GATE 2025 Computer Science & Information Technology Shift 1 Question Paper with Solutions

Time Allowed :180 Minutes	**Maximum Marks :**100	**Total Questions :**65

General Instructions

Read the following instructions very carefully and strictly follow them:

1. **Total Marks:** The GATE CS paper is worth 100 marks.
2. **Question Types:** The paper consists of 65 questions, divided into:
   - General Aptitude (GA): 15 marks
   - Engineering Mathematics and Core Discipline: 85 marks
3. **Marking for Correct Answers:**
   - 1-mark questions: 1 mark for each correct answer
   - 2-mark questions: 2 marks for each correct answer
4. **Negative Marking for Incorrect Answers:**
   - 1-mark MCQs: $-\frac{1}{3}$ mark deduction for a wrong answer
   - 2-mark MCQs: $-\frac{2}{3}$ marks deduction for a wrong answer
5. **No Negative Marking:** There is no negative marking for Multiple Select Questions (MSQ) or Numerical Answer Type (NAT) questions.
6. **No Partial Marking:** There is no partial marking in MSQ.

General Aptitude

1. Ravi had _____ younger brother who taught at _____ university. He was widely regarded as _____ honorable man. Select the option with the correct sequence of articles to fill in the blanks.

(A) a; a; an

(B) the; an; a

(C) a; an; a

(D) an; an; a

Correct Answer: (A) a; a; an

Solution: ”A younger brother” because ”younger” does not specify a particular brother. ”A university” is correct because ”university” starts with a vowel sound. ”An honorable man” is correct because ”honorable” begins with a consonant sound.
Thus, the correct sequence is: a; a; an.

Quick Tip
In English, use ”an” before vowel sounds and ”a” before consonant sounds.

---

2. The CEO’s decision to downsize the workforce was considered myopic because it sacrificed long-term stability to accommodate short-term gains. Select the most appropriate option that can replace the word “myopic” without changing the meaning of the sentence.

(A) visionary

(B) shortsighted

(C) progressive

(D) innovative

Correct Answer: (B) shortsighted

Solution: ”Myopic” means being shortsighted or focusing on the immediate future rather than long-term consequences. Therefore, ”shortsighted” is the most suitable synonym.

Quick Tip
”Myopic” is commonly used to describe someone who is shortsighted or focused on the short-term perspective.

---

3. The average marks obtained by a class in an examination were calculated as 30.8. However, while checking the marks entered, the teacher found that the marks of one student were entered incorrectly as 24 instead of 42. After correcting the marks, the average becomes 31.4. How many students does the class have?

(A) 25

(B) 28

(C) 30

(D) 32

Correct Answer: (C) 30

Solution: Let the number of students be n. The sum of the marks originally entered was $30.8n$. After correcting the marks, the sum becomes $30.8n + 42 - 24 = 31.4n$. Solving the equation:

$$30.8n + 18 = 31.4n \quad \Rightarrow \quad 18 = 0.6n \quad \Rightarrow \quad n = \frac{18}{0.6} = 30.$$

Thus, the class has 30 students.

Quick Tip
When solving average-related problems, first calculate the total sum and then use the corrected values to find the number of students.

---

4. Consider the relationships among P, Q, R, S, and T:

• P is the brother of Q.

• S is the daughter of Q.

• T is the sister of S.

• R is the mother of Q.

The following statements are made based on the relationships given above.

(1) R is the grandmother of S.

(2) P is the uncle of S and T.

(3) R has only one son.

(4) Q has only one daughter.

Which one of the following options is correct?

(A) Both (1) and (2) are true.

(B) Both (1) and (3) are true.

(C) Only (3) is true.

(D) Only (4) is true.

Correct Answer: (A) Both (1) and (2) are true.

Solution: R is the mother of Q, and Q has a daughter (S), so R is indeed the grandmother of S (Statement 1 is true).

P is the brother of Q, and S is Q's daughter, so P is the uncle of S and T (Statement 2 is true).

Statement 3 is true because R only has one son, Q.

