NIMCET Computer Awareness Sample Paper-3

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 36-bit address bus and is byte addressable.

Assuming every address refers to exactly one byte of memory, determine the maximum amount of memory that can be directly addressed by the processor.

(A) 32 GB

(B) 64 GB

(C) 128 GB

(D) 256 GB

Q2. Modern processors are often classified as either RISC or CISC architectures.

Which of the following features is generally associated with CISC processors rather than RISC processors?

(A) Fixed-length instructions

(B) Large register set

(C) Complex instruction decoding logic

(D) Load-store architecture

Q3. A processor implements a 5-stage instruction pipeline consisting of Fetch, Decode, Execute, Memory Access, and Write Back stages.

If all hazards and stalls are completely eliminated, then after the pipeline is filled, how many clock cycles are required to complete each additional instruction?

(A) 1 cycle

(B) 2 cycles

(C) 5 cycles

(D) 10 cycles

Q4. Consider the assembly language instruction:

$$MOV\ R1,\ (R2)$$

In this instruction, register $R2$ contains the address of the memory location where the operand is stored.

Which addressing mode is being used?

(A) Immediate Addressing

(B) Register Direct Addressing

(C) Register Indirect Addressing

(D) Indexed Addressing

Q5. An integer value of -45 is stored in a computer using an 8-bit two's complement representation.

Determine the corresponding binary representation of this number.

(A) 11010011

(B) 11010010

(C) 10110011

(D) 11100011

Q6. A hexadecimal number is given as

$$(3F6A)_{16}$$

Convert the above hexadecimal value into its equivalent decimal representation.

(A) 16234

(B) 16218

(C) 16106

(D) 16314

Q7. A binary encoding scheme is to be used for representing all decimal numbers from 0 to 999 inclusive.

What is the minimum number of binary bits required so that every such number can be uniquely represented?

(A) 9

(B) 10

(C) 11

(D) 12

Q8. The binary number

101101

is to be converted into its corresponding Gray code representation.

Which of the following options is correct?

(A) 111011

(B) 111101

(C) 111111

(D) 110111

Q9. The following 8-bit number is stored using the two's complement representation:

11101010

Determine its decimal equivalent.

(A) -22

(B) -20

(C) -18

(D) -26

Q10. Binary Coded Decimal (BCD) is commonly used in digital systems where decimal digits must be represented individually.

How many bits are required to represent a single decimal digit in standard BCD?

(A) 3 bits

(B) 4 bits

(C) 8 bits

(D) 16 bits

Q11. A computer system contains a cache memory with an access time of 10 ns.

The main memory access time is 100 ns, and the cache hit ratio is 90%.

What is the average memory access time for the system?

(A) 19 ns

(B) 20 ns

(C) 25 ns

(D) 30 ns

Q12. High-speed cache memories are designed to reduce the average memory access time of a processor.

Which of the following memory technologies is most commonly used for implementing cache memory?

(A) DRAM

(B) SRAM

(C) EEPROM

(D) Flash Memory

Q13. During the execution of a program, the processor repeatedly fetches instructions from memory.

Which of the following is stored in the Program Counter (PC) register at any given instant?

(A) The currently executing instruction

(B) Address of the next instruction to be fetched

(C) Address of the data operand

(D) Result produced by the ALU

Q14. Different memory devices exhibit different characteristics with respect to data retention.

Which of the following memory types loses its contents when power is removed?

(A) ROM

(B) Flash Memory

(C) SRAM

(D) EEPROM

Q15. Using the laws of Boolean Algebra, simplify the following expression:

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

The simplified result is:

(A) $A + B$

(B) A

(C) B

(D) AB

Q16. Consider the Boolean expression

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

By applying the appropriate Boolean identities, determine its simplest equivalent form.

(A) A

(B) B

(C) AB

(D) $A + B$

Q17. A Karnaugh Map is used for simplifying Boolean functions.

If a Boolean function contains five variables, how many cells will be present in its Karnaugh Map?

(A) 16

(B) 25

(C) 32

(D) 64

Q18. Network administrators often require secure access to remote machines over the Internet.

Which protocol provides encrypted communication and is widely used for secure remote login?

(A) FTP

(B) Telnet

(C) SSH

(D) SMTP

Q19. Operating systems can be categorized as proprietary or open source depending upon the availability of their source code.

Which of the following operating systems is open source?

(A) Windows 11

(B) macOS

(C) Linux

(D) MS-DOS

Q20. A process requests an input/output operation and cannot continue execution until the operation is completed.

