NIMCET Computer Awareness Sample Paper-17

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 1 MB cache organized as a 16-way set associative cache. The block size is 64 bytes.

How many sets are present in the cache?

(A) 512

(B) 1024

(C) 2048

(D) 4096

Q2. A processor executes instructions using a 6-stage pipeline. Due to branch hazards, the pipeline is stalled for 2 cycles on 5% of all instructions.

Assuming the ideal CPI is 1, what is the effective CPI?

(A) 1.05

(B) 1.10

(C) 1.15

(D) 1.20

Q3. A processor has 32 general-purpose registers and uses fixed-length instructions.

How many bits are required to uniquely identify a register?

(A) 4

(B) 5

(C) 6

(D) 8

Q4. A CPU uses memory-mapped I/O.

Which of the following is a consequence of using memory-mapped I/O instead of isolated I/O?

(A) Separate address space for I/O devices

(B) I/O devices can be accessed using ordinary load/store instructions

(C) Cache memory becomes unnecessary

(D) DMA cannot be used

Q5. The binary number

$$110101101011_2$$

is converted directly into hexadecimal notation.

Which of the following hexadecimal numbers is obtained?

(A) D6A

(B) DAB

(C) E6B

(D) CAB

Q6. A signed 12-bit two's complement number has the binary representation

$$111110101011.$$

What is its decimal value?

(A) -85

(B) -91

(C) -101

(D) -117

Q7. A communication system uses an 8-bit checksum for error detection.

What is the maximum unsigned decimal value that can be represented by the checksum field?

(A) 127

(B) 255

(C) 511

(D) 1023

Q8. An image has a resolution of 1024×768 pixels and uses 24 bits per pixel.

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

(A) 1.5 MB

(B) 2.25 MB

(C) 3 MB

(D) 4.5 MB

Q9. In IEEE-754 single precision format, the fraction (mantissa) field contains 23 explicitly stored bits.

Including the hidden leading bit for normalized numbers, how many significant binary digits of precision are available?

(A) 22

(B) 23

(C) 24

(D) 32

Q10. A system uses ASCII encoding.

If 7-bit ASCII is employed, how many distinct character codes can be represented?

(A) 64

(B) 96

(C) 128

(D) 256

Q11. A virtual memory system uses a Translation Lookaside Buffer (TLB). The TLB hit ratio is 95%.

If a TLB lookup requires 10 ns and a memory access requires 100 ns, what is the effective address translation time?

(A) 105 ns

(B) 110 ns

(C) 115 ns

(D) 120 ns

Q12. A process arrives at time 0 with a CPU burst time of 20 ms.

If Round Robin scheduling with a time quantum of 4 ms is used and there are no other processes, how many context switches will occur before the process completes?

(A) 3

(B) 4

(C) 5

(D) 0

Q13. In a RAID-1 configuration consisting of four identical disks of capacity 500 GB each, what is the total usable storage capacity?

(A) 500 GB

(B) 1 TB

(C) 2 TB

(D) 4 TB

Q14. A hard disk controller uses CHS (Cylinder-Head-Sector) addressing. The disk contains 1024 cylinders, 64 heads, and 32 sectors per track.

Ignoring overhead, how many sectors are present on the disk?

(A) 1,048,576

(B) 2,097,152

(C) 4,194,304

(D) 8,388,608

Q15. The Boolean expression

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

is equivalent to which of the following?

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

(B) $A + B$

(C) AB

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

Q16. A Boolean function of 5 variables contains exactly 10 minterms.

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

(A) 10

(B) 22

(C) 32

(D) 42

Q17. Which of the following logic gates produces an output of 1 if and only if an odd number of inputs are equal to 1?

(A) NAND

(B) NOR

(C) XOR

(D) XNOR

Q18. An IPv6 address contains 128 bits. If the network prefix length is /64, how many bits remain available for interface identifiers?