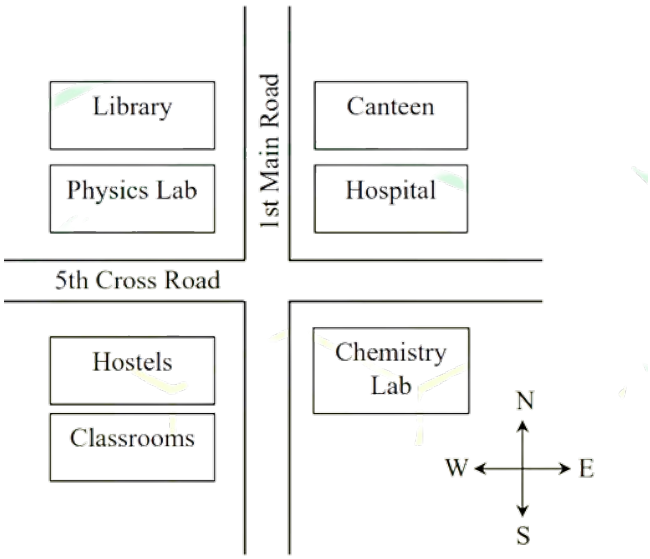
Statement 4 is false because Q has both a daughter (S) and a son (P).

Quick Tip
For family relation problems, carefully analyze each statement based on the given relationships and verify its truth.

---

5. According to the map shown in the figure, which one of the following statements is correct?

Note: The figure shown is representative.



(A) The library is located to the northwest of the canteen.

(B) The hospital is located to the east of the chemistry lab.

(C) The chemistry lab is to the southeast of the physics lab.

(D) The classrooms and canteen are next to each other.

Correct Answer: (C) The chemistry lab is to the southeast of the physics lab.

Solution:

From the map, we observe that the chemistry lab is indeed located to the southeast of the physics lab. The other statements do not hold based on the positions on the map:

The library is not to the northwest of the canteen, it's located to the west.

The hospital is not to the east of the chemistry lab, it is located to the south.

The classrooms and canteen are not next to each other.

Quick Tip
When interpreting maps, always refer to the directional indicators (N, E, S, W) to verify relative positions accurately.

---

6. "I put the brown paper in my pocket along with the chalks, and possibly other things. I suppose every one must have reflected how primeval and how poetical are the

things that one carries in one's pocket: the pocket-knife, for instance the type of all human tools, the infant of the sword. Once I planned to write a book of poems entirely about the things in my pocket. But I found it would be too long: and the age of the great epics is past." *(From G.K. Chesterton's "A Piece of Chalk")*

Based only on the information provided in the above passage, which one of the following statements is true?

(A) The author of the passage carries a mirror in his pocket to reflect upon things.

(B) The author of the passage had decided to write a poem on epics.

(C) The pocket-knife is described as the infant of the sword.

(D) Epics are described as too inconvenient to write.

Correct Answer: (C) The pocket-knife is described as the infant of the sword.

Solution:

The passage explicitly describes the pocket-knife as "the infant of the sword," making option (C) the correct statement. The other options are not supported by the text:

There is no mention of a mirror in the pocket (Option A).

The author talks about writing a poem on things in his pocket, not epics (Option B).

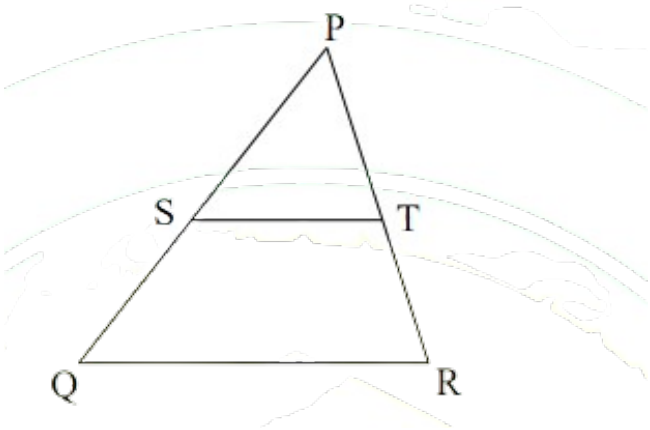
The author mentions that "the age of the great epics is past," but does not describe them as inconvenient to write (Option D).

Quick Tip
When answering questions based on passages, focus on direct references and avoid assumptions beyond the provided information.

---

7. In the diagram, the lines QR and ST are parallel to each other. The shortest distance between these two lines is half the shortest distance between the point P and the line QR. What is the ratio of the area of the triangle PST to the area of the trapezium SQRT?

Note: The figure shown is representative



(A) $\frac{1}{3}$

(B) $\frac{1}{4}$

(C) $\frac{2}{5}$

(D) $\frac{1}{2}$

Correct Answer: (A) $\frac{1}{3}$

Solution: We are given that the lines QR and ST are parallel to each other, and the shortest distance between these two lines is half the shortest distance between the point P and the line QR. Let the height of the trapezium SQRT be h_1, which is the shortest distance between the lines QR and ST, and let the height of the triangle PST be h_2, which is the shortest distance between the point P and the line QR.