During this period, the process is generally said to be in which state?

(A) Running

(B) Ready

(C) Blocked / Waiting

(D) Terminated

Detailed Solutions

Q1.

Solution

Concept: Memory size calculations for a byte-addressable system:

$$\text{Memory Size} = 2^N \times \text{addressable unit}$$

where N is the number of address lines (width of the address bus). For a byte-addressable system, each unique address refers to exactly 1 Byte (B). We use the standard power of two prefixes: 2^{10} B = 1 KB, 2^{20} B = 1 MB, and 2^{30} B = 1 GB.

Solution: Step 1: Identify the given parameters.

- Address bus size: $N = 36$ bits
- Addressing scheme: Byte-addressable (1 addressable unit = 1 Byte)

Step 2: Calculate the total number of unique addressable locations.

$$\text{Total Addresses} = 2^{36}$$

Step 3: Calculate the maximum addressable memory capacity.

$$\text{Memory Capacity} = 2^{36} \times 1 \text{ Byte} = 2^{36} \text{ B}$$

Step 4: Convert the capacity into gigabytes (GB).

$$2^{36} \text{ B} = 2^6 \times 2^{30} \text{ B} = 64 \times 1 \text{ GB} = 64 \text{ GB}$$

Thus, the processor can directly address a maximum of 64 GB of memory.

Final Answer: 64 GB

Answer: (B)

Go Back to Question 1

Q2.

Solution

Concept: Architectural differences between Complex Instruction Set Computer (CISC) and Reduced Instruction Set Computer (RISC) architectures:

- **RISC:** Focuses on reducing instruction complexity via software compilation. Key traits include fixed-length instructions, a large general-purpose register set, a simple instruction pipeline with fast and simple decoding logic, and a strict load-store architecture.
- **CISC:** Focuses on completing tasks in as few lines of assembly as possible by incorporating complex instructions directly into the hardware. Key traits include variable-length instructions, complex and multi-cycle execution, direct memory-to-memory operations, and a smaller register set.

Solution: Step 1: Evaluate each option against RISC and CISC principles.

- **Option A (Fixed-length instructions):** This is a standard characteristic of RISC. Having a set size (e.g., 32 bits) simplifies decoding and alignment in pipelining.
- **Option B (Large register set):** Typically associated with RISC, where multiple registers minimize the need to access slower external memory.
- **Option C (Complex instruction decoding logic):** Because CISC instructions have variable lengths and diverse formatting (varying source/destination structures), the hardware requires complex decoding circuits to translate instructions into control signals. This is a characteristic feature of CISC.
- **Option D (Load-store architecture):** A hallmark of RISC where memory is accessed only via explicit Load/Store instructions, and other operations occur solely within registers. CISC allows operations directly on memory operands.

Step 2: Conclude that complex instruction decoding logic is associated with CISC processors rather than RISC.

Final Answer: Complex instruction decoding logic

Answer: (C)

Go Back to Question 2

Q3.

Solution

Concept: An instruction pipeline overlaps the execution of multiple instructions. For an ideal k-stage pipeline, the time required to complete the very first instruction is k clock cycles. Once the pipeline is fully filled, and assuming there are no pipeline hazards (data, structural, or control) and no stalls, one instruction completes on every single subsequent clock cycle. The Cycles Per Instruction (CPI) under ideal operating conditions is 1.

Solution: Step 1: Identify the pipeline stages. The processor has $k = 5$ stages: Fetch (IF), Decode (ID), Execute (EX), Memory Access (MEM), and Write Back (WB).

Step 2: Analyze the clock cycles required.

- The first instruction requires 5 clock cycles to travel through all stages and complete (at which point the pipeline is filled).
- On cycle 6, the second instruction finishes its Write Back (WB) stage and completes.
- On cycle 7, the third instruction completes, and so on.

Step 3: Visualize the pipeline execution using a cycle-by-cycle grid.

Instruction	CC 1	CC 2	CC 3	CC 4	CC 5	CC 6	CC 7
I_1	IF	ID	EX	MEM	WB		
I_2		IF	ID	EX	MEM	WB	
I_3			IF	ID	EX	MEM	WB

As shown above, after the pipeline is filled (from CC 5 onwards), each additional instruction requires exactly 1 clock cycle to finish execution.

Final Answer: 1 cycle

Answer: (A)

Go Back to Question 3

Q4.