(A) 32

(B) 48

(C) 64

(D) 96

Q19. A transaction reads a data item, performs computations, and writes the modified value back to the database.

Which ACID property ensures that concurrent transactions do not interfere with one another's intermediate states?

(A) Atomicity

(B) Consistency

(C) Isolation

(D) Durability

Q20. In an operating system, thrashing occurs when:

Choose the most appropriate description.

(A) CPU utilization becomes 100%

(B) The system spends most of its time servicing page faults rather than executing processes

(C) All processes enter the blocked state

(D) Deadlock occurs among all processes

Detailed Solutions

Q1.

Solution

Concept: The number of sets in a set-associative cache is determined by the total number of blocks (cache lines) and the associativity (number of blocks per set).

$$\text{Number of Sets} = \frac{\text{Total Cache Size}}{\text{Block Size} \times \text{Associativity}}$$

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

$$\text{Total Cache Size} = 1 \text{ MB} = 1 \times 1024 \times 1024 \text{ bytes} = 1,048,576 \text{ bytes} = 2^{20} \text{ bytes}$$

Step 2: Identify the block size and convert it to a power of 2:

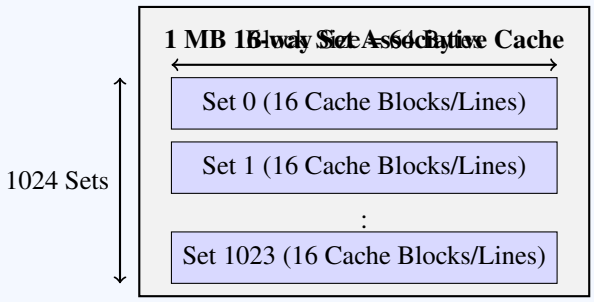
$$\text{Block Size} = 64 \text{ bytes} = 2^6 \text{ bytes}$$

Step 3: Identify the associativity (number of ways):

$$\text{Associativity} = 16\text{-way} = 2^4$$

Step 4: Calculate the number of sets in the cache:

$$\text{Number of Sets} = \frac{2^{20}}{2^6 \times 2^4} = \frac{2^{20}}{2^{10}} = 2^{10} = 1024 \text{ sets}$$



Final Answer: 1024

Answer: (B)

Go Back to Question 1

Q2.

Solution

Concept: The effective Cycles Per Instruction (CPI) of a pipelined processor is computed by adding the stall cycles introduced by hazards to the ideal CPI:

$$\text{CPI}_{\text{effective}} = \text{CPI}_{\text{ideal}} + (\text{Hazard Probability} \times \text{Stall Cycles per Hazard})$$

Solution: Step 1: Extract the parameters given in the problem:

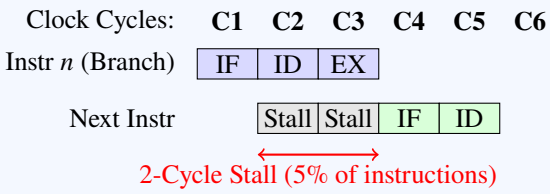
- Ideal CPI ($\text{CPI}_{\text{ideal}}$) = 1
- Branch hazard probability (fraction of instructions stalled) = 5% = 0.05
- Stall cycles per branch hazard = 2 cycles

Step 2: Calculate the branch stall overhead per instruction:

$$\text{Branch Stalls} = 0.05 \times 2 = 0.10 \text{ cycles per instruction}$$

Step 3: Sum the ideal CPI and the stall overhead:

$$\text{CPI}_{\text{effective}} = 1 + 0.10 = 1.10$$



Final Answer: 1.10

Answer: (B)

Go Back to Question 2

Q3.