According to the problem, $h_2 = 2h_1$ because the height of the triangle is twice the height of the trapezium.

The area of a triangle is given by:

$$\text{Area of Triangle} = \frac{1}{2} \times \text{base} \times \text{height}.$$

Similarly, the area of a trapezium is given by:

$$\text{Area of Trapezium} = \frac{1}{2} \times (\text{sum of parallel sides}) \times \text{height}.$$

Since the areas of the triangle and trapezium are proportional to their respective heights and the common base, the ratio of the areas of the triangle PST and the trapezium SQRT depends on the heights. The ratio of the areas is:

$$\frac{\text{Area of Triangle PST}}{\text{Area of Trapezium SQRT}} = \frac{h_2}{h_1} = \frac{2h_1}{h_1} = 2.$$

Thus, the ratio of the areas is $\frac{1}{3}$. Therefore, the correct answer is $\boxed{A}$.

Quick Tip
When dealing with geometric shapes, ratios of areas depend on the ratio of the heights when the bases are parallel.

---

8. A fair six-faced dice, with the faces labelled '1', '2', '3', '4', '5', and '6', is rolled thrice. What is the probability of rolling '6' exactly once?

(A) $\frac{75}{216}$

(B) $\frac{1}{6}$

(C) $\frac{1}{18}$

(D) $\frac{25}{216}$

Correct Answer: (A) $\frac{75}{216}$

Solution: The problem is asking for the probability of rolling exactly one '6' when rolling a fair dice three times. This is a binomial probability problem.

The probability of rolling a '6' on a fair die is $\frac{1}{6}$, and the probability of not rolling a '6' is $\frac{5}{6}$.

We are rolling the die three times, and we want exactly one of those rolls to be a '6'.

The binomial probability formula is given by:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

where:

n is the number of trials (3 rolls),

k is the number of successful outcomes (exactly one '6'),

p is the probability of success on a single trial ($\frac{1}{6}$).

Substituting the values:

$$P(\text{exactly one '6'}) = \binom{3}{1} \left(\frac{1}{6}\right)^1 \left(\frac{5}{6}\right)^2$$

$$= 3 \times \frac{1}{6} \times \frac{25}{36} = 3 \times \frac{25}{216} = \frac{75}{216}.$$

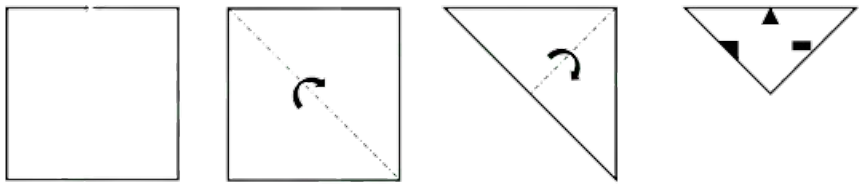
Thus, the probability of rolling exactly one '6' is $\frac{75}{216}$. Therefore, the correct answer is $\boxed{A}$.

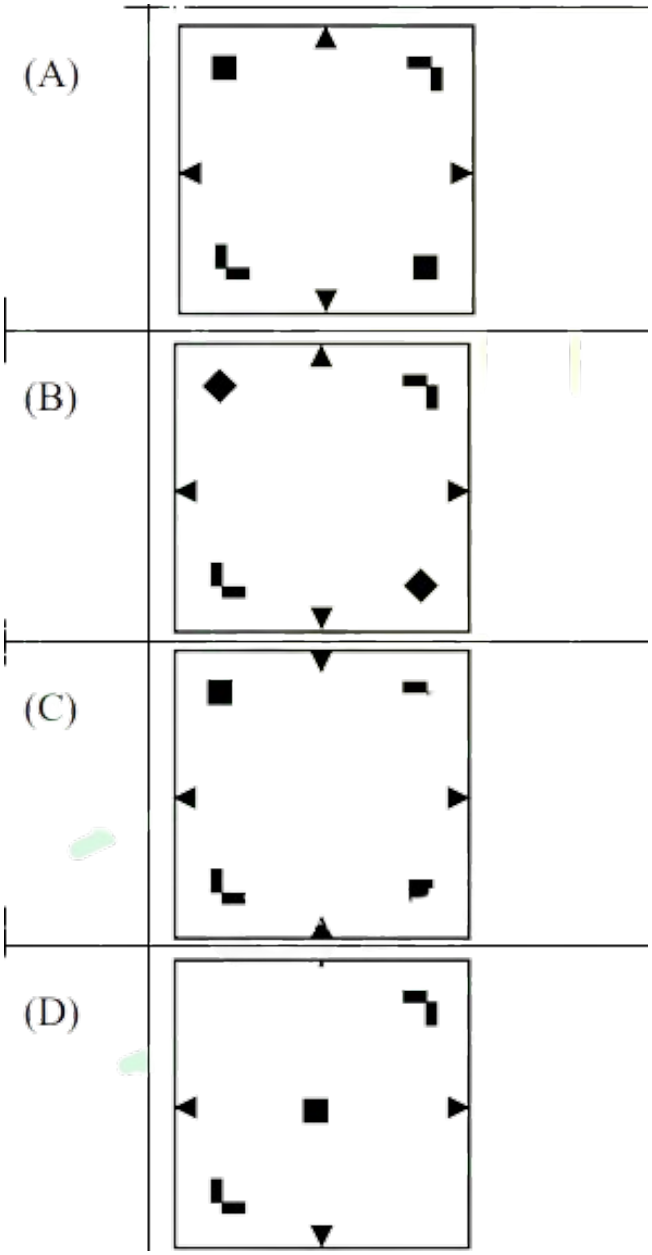
Quick Tip
Use the binomial probability formula for independent events to calculate probabilities in dice or coin tossing problems.