Solution

Concept: Addressing modes describe how the effective address of an instruction's operand is calculated:

- **Immediate Addressing:** The operand is an explicit constant in the instruction (e.g., `MOV R1, #10`).
- **Register Direct Addressing:** The operand is stored in a register (e.g., `MOV R1, R2`).
- **Register Indirect Addressing:** The register stores a memory pointer/address where the operand resides (e.g., `MOV R1, (R2)`).
- **Indexed Addressing:** The effective address is the sum of a register value and an offset (e.g., `MOV R1, 5(R2)`).

Solution: Step 1: Parse the instruction structure:

$$\text{MOV } R1, (R2)$$

Step 2: Determine how the second operand is accessed. The syntax $(R2)$ or $[R2]$ denotes that the register $R2$ contains the physical memory address of the target data, not the operand itself. The processor must first read $R2$ to get the address, and then read the memory contents at that address.

Step 3: Match with the appropriate addressing mode. This mechanism is the definition of **Register Indirect Addressing**.

Final Answer: Register Indirect Addressing

Answer: (C)

Go Back to Question 4

Q5.

Solution

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

(a) Represent the positive magnitude $+X$ as an 8-bit unsigned binary number.

(b) Find the one's complement by inverting all the bits ($0 \rightarrow 1$ and $1 \rightarrow 0$).

(c) Add 1 to the one's complement result to obtain the two's complement representation.

Solution: Step 1: Write down +45 in 8-bit binary.

$$45 = 32 + 8 + 4 + 1 = 2^5 + 2^3 + 2^2 + 2^0$$

$$\text{Binary of } (+45) = 00101101_2$$

Step 2: Convert to one's complement by inverting the bits.

$$\text{One's Complement of } 00101101_2 = 11010010_2$$

Step 3: Obtain the two's complement by adding 1.

$$11010010_2 + 1 = 11010011_2$$

Step 4: Verify the signed representation.

$$\text{Decimal} = (-128 \times 1) + (64 \times 1) + (32 \times 0) + (16 \times 1) + (8 \times 0) + (4 \times 0) + (2 \times 1) + (1 \times 1)$$

$$\text{Decimal} = -128 + 64 + 16 + 2 + 1 = -45$$

This confirms that 11010011 is the correct 8-bit binary representation.

Final Answer: $\boxed{11010011}$

Answer: (A)

Go Back to Question 5

Q6.

Solution

Concept: To convert a hexadecimal value $(h_n h_{n-1} \ldots h_0)_{16}$ into its decimal equivalent, we sum each hexadecimal digit multiplied by its positional weight (powers of 16):

$$\text{Decimal Value} = \sum_{i=0}^{n} (h_i \times 16^i)$$

Where the letters represent decimal equivalents as:

$$\text{A} = 10, \quad \text{B} = 11, \quad \text{C} = 12, \quad \text{D} = 13, \quad \text{E} = 14, \quad \text{F} = 15$$

Solution: Step 1: Expand the hexadecimal number $(3F6A)_{16}$ into positional weights.

$$\text{Decimal Value} = 3 \times 16^3 + \text{F} \times 16^2 + 6 \times 16^1 + \text{A} \times 16^0$$

Step 2: Substitute the decimal values for the non-numeric symbols (F = 15, A = 10).

$$\text{Decimal Value} = 3 \times 16^3 + 15 \times 16^2 + 6 \times 16^1 + 10 \times 16^0$$

Step 3: Compute the powers of 16.

$$16^0 = 1$$
$$16^1 = 16$$
$$16^2 = 256$$
$$16^3 = 4096$$

Step 4: Multiply each digit by its positional weight.

$$3 \times 4096 = 12288$$
$$15 \times 256 = 3840$$
$$6 \times 16 = 96$$
$$10 \times 1 = 10$$

Step 5: Sum up all the terms.

$$\text{Decimal Value} = 12288 + 3840 + 96 + 10 = 16234$$

Therefore, $(3F6A)_{16}$ converts to the decimal value 16234.

Final Answer: 16234

Answer: (A)

Go Back to Question 6

Q7.

Solution

Concept: The number of distinct decimal values that can be uniquely encoded using N binary bits is 2^N. To find the minimum number of bits required to represent M distinct numbers, we select the smallest integer N such that:

$$2^N \geq M$$

Taking the ceiling of log base 2:

$$N = \lceil \log_2(M) \rceil$$

Solution: Step 1: Determine the total quantity of numbers, M, in the range $[0, 999]$. Since the range is inclusive of both 0 and 999:

$$M = 999 - 0 + 1 = 1000 \text{ values}$$

Step 2: Set up the inequality to find the minimum N bits.

$$2^N \geq 1000$$

Step 3: Analyze powers of 2.

$$2^9 = 512 \quad \text{(insufficient, } 512 < 1000\text{)}$$
$$2^{10} = 1024 \quad \text{(sufficient, } 1024 \geq 1000\text{)}$$

Step 4: Conclude the minimum bit requirements. A minimum of 10 bits is needed to uniquely encode 1000 distinct values. This matches Option B.