Solution

Concept: The number of bits k required in an instruction field to uniquely identify one of N registers is governed by the inequality:

$$2^k \geq N$$

Solving for k yields:

$$k \geq \lceil \log_2(N) \rceil$$

Solution: Step 1: Identify the total number of general-purpose registers (N):

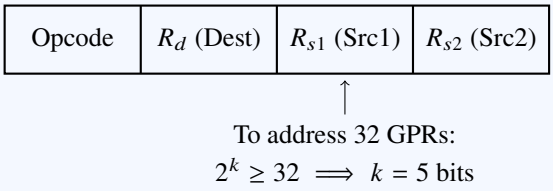
$$N = 32$$

Step 2: Substitute $N = 32$ into the logarithmic relation:

$$k = \log_2(32)$$

Step 3: Solve for the exact number of bits:

$$2^5 = 32 \implies k = 5 \text{ bits}$$



Final Answer: 5

Answer: (B)

Go Back to Question 3

Q4.

Solution

Concept: Memory-Mapped I/O (MMIO) and Isolated I/O are two distinct strategies for interfacing input/output devices with a processor:

- Isolated I/O: Uses a completely separate address space for I/O devices, requiring specialized hardware control signals and dedicated instruction set architecture (ISA) instructions (such as ‘IN‘ and ‘OUT‘ in x86).
- Memory-Mapped I/O: Maps both physical main memory and I/O registers to the same unified address space. This allows the processor to interact with peripheral registers using standard memory reference instructions (such as ordinary loads and stores).

Solution: Step 1: Assess Option A:
Memory-mapped I/O uses a shared (unified) address space, not a separate one. Thus, Option A describes Isolated I/O.

Step 2: Assess Option B:
Under memory-mapped I/O, peripheral device registers are assigned standard memory addresses. This means they can be read from and written to using standard load/store instructions (e.g., ‘LDR‘/‘STR‘ in ARM, or memory-referencing ‘MOV‘ in x86). This is the key consequence of MMIO.

Step 3: Assess Options C and D:
Cache memory remains essential for system performance, though I/O-mapped memory pages must be marked non-cacheable to ensure immediate visibility of external states. Direct Memory Access (DMA) can be, and is extensively, used with memory-mapped I/O.

Final Answer: I/O devices can be accessed using ordinary load/store instructions

Answer: (B)

Go Back to Question 4

Q5.

Solution

Concept: To convert a binary number directly into hexadecimal, group the binary digits into sets of four (nibbles) starting from the rightmost bit (least significant bit). Each 4-bit block is then mapped directly to its equivalent hexadecimal digit (0–9, A–F).

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

$$110101101011_2$$

Step 2: Group the bits into sets of four starting from the right:

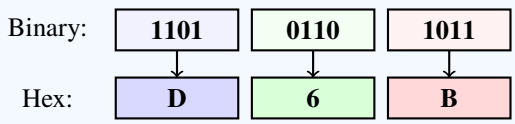
$$1101 \quad 0110 \quad 1011$$

Step 3: Convert each 4-bit binary group to its decimal equivalent:

- $1101_2 = 2^3 + 2^2 + 2^0 = 8 + 4 + 1 = 13$
- $0110_2 = 2^2 + 2^1 = 4 + 2 = 6$
- $1011_2 = 2^3 + 2^1 + 2^0 = 8 + 2 + 1 = 11$

Step 4: Convert each 4-bit group to hexadecimal:

- $1101_2 = D_{16}$
- $0110_2 = 6_{16}$
- $1011_2 = B_{16}$



Final Answer: D6B

Answer: (A)

Go Back to Question 5

Q6.

Solution

Concept: For an n-bit signed two's complement binary number, the most significant bit (MSB) acts as the sign bit.

- If the MSB is 0, the number is positive and can be converted directly to decimal.
- If the MSB is 1, the number is negative. Its magnitude is determined by inverting all the bits (obtaining the one's complement) and then adding 1 to the result.

