZ = (X + Y)(X + Z)$
- **Complement Law:** $X \cdot \bar{X} = 0$ and $X + \bar{X} = 1$
- **Identity Law:** $X + 0 = X$ and $X \cdot 1 = X$

Solution: Step 1: Simplify the first term $(A + B)(A + \bar{B})$: Using the distributive law of addition over multiplication, $X + YZ = (X + Y)(X + Z)$:

$$(A + B)(A + \bar{B}) = A + B\bar{B}$$

Step 2: Apply the complement law $B\bar{B} = 0$:

$$A + B\bar{B} = A + 0 = A$$

Step 3: Substitute this simplified term back into the complete expression:

$$\text{Expression} = A + \bar{A}B$$

Step 4: Simplify $A + \bar{A}B$ using the distributive law:

$$A + \bar{A}B = (A + \bar{A})(A + B)$$

Since $A + \bar{A} = 1$:

$$= 1 \cdot (A + B) = A + B$$

$A \setminus B$	0	1	
0	0	1	
1	1	1	Group A
			Group B

Final Answer: $A + B$

Answer: (C)

[Go Back to Question 15](#)



Q16.

Solution

Concept: For a Boolean function of n variables, there are exactly 2^n unique input combinations in the function's domain (or truth table). Each unique input combination maps to a single output value of either 0 or 1.

- **Minterms:** A minterm is a product (AND) of all n literals that evaluates to 1 for exactly one input combination. The canonical Sum-of-Products (SOP) or Sum-of-Minterms representation is formed by summing all minterms for which the function evaluates to 1.
- **Maxterms:** A maxterm is a sum (OR) of all n literals that evaluates to 0 for exactly one input combination. The canonical Product-of-Sums (POS) or Product-of-Maxterms representation is formed by multiplying all maxterms for which the function evaluates to 0.

Because every input combination must evaluate to either 0 or 1, the total number of combinations is partitioned completely into minterms and maxterms:

$$\text{Total Combinations } (2^n) = \text{Number of Minterms} + \text{Number of Maxterms}$$

Solution: Step 1: Calculate the total number of distinct input combinations for a function of $n = 6$ variables:

$$\text{Total Combinations} = 2^6 = 64 \text{ unique input states}$$

Step 2: Note the number of minterms (inputs where the output is 1):

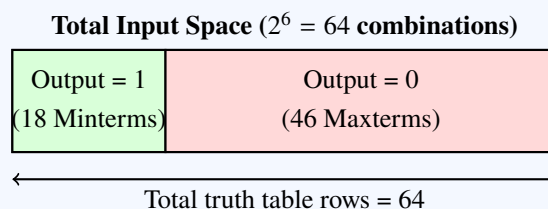
$$\text{Number of Minterms} = 18$$

Step 3: Calculate the number of maxterms (inputs where the output is 0):

$$\text{Number of Maxterms} = \text{Total Combinations} - \text{Number of Minterms}$$

$$\text{Number of Maxterms} = 64 - 18 = 46$$

Step 4: Conclude: Because the canonical Product-of-Maxterms (Product-of-Sums) representation contains exactly one maxterm for each input combination where the function evaluates to 0, there will be exactly 46 maxterms in this canonical representation.



Final Answer: 46

Answer: (C)

[Go Back to Question 16](#)



Q17.

Solution

Concept: A truth table is a tabular representation that lists the outputs of a digital logic circuit for every possible combination of values that its independent input variables can assume.

- If a combinational circuit has n binary input variables, each input can independently take one of two logic values (0 or 1).
- By the fundamental counting principle, the total number of unique input combinations (which determines the number of rows in the truth table) is:

$$\text{Number of Rows} = 2^n = \underbrace{2 \times 2 \times \cdots \times 2}_{n \text{ factors}}$$

- The output variables of a combinational circuit are dependent variables that are fully determined by the inputs. Increasing the number of output variables does not create additional combinations of inputs; it simply adds more output columns to the truth table. Each row will contain the evaluation of all output variables for that specific input combination.