Final Answer: $\boxed{10}$

Answer: (B)

Go Back to Question 7

Q8.

Solution

Concept: Converting a binary number $B = b_{n-1}b_{n-2}\ldots b_1b_0$ into its equivalent Gray code $G = g_{n-1}g_{n-2}\ldots g_1g_0$ uses the following logical mapping:

$$g_{n-1} = b_{n-1} \quad \text{(the Most Significant Bit remains unchanged)}$$
$$g_i = b_{i+1} \oplus b_i \quad \text{for } 0 \le i < n-1$$

where $\oplus$ represents the bitwise exclusive OR (XOR) operation.

Solution: Step 1: Map the individual bits of the given binary number $B = 101101_2$ (6-bit length).

$$b_5 = 1, \quad b_4 = 0, \ldots, b_0 = 1$$

Step 2: Calculate each Gray code bit using the XOR relation.

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

Step 3: Assemble the Gray code bitstream.

$$G = 111011$$

This corresponds directly to Option A.

Final Answer: $\boxed{111011}$

Answer: (A)

Go Back to Question 8

Q9.

Solution

Concept: An N-bit signed integer represented in two's complement can be evaluated in two ways:

- **Weighted Sum:** Sum the bits multiplied by their respective powers, where the MSB has a negative weight:

$$\text{Value} = -b_{N-1}2^{N-1} + \sum_{i=0}^{N-2} b_i 2^i$$

- **Magnitude Inversion:** Since the MSB is 1 (indicating a negative value), we can find its positive magnitude by taking the two's complement of the binary number, converting that magnitude to decimal, and adding a negative sign.

Solution: Step 1: Check the MSB of the 8-bit binary pattern $B = 11101010_2$. The MSB is 1, so the decimal value is negative.

Step 2: Find the magnitude by performing a two's complement step.

- Invert the bits (one's complement): $11101010_2 \rightarrow 00010101_2$
- Add 1 to get the magnitude: $00010101_2 + 1 = 00010110_2$

Step 3: Convert the positive binary magnitude to decimal.

$$00010110_2 = (1 \times 16) + (1 \times 4) + (1 \times 2) = 22$$

Step 4: Append the negative sign to get the final signed value.

$$\text{Decimal Value} = -22$$

Step 5: Verify using the weighted sum method.

$$\text{Value} = -128 \cdot (1) + 64 \cdot (1) + 32 \cdot (1) + 16 \cdot (0) + 8 \cdot (1) + 4 \cdot (0) + 2 \cdot (1) + 1 \cdot (0)$$

$$\text{Value} = -128 + 64 + 32 + 8 + 2 = -128 + 106 = -22$$

Both methods yield the same result, which is Option A.

Final Answer: -22

Answer: (A)

Go Back to Question 9

Q10.

Solution

Concept: Binary Coded Decimal (BCD) is a digital encoding method where each decimal digit is written as a discrete binary pattern. To choose how many bits are required, we count the number of states needed to uniquely represent all 10 decimal digits $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$. Using N bits, we can represent 2^N unique states. Thus, we select the smallest integer N satisfying:

$$2^N \geq 10$$

Solution: Step 1: Analyze the binary states possible for different bit widths.

- $N = 3 \implies 2^3 = 8$ unique states (not enough to cover all 10 decimal digits)
- $N = 4 \implies 2^4 = 16$ unique states (enough to represent 10 decimal digits, leaving 6 states unused)

Step 2: Conclude the bit requirement. Standard BCD (specifically the 8421 BCD encoding) utilizes a fixed length of exactly 4 bits per decimal digit (from `0000` for 0 to `1001` for 9).
This matches Option B.

Final Answer: 4 bits

Answer: (B)

Go Back to Question 10

Q11.

Solution

Concept: The average memory access time (AMAT) of a cache-memory system depends on how cache and main memory are accessed:

(a) **Simultaneous (Parallel) Access:** Cache and main memory are accessed together.

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

(b) **Hierarchical (Serial) Access:** The cache is checked first; on a miss, main memory is accessed.

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

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

Solution: Step 1: Identify the given parameters.