Solution: Step 1: Write down the given signed 12-bit two's complement binary number:

$$111110101011_2$$

Because the MSB (leftmost bit) is 1, this is a negative value.
Step 2: Invert all bits (one's complement):

$$\text{Bitwise Inversion} = 000001010100_2$$

Step 3: Add 1 to find the positive magnitude in two's complement:

$$000001010100_2 + 1_2 = 000001010101_2$$

Step 4: Convert the positive binary magnitude to decimal:

$$\text{Decimal Value} = 2^6 + 2^4 + 2^2 + 2^0$$

$$\text{Decimal Value} = 64 + 16 + 4 + 1 = 85$$

Step 5: Apply the negative sign to represent the original signed value:

$$\text{Signed Value} = -85$$

MSB (Sign = -) Magnitude (11 bits)

1	11110101011

↑

Invert bits: 000001010100

Add 1: $000001010101_2 \implies 64 + 16 + 4 + 1 = 85$

Decimal Value = −85

Final Answer: -85

Answer: (A)

Go Back to Question 6

Q7.

Solution

Concept: An unsigned binary integer represented using k bits can express 2^k distinct numerical values ranging from 0 up to a maximum value of $2^k - 1$:

$$\text{Maximum Unsigned Value} = 2^k - 1$$

Solution: Step 1: Identify the size of the checksum field:

$$k = 8 \text{ bits}$$

Step 2: Calculate the total number of unique states representable:

$$\text{Total States} = 2^8 = 256 \text{ states}$$

Step 3: Determine the maximum unsigned decimal value:

$$\text{Maximum Value} = 2^8 - 1 = 256 - 1 = 255$$

Step 4: Analyze options: Option B represents the correct maximum value (255). Option A (127) corresponds to the maximum signed positive value in an 8-bit two's complement scheme ($2^7 - 1$). Options C (511) and D (1023) correspond to 9-bit and 10-bit maximum values respectively.

Final Answer: 255

Answer: (B)

Go Back to Question 7

Q8.

Solution

Concept: The uncompressed digital file size of an image is computed by multiplying the total number of pixels by the color depth (expressed in bytes per pixel):

$$\text{Total Pixels} = \text{Resolution Width} \times \text{Resolution Height}$$

$$\text{Image Size in Bytes} = \text{Total Pixels} \times \frac{\text{Color Depth in Bits}}{8}$$

To find the size in Megabytes, the byte total is converted using standard binary sizing ($1\ \text{MB} = 2^{20}$ bytes $= 1,048,576$ bytes).

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

$$\text{Total Pixels} = 1024 \times 768 = 786,432 \text{ pixels}$$

Step 2: Convert the color depth from bits to bytes:

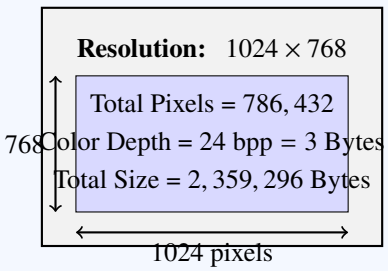
$$\text{Bytes per Pixel} = \frac{24 \text{ bits}}{8} = 3 \text{ bytes}$$

Step 3: Calculate the uncompressed image size in bytes:

$$\text{Size in Bytes} = 786,432 \text{ pixels} \times 3 \text{ bytes/pixel} = 2,359,296 \text{ bytes}$$

Step 4: Convert bytes to Megabytes (MB) using binary prefixes:

$$\text{Size in MB} = \frac{2,359,296 \text{ bytes}}{1,048,576 \text{ bytes/MB}} = 2.25 \text{ MB}$$



Final Answer: 2.25 MB

Answer: (B)

Go Back to Question 8

Q9.

Solution

Concept: The IEEE-754 floating-point standard optimizes bit usage for normalized binary numbers. Because a normalized binary number is always written in scientific notation with a leading integer value of 1 (e.g., $1.f \times 2^e$), this leading integer bit of 1 is constant. Consequently, it is not explicitly stored in physical memory, saving one bit.