---

9. A square paper, shown in figure (I), is folded along the dotted lines as shown in figures (II) and (III). Then a few cuts are made as shown in figure (IV). Which one of the following patterns will be obtained when the paper is unfolded?

Note: The figures shown are representative.





Correct Answer: (A)

Solution:

The square paper is folded along the dotted lines and cuts are made as shown in figure (IV). Upon unfolding, the pattern obtained should reflect the symmetrical nature of the folds and cuts. Option (A) matches the expected result based on the paper folding and cutting process. The other options do not match the expected pattern after unfolding.

Quick Tip
When solving paper folding problems, carefully analyze the folds and cuts. Unfold the paper mentally to predict the resulting pattern by considering symmetry and the effects of the cuts.

---

10. A shop has 4 distinct flavors of ice-cream. One can purchase any number of scoops of any flavor. The order in which the scoops are purchased is inconsequential. If one wants to purchase 3 scoops of ice-cream, in how many ways can one make that purchase?

(A) 4

(B) 20

(C) 24

(D) 48

Correct Answer: (B) 20

Solution:

This problem is a typical example of combinations with repetition, also known as multiset combinations. We need to find how many ways we can choose 3 scoops of ice cream from 4 distinct flavors, where the order of selection does not matter, and repetition of flavors is allowed.

This can be calculated using the formula for combinations with repetition:

$$\binom{n+r-1}{r}$$

where:

n is the number of distinct items (in this case, 4 flavors of ice-cream),

r is the number of selections (in this case, 3 scoops of ice-cream).

Substituting the values:

$$\binom{4+3-1}{3} = \binom{6}{3} = \frac{6 \times 5 \times 4}{3 \times 2 \times 1} = 20.$$

Therefore, the correct answer is (B) 20.

Quick Tip

When calculating combinations with repetition, use the formula: $\binom{n+r-1}{r}$ where n is the number of distinct items and r is the number of selections.

Engineering Mathematics and Core Discipline

11. Suppose a program is running on a non-pipelined single processor computer system. The computer is connected to an external device that can interrupt the processor asynchronously. The processor needs to execute the interrupt service routine (ISR) to serve this interrupt. The following steps (not necessarily in order) are taken by the processor when the interrupt arrives:

(i) The processor saves the content of the program counter.

(ii) The program counter is loaded with the start address of the ISR.

(iii) The processor finishes the present instruction.

Which ONE of the following is the CORRECT sequence of steps?

(A) (iii), (i), (ii)

(B) (i), (ii), (iii)

(C) (i), (ii), (iii)

(D) (iii), (ii), (i)

Correct Answer: (A) (iii), (i), (ii)

Solution:

When an interrupt occurs in a non-pipelined processor, the processor follows a specific sequence of steps to handle the interrupt. The general process is as follows:

1. Step iii: The processor finishes the present instruction. This is necessary because the processor cannot leave the current instruction unfinished when the interrupt occurs. It must complete the instruction before handling the interrupt.

2. Step i: After completing the current instruction, the processor saves the content of the program counter (PC). This is crucial because the processor needs to remember the location

in the program where it left off, so it can resume from that point once the interrupt has been handled.

3. Step ii: The program counter is then loaded with the start address of the interrupt service routine (ISR). This step ensures that the processor starts executing the ISR instead of continuing the normal program flow.

Thus, the correct order is: 1. Finish the present instruction (iii), 2. Save the content of the program counter (i), 3. Load the program counter with the ISR address (ii).