- Cache access time (T_{cache}) = 10 ns
- Main memory access time (T_{main}) = 100 ns
- Hit ratio (H) = 90% = 0.90
- Miss ratio ($1 - H$) = 10% = 0.10

Step 2: Calculate the average access time under the simultaneous access model.

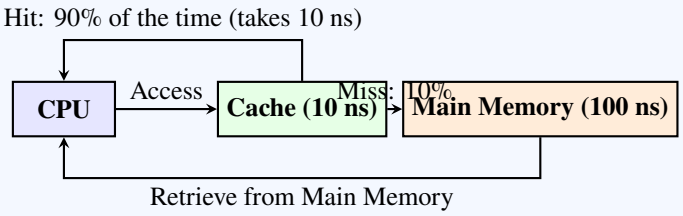
$$\text{AMAT} = (0.90 \times 10 \text{ ns}) + (0.10 \times 100 \text{ ns})$$

$$\text{AMAT} = 9 \text{ ns} + 10 \text{ ns} = 19 \text{ ns}$$

Step 3: Calculate the average access time under the hierarchical access model.

$$\text{AMAT} = 10 \text{ ns} + (0.10 \times 100 \text{ ns})$$

$$\text{AMAT} = 10 \text{ ns} + 10 \text{ ns} = 20 \text{ ns}$$



Final Answer: 19 ns

Answer: (A)

Go Back to Question 11

Q12.

Solution

Concept: Memory technologies are selected based on tradeoffs between speed, cost, and density:

- **SRAM (Static Random-Access Memory):** Extremely fast because it uses flip-flops (typically 6 transistors per cell) to store each bit. It does not require periodic refreshing. However, it is expensive and has low density, making it ideal for CPU registers and cache memory.
- **DRAM (Dynamic Random-Access Memory):** Uses a single transistor and capacitor per cell. It is slower than SRAM and requires constant refreshing to maintain charge, but it is cheap and highly dense, making it ideal for main memory.
- **EEPROM and Flash Memory:** Non-volatile technologies used for firmware storage and mass storage respectively, which are far too slow for cache-speed operations.

Solution: Step 1: Relate cache requirements to memory characteristics. Cache memory requires the highest possible operational speeds to keep up with the CPU clock cycles.

Step 2: Identify the technology that provides the fastest access times without the overhead of refreshing cycles. SRAM is the primary technology used to build L1, L2, and L3 caches.

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

Final Answer: SRAM

Answer: (B)

Go Back to Question 12

Q13.

Solution

Concept: The execution of an instruction involves the Fetch-Decode-Execute cycle. Dedicated registers track this workflow:

- **Program Counter (PC):** Automatically holds the memory address of the next instruction scheduled for fetching. Once a fetch completes, the PC is automatically incremented to point to the succeeding instruction (unless a branch occurs).
- **Instruction Register (IR):** Stores the binary instructions currently being decoded and executed.
- **Memory Address Register (MAR):** Holds physical addresses of operands/data.
- **Accumulator (AC) / General Registers:** Hold immediate ALU computational outputs.

Solution: Step 1: Identify the role of the Program Counter (PC). By definition, the PC serves as a pointer in memory to keep track of the instruction stream execution sequence.

Step 2: Map the PC's contents at any given moment. At any instant, it holds the memory address of the next instruction to be retrieved.

Step 3: Match with the choices. Option B represents this definition.

Final Answer: Address of the next instruction to be fetched

Answer: (B)

Go Back to Question 13

Q14.

Solution

Concept: Data retention characteristics classify memory into two categories:

- **Volatile Memory:** Requires electrical power to maintain its stored state. If the power supply is interrupted, the saved information is lost almost instantly (e.g., SRAM, DRAM).
- **Non-Volatile Memory:** Retains its saved state even when completely powered off (e.g., ROM, Flash, EEPROM).

Solution: Step 1: Evaluate each option's volatility.

- **ROM (Read-Only Memory):** Non-volatile; used to store bootstrap code.
- **Flash Memory:** Non-volatile; used in solid-state and thumb drives.
- **SRAM (Static RAM):** Volatile; loses its data when power is disconnected.
- **EEPROM (Electrically Erasable Programmable ROM):** Non-volatile; retains parameters across restarts.

Step 2: Identify the memory type that loses its contents. SRAM is volatile.

Final Answer: SRAM

Answer: (C)

Go Back to Question 14

Q15.

Solution

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

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

Solution: Step 1: Write down the expression to simplify.