Total Significant Bits of Precision = Explicitly Stored Fraction Bits + 1 (Implicit Hidden Bit)

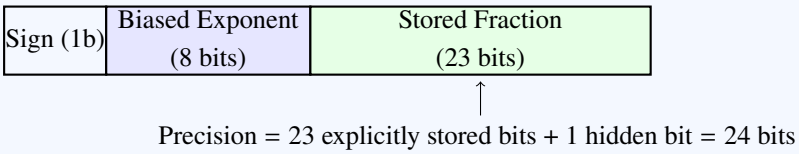
Solution: Step 1: Identify the specifications for IEEE-754 single precision:

- Total size = 32 bits
- Sign field = 1 bit
- Exponent field = 8 bits
- Explicit fraction (mantissa) field = 23 bits

Step 2: Add the implicit leading bit:

Total Precision Bits = 23 (stored) + 1 (hidden) = 24 bits

Step 3: Conclude: Although only 23 bits are physically allocated to the mantissa in hardware, normalized single-precision numbers achieve exactly 24 bits of binary precision.



Final Answer: 24

Answer: (C)

Go Back to Question 9

Q10.

Solution

Concept: In any digital character encoding system, the maximum number of unique characters that can be represented with n binary bits is equal to the number of unique mathematical states provided by those bits:

$$\text{Number of Unique Codes} = 2^n$$

Solution: Step 1: Identify the bit length used in the standard ASCII character set:

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

Step 2: Calculate the number of distinct representations:

$$\text{Number of Unique Character Codes} = 2^7 = 128 \text{ codes}$$

Step 3: Analyze the range: These codes are defined as integers from 0 up to 127.

- Codes 0–31 and 127 represent non-printable control characters.
- Codes 32–126 represent printable characters, including alphanumeric symbols and basic punctuation.

Option D ($256 = 2^8$) corresponds to Extended ASCII variations that use an entire 8-bit byte.

Final Answer: $\boxed{128}$

Answer: (C)

Go Back to Question 10

Q11.

Solution

Concept: In a system with a TLB:

- **TLB Hit:**

$$\text{Access Time}_{\text{hit}} = T_{\text{tlb}} + T_{\text{mem}}$$

- **TLB Miss:**

$$\text{Access Time}_{\text{miss}} = T_{\text{tlb}} + 2T_{\text{mem}}$$

If the TLB hit ratio is h, the Effective Memory Access Time (EMAT) is:

$$\text{EMAT} = h(T_{\text{tlb}} + T_{\text{mem}}) + (1 - h)(T_{\text{tlb}} + 2T_{\text{mem}})$$

Simplifying,

$$\boxed{\text{EMAT} = T_{\text{tlb}} + T_{\text{mem}} + (1 - h)T_{\text{mem}}}$$

Solution: Step 1: Identify the system parameters:

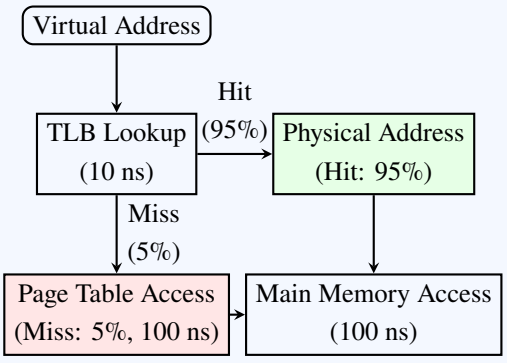
- TLB hit ratio (h) = 95% = 0.95
- TLB lookup time (T_{tlb}) = 10 ns
- Main memory access time (T_{mem}) = 100 ns

Step 2: Apply the values to the EMAT equation:

$$\text{EMAT} = 10\text{ ns} + 100\text{ ns} + (1 - 0.95) \cdot 100\text{ ns}$$

$$\text{EMAT} = 110\text{ ns} + 0.05 \cdot 100\text{ ns}$$

$$\text{EMAT} = 110\text{ ns} + 5\text{ ns} = 115\text{ ns}$$



Final Answer: 115 ns

Answer: (C)

Go Back to Question 11

Q12.