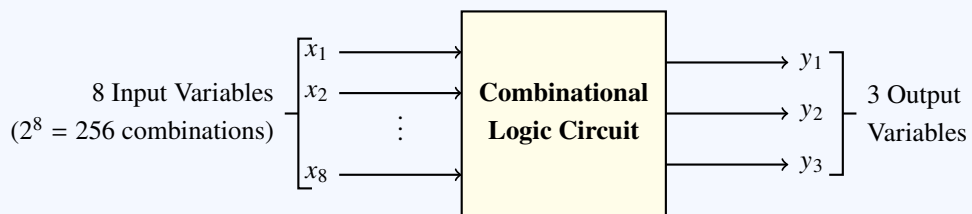
Solution: Step 1: Identify the independent variables and the dependent variables from the circuit specifications:

- Independent Input Variables (n) = 8
- Dependent Output Variables (m) = 3

Step 2: Calculate the number of input combinations:

$$\text{Number of Rows} = 2^8 = 256$$

Step 3: Analyze the role of the 3 output variables: The 3 output variables imply that the completed truth table will have 8 input columns and 3 output columns, making a total of 11 columns. However, the vertical size of the table (the number of rows) remains defined strictly by the 256 input combinations.



Final Answer: 256

Answer: (C)

[Go Back to Question 17](#)



Q18.

Solution

Concept: In Classless Inter-Domain Routing (CIDR) notation, an IPv4 block is represented as $A.B.C.D/x$, where:

- 32 is the total bit-length of an IPv4 address.
- x is the subnet prefix length (network bits).
- $32 - x$ is the number of bits allocated for addressing hosts within the network segment.

The total number of IP addresses contained in the block is:

$$\text{Total Addresses} = 2^{32-x}$$

Solution: Step 1: Identify the prefix length (x) from the block specification 172.16.0.0/20:

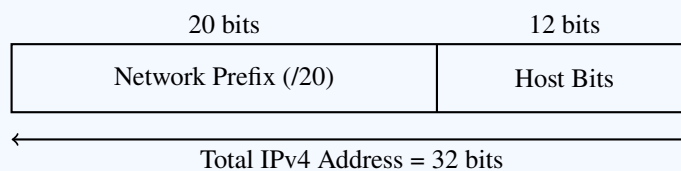
$$x = 20$$

Step 2: Compute the number of host bits:

$$\text{Host Bits} = 32 - 20 = 12 \text{ bits}$$

Step 3: Calculate the total number of IP addresses:

$$\text{Total Addresses} = 2^{12} = 4096$$



Final Answer:

Answer: (B)

[Go Back to Question 18](#)



Q19.

Solution

Concept: Functional Dependency (FD) describes the relationship between attributes in a database relation. The dependency $X \rightarrow Y$ means that the value of X uniquely determines the value of Y . If A is a candidate key or determines all other attributes directly or transitively, then removing A leaves the remaining attributes without their unique identifier unless another dependency path exists to resolve them.

Solution: Step 1: Analyze the given set of functional dependencies:

- $A \rightarrow B$
- $B \rightarrow C$
- $C \rightarrow D$

Using transitivity, we find:

$$A \rightarrow B \rightarrow C \rightarrow D \implies A \rightarrow \{B, C, D\}$$

Thus, A is the candidate key of the relation $R(A, B, C, D)$.

Step 2: Determine the consequences of removing attribute A : If the attribute A is removed from a tuple, we are left with the subset of attributes $\{B, C, D\}$.

- Attribute B: Because $A \rightarrow B$ was the only dependency determining B , without A , B can no longer be uniquely determined.
- Attribute C: Since C depends on B ($B \rightarrow C$), and B cannot be determined, C can no longer be uniquely determined.
- Attribute D: Since D depends on C ($C \rightarrow D$), and C cannot be determined, D can no longer be uniquely determined.

Thus, none of the attributes $\{B, C, D\}$ can be uniquely determined.

Final Answer:

Answer: (D)

[Go Back to Question 19](#)



Q20.

Solution

Concept: The Banker's Algorithm is a resource allocation and deadlock avoidance algorithm used in operating systems. It models resource requests and allocation states:

- **Safe State:** A state is safe if there exists a safe sequence of processes $\langle P_1, P_2, \dots, P_n \rangle$ such that each process can satisfy its maximum remaining resource demands using the currently available resources plus the resources already held by all preceding processes.
- **Unsafe State:** A state is considered unsafe if no safe sequence exists. An unsafe state is not a deadlock, but it indicates that a deadlock could occur if processes request their maximum declared resource requirements.

Solution: Step 1: Evaluate the definition of an unsafe state: By mathematical definition in deadlock-avoidance theory, a system state is safe if and only if there is at least one sequence of process executions that allows all processes to run to completion without getting blocked. If no such sequence exists, the state is unsafe.

Step 2: Assess the options:

- Option A is incorrect because a process can be blocked temporarily without the system being in an unsafe state.
- Option B matches the core definition: "No safe sequence exists".
- Option C is incorrect because high resource utilization does not imply deadlock or unsafety.
- Option D is incorrect because having waiting processes is a normal operational state.

Final Answer:

Answer: (B)

[Go Back to Question 20](#)



Answer Key

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