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

Step 2: Apply the distributive law of addition over multiplication. Here, treating A as the common term:

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

Step 3: Simplify the complementary product. Since a variable and its inverse cannot both be true simultaneously:

$$B \cdot \overline{B} = 0$$

Step 4: Substitute the value back into the expression.

$$A + 0 = A$$

Thus, the simplified expression is A, which corresponds to Option B.

Final Answer: $\boxed{A}$

Answer: (B)

Go Back to Question 15

Q16.

Solution

Concept: Boolean simplification relies on factoring common variables and applying identity laws:

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

Solution: Step 1: Analyze the given Boolean expression.

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

Step 2: Factor out the common term A using the distributive law.

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

Step 3: Simplify the term inside the parentheses. Since either a variable or its complement must always be true:

$$B + \overline{B} = 1$$

Step 4: Apply the multiplicative identity law.

$$A(1) = A$$

The simplest equivalent form is A, which corresponds to Option A.

Final Answer: $\boxed{A}$

Answer: (A)

Go Back to Question 16

Q17.

Solution

Concept: A Karnaugh Map (K-map) is a graphical representation used to simplify Boolean expressions. Every cell in a K-map corresponds to a unique minterm (combination of input variables). For a Boolean function containing n variables, the total number of cells in the K-map is:

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

Solution: Step 1: Identify the number of variables n in the given function.

$$n = 5$$

Step 2: Calculate the number of cells using the formula.

$$\text{Number of Cells} = 2^5 = 32$$

Step 3: Match the result with the options. A 5-variable K-map contains exactly 32 cells (often visualized as two adjacent 4-variable K-maps of 16 cells each). This corresponds to Option C.

Final Answer: 32

Answer: (C)

Go Back to Question 17

Q18.

Solution

Concept: Network protocols operate under different security paradigms:

- **FTP (File Transfer Protocol):** Used for transferring files; transmits credentials and data in plaintext.
- **Telnet:** Used for remote command-line access; transmits all communication (including passwords) in unencrypted plaintext, making it highly insecure.
- **SSH (Secure Shell):** A cryptographic network protocol that establishes an encrypted tunnel for secure remote command-line execution and data transmission.
- **SMTP (Simple Mail Transfer Protocol):** Used specifically for sending electronic mail.

Solution: Step 1: Identify the main requirement. The protocol must provide encrypted communication specifically for secure remote login.

Step 2: Analyze the protocols. Telnet is the insecure remote login option, whereas SSH was designed directly as a secure replacement for Telnet.

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

Final Answer: SSH

Answer: (C)

Go Back to Question 18

Q19.

Solution

Concept: Operating systems are categorized based on their licensing and development models:

- **Proprietary (Closed Source):** The source code is owned by a specific organization and is not publicly accessible or alterable by users (e.g., Windows, macOS, MS-DOS).
- **Open Source:** The source code is publicly available, allowing anyone to view, modify, and distribute it freely under open-source licenses (e.g., Linux).

Solution: Step 1: Examine the classification of each listed operating system.

- **Windows 11:** Proprietary operating system owned by Microsoft.
- **macOS:** Proprietary operating system owned by Apple.
- **Linux:** Open-source operating system kernel developed collaboratively by a global community.
- **MS-DOS:** Historically proprietary operating system owned by Microsoft.

Step 2: Identify the open-source system. Linux is open source, corresponding to Option C.

Final Answer: Linux

Answer: (C)

Go Back to Question 19

Q20.

Solution

Concept: During its lifecycle, a process transitions through various states in an operating system:

- **Running:** The process's instructions are currently executing on the CPU.
- **Ready:** The process is prepared to execute and is waiting to be allocated CPU time by the scheduler.
- **Blocked / Waiting:** The process cannot execute because it is waiting for an external event, such as a user keystroke, disk I/O, or network packet.
- **Terminated:** The process has finished executing.

Solution: Step 1: Identify the trigger event. The process makes an active request for an input/output (I/O) operation.

Step 2: Understand the scheduler's behavior. Because I/O operations are extremely slow compared to CPU cycles, the CPU scheduler preempts the process from the **Running** state and puts it aside so another process can use the CPU.

Step 3: Define the state. The process remains in the **Blocked / Waiting** state until the hardware signals that the I/O operation is complete. Once completed, the process transitions back to the **Ready** state.

Step 4: Match with the options. This state is Blocked / Waiting, corresponding to Option C.

Final Answer: Blocked / Waiting

Answer: (C)

Go Back to Question 20

Answer Key

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