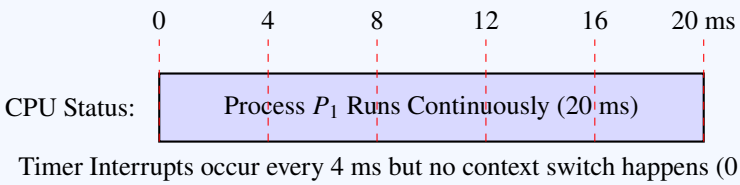
Solution

Concept: A context switch is the mechanism by which an operating system kernel suspends the execution of a currently active process on a CPU and resumes execution of a different process. In Round Robin (RR) scheduling, if a process's execution exceeds the time quantum, it is preempted. However, if there are no other active processes waiting in the ready queue, the scheduler has no other task to execute. Consequently, the CPU simply re-allocates the next time quantum to the same process, and no actual context switch to another process takes place.

Solution: Step 1: Analyze the process queue status: The process P_1 arrives at time $t = 0$ with a burst time of 20 ms. The ready queue contains only P_1.
Step 2: Simulate the scheduling cycles under time quantum $q = 4$ ms:

- $t = 0$ to 4 ms: P_1 executes its first quantum. At $t = 4$ ms, the timer interrupt is triggered. The scheduler checks the ready queue and finds it empty of other processes. P_1 is selected to continue executing immediately. No context switch occurs.
- $t = 4$ to 8 ms: P_1 executes its second quantum. No other processes arrive. At $t = 8$ ms, P_1 continues execution.
- $t = 8$ to 12 ms: P_1 executes its third quantum.
- $t = 12$ to 16 ms: P_1 executes its fourth quantum.
- $t = 16$ to 20 ms: P_1 executes its fifth quantum and terminates upon completion.

Step 3: Count the context switches: Because P_1 is the sole process in the system, the CPU never swaps execution state between different processes. Thus, 0 context switches occur.



Timer Interrupts occur every 4 ms but no context switch happens (0 switches)

Final Answer: 0

Answer: (D)

Go Back to Question 12

Q13.

Solution

Concept: RAID-1 is a mirroring storage technology. It duplicates the exact same data blocks across all the identical disks in the array to provide maximum fault tolerance. Because every disk in a pure RAID-1 array contains an identical copy of the data, the usable storage capacity is limited to the capacity of a single disk:

$$\text{Usable Capacity}_{\text{RAID-1}} = \text{Capacity of a single disk} = C$$

Regardless of whether there are 2, 3, or 4 disks in a standard RAID-1 group, the net storage capacity remains equal to one single drive's capacity.

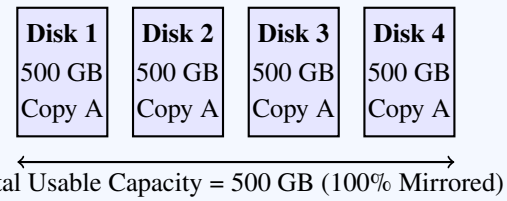
Solution: Step 1: Identify the disk parameters:

- Number of disks (N) = 4
- Capacity of each disk (C) = 500 GB

Step 2: Determine the usable capacity: In a pure RAID-1 configuration, all 4 disks are synchronized clones.

$$\text{Usable Capacity} = 500 \text{ GB}$$

Step 3: Analyze alternatives: If the 4 disks were instead configured in a striped-mirror setup like RAID-10 (RAID 1+0), they would form two mirrored pairs striped together, which would yield a capacity of 2×500 GB = 1 TB. However, for a standard pure RAID-1 mirrored array, the capacity is restricted to a single drive.



Final Answer: 500 GB

Answer: (A)

Go Back to Question 13

Q14.

Solution

Concept: Under the Cylinder-Head-Sector (CHS) physical addressing system, a hard disk's physical sectors are arranged in a multi-dimensional geometry. The total number of sectors present on the disk is the product of its physical components:

$$\text{Total Sectors} = \text{Cylinders} \times \text{Heads} \times \text{Sectors per Track}$$

Solution: Step 1: Extract the given CHS specifications:

- Number of Cylinders = $1024 = 2^{10}$
- Number of Heads = $64 = 2^6$
- Sectors per Track = $32 = 2^5$

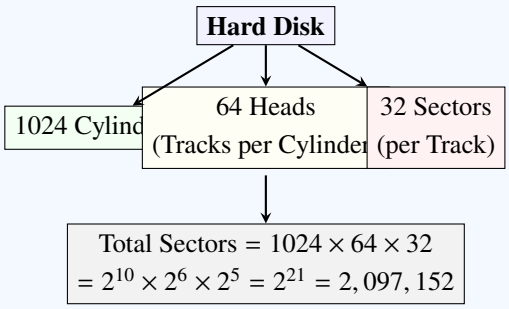
Step 2: Multiply the values to calculate total sectors:

$$\text{Total Sectors} = 1024 \times 64 \times 32$$

Using powers of 2 simplifies the calculation:

$$\text{Total Sectors} = 2^{10} \times 2^6 \times 2^5 = 2^{21}$$

$$\text{Total Sectors} = 2,097,152$$



Final Answer: $2,097,152$

Answer: (B)

Go Back to Question 14

Q15.

Solution

Concept: We can simplify a Product-of-Sums Boolean expression by expanding it algebraically using standard distributive and consensus laws:

- Distributive Law: $(W + X)(Y + Z) = WY + WZ + XY + XZ$
- Complement Law: $X \cdot \overline{X} = 0$

Solution: Step 1: Write down the Boolean expression:

$$\text{Expression} = (A + \overline{B})(\overline{A} + B)$$

Step 2: Expand the terms using the distributive law:

$$\text{Expression} = A\overline{A} + AB + \overline{B}\,\overline{A} + \overline{B}B$$

Step 3: Apply the complement laws ($A\overline{A} = 0$ and $\overline{B}B = 0$):

$$\text{Expression} = 0 + AB + \overline{A}\,\overline{B} + 0$$

$$\text{Expression} = AB + \overline{A}\,\overline{B}$$

Step 4: Analyze logic function equivalence: The resulting expression $AB + \overline{A}\,\overline{B}$ is the standard algebraic form of the Exclusive-NOR (XNOR) logical operator:

$$(A + \overline{B})(\overline{A} + B) = A \odot B = \overline{A \oplus B}$$

This is equivalent to Option A.

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

Final Answer: $\boxed{AB + \overline{A}\,\overline{B}}$

Answer: (A)

Go Back to Question 15

Q16.

Solution

Concept: For any Boolean function of n variables, there are 2^n possible unique input combinations in its domain. These input states are partitioned completely into combinations that produce a logical 1 (represented by minterms) and combinations that produce a logical 0 (represented by maxterms).

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

Solution: Step 1: Calculate the total possible input combinations for $n = 5$ variables:

$$\text{Total Combinations} = 2^5 = 32$$

Step 2: Identify the number of minterms (combinations where the output is 1):

$$\text{Minterms} = 10$$

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

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

$$\text{Maxterms} = 32 - 10 = 22$$

Step 4: Conclude: Because the canonical Product-of-Maxterms representation contains exactly one maxterm for each state yielding a logical 0, there are exactly 22 maxterms in this canonical form.

Total Input Combinations ($2^5 = 32$)

Output = 1 (10 Minterms)	Output = 0 (22 Maxterms)

Final Answer: 22

Answer: (B)

Go Back to Question 16

Q17.

Solution

Concept: The fundamental behavior of different multi-input logic gates can be defined by their output conditions:

- NAND: Produces 0 if and only if all inputs are 1.
- NOR: Produces 1 if and only if all inputs are 0.
- XOR (Exclusive OR): Produces 1 if and only if the total number of input variables equal to 1 is odd. This is why XOR is commonly utilized as an odd-parity detector.
- XNOR (Exclusive NOR): Produces 1 if and only if the total number of input variables equal to 1 is even.

Solution: Step 1: Evaluate the mathematical definition of the XOR operator for inputs $x_1, x_2, \ldots, x_k$:

$$\text{Output} = x_1 \oplus x_2 \oplus \cdots \oplus x_k$$

This expression evaluates to 1 when the sum $\sum_{i=1}^{k} x_i$ is an odd integer.

Step 2: Match with the options: Option C (XOR) is the correct gate.

Final Answer: XOR

Answer: (C)

Go Back to Question 17

Q18.

Solution

Concept: An IPv6 address is exactly 128 bits in length and is structurally divided into two main parts:

- Network Prefix: Identifies the specific network segment. Its length is defined by the CIDR notation prefix $/n$.
- Interface Identifier: Identifies a specific interface (host) on that subnet. The remaining bits are used for this identifier.

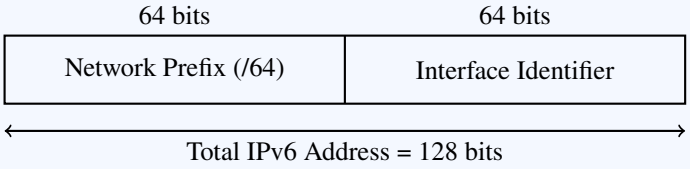
$$\text{Interface Identifier Bits} = 128 - \text{Prefix Bits}$$

Solution: Step 1: Identify the given IPv6 configuration parameters:

- Total IPv6 bits = 128
- Prefix length = /64

Step 2: Calculate the remaining bits available for interface identifiers:

$$\text{Interface Identifier Bits} = 128 - 64 = 64 \text{ bits}$$



Final Answer: 64

Answer: (C)

Go Back to Question 18

Q19.

Solution

Concept: The ACID properties define a set of four core requirements that guarantee database transactions are processed reliably:

- Atomicity: Ensures that a transaction is treated as a single unit, which either succeeds completely or fails completely ("all-or-nothing").
- Consistency: Guarantees that a transaction takes the database from one valid state to another, maintaining all schema constraints.
- Isolation: Ensures that the execution of concurrent transactions does not interfere with each other. No transaction should be able to view the intermediate, uncommitted states of other concurrent transactions.
- Durability: Guarantees that once a transaction commits, its modifications are permanently recorded in non-volatile storage, surviving subsequent system failures.

Solution: Step 1: Identify the scenario in the question: Multiple concurrent transactions are reading and writing shared items. We must prevent them from seeing or interfering with each other's uncommitted intermediate data states.

Step 2: Match with the appropriate property: The property that isolates running transactions from one another to ensure concurrency safety is Isolation (Option C).

Final Answer: Isolation

Answer: (C)

Go Back to Question 19

Q20.

Solution

Concept: Thrashing is a severe performance degradation phenomenon in operating systems that use virtual memory with demand paging.

- It occurs when the active working sets of all executing processes exceed the total available physical memory (RAM).
- As a result, the OS is forced to constantly swap pages out of RAM to disk and immediately swap other pages back in.
- Because disk access latency is orders of magnitude slower than physical memory access, the system spends almost all of its cycles handling page-fault interrupts rather than running application code.

Solution: Step 1: Analyze the characteristics of thrashing: During thrashing, the CPU utilization drops close to zero because the CPU is constantly idle while waiting for disk I/O operations to complete. The throughput of the system collapses because the system is spent servicing page faults instead of executing process instructions.

Step 2: Match with the options: Option B perfectly describes this state: "The system spends most of its time servicing page faults rather than executing processes".

Final Answer:

The system spends most of its time servicing page faults rather than executing processes

Answer: (B)

Go Back to Question 20

Answer Key

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