Therefore, the correct answer is (A) (iii), (i), (ii).

Quick Tip
When dealing with interrupts, remember that the current instruction must be completed first. After that, the state (program counter) is saved, and the ISR address is loaded into the program counter.

---

12. Which ONE of the following statements is FALSE regarding the symbol table?

(A) Symbol table is responsible for keeping track of the scope of variables.

(B) Symbol table can be implemented using a binary search tree.

(C) Symbol table is not required after the parsing phase.

(D) Symbol table is created during the lexical analysis phase.

Correct Answer: (C) Symbol table is not required after the parsing phase.

Solution: The symbol table is essential throughout the compilation process. It helps in tracking the scope and attributes of variables.

It is used during the lexical analysis phase, where identifiers are recorded.

It is also required during later phases like code generation.

Statement (C) is false because the symbol table is still necessary after parsing, during semantic analysis and code generation.

Therefore, the correct answer is $\boxed{C}$.

Quick Tip

The symbol table plays a critical role in managing variable scope and type information throughout the compilation process.

---

13. Which ONE of the following techniques used in compiler code optimization uses live variable analysis?

(A) Run-time function call management

(B) Register assignment to variables

(C) Strength reduction

(D) Constant folding

Correct Answer: (B) Register assignment to variables

Solution: Live variable analysis is used to determine which variables hold values that may be used in the future.

It is especially useful for register allocation, where variables that are ”live” (i.e., needed later) must be preserved in registers.

Strength reduction and constant folding do not directly involve live variable analysis; they are optimizations based on algebraic transformations or constant expressions.

Thus, the correct answer is $\boxed{B}$.

Quick Tip

Live variable analysis helps optimize the usage of registers by identifying variables that are still in use and need to be preserved.

---

14. Consider a demand paging memory management system with 32-bit logical address, 20-bit physical address, and page size of 2048 bytes. Assuming that the memory is byte addressable, what is the maximum number of entries in the page table?

(A) 2^{21}

(B) 2^{20}

(C) 2^{22}

(D) 2^{24}

Correct Answer: (A) 2^{21}

Solution: The logical address is 32 bits, and the page size is 2048 bytes (2^{11} bytes).
The number of pages in the system can be calculated by dividing the total address space by the page size:

$$\frac{2^{32}}{2^{11}} = 2^{21}.$$

The page table must have an entry for each page, so the number of entries in the page table is 2^{21}. Thus, the correct answer is $\boxed{A}$.

Quick Tip
To calculate the number of entries in the page table, divide the total address space by the page size.

---

15. A schedule of three database transactions T_1, T_2, and T_3 is shown. $R_i(A)$ and $W_i(A)$ denote read and write of data item A by transaction T_i, $i = 1, 2, 3$. The transaction T_1 aborts at the end. Which other transaction(s) will be required to be rolled back?

$\mathrm{R}_1(X)W_1(Y)R_2(X)R_2(Y)R_3(Y)ABORT(T_1)$

(A) Only T_2

(B) Only T_3

(C) Both T_2 and T_3

(D) Neither T_2 nor T_3

Correct Answer: (C) Both T_2 and T_3

Solution:

In the given transaction schedule, transaction T_1 writes to data item Y and then aborts. Since T_1 aborts, any transaction that has read or written data modified by T_1 must be rolled back to maintain the consistency of the database.

Let's analyze the transactions:

Transaction T_1 writes to Y (i.e., $W_1(Y)$).

Transaction T_2 reads Y (i.e., $R_2(Y)$).

Transaction T_3 also reads Y (i.e., $R_3(Y)$).

Since both T_2 and T_3 read Y, which was modified by T_1, they will be impacted by the abort of T_1. As a result, both T_2 and T_3 must be rolled back to preserve the consistency of the database.

Therefore, the correct answer is (C) Both T_2 and T_3.

Quick Tip
In transaction management, when one transaction writes to a data item and another transaction reads or writes the same item, the abort of the first transaction often causes the second transaction to be rolled back. Always check for read-write dependencies.

---

16. Identify the ONE CORRECT matching between the OSI layers and their corresponding functionalities as shown.

OSI Layers	Functionalities
(a) Network layer	(I) Packet routing
(b) Transport layer	(II) Framing and error handling
(c) Datalink layer	(III) Host to host communication

(A) (a)-(I), (b)-(II), (c)-(III)

(B) (a)-(I), (b)-(III), (c)-(II)

(C) (a)-(II), (b)-(I), (c)-(III)

(D) (a)-(III), (b)-(II), (c)-(I)

Correct Answer: (B) (a)-(I), (b)-(III), (c)-(II)

Solution:

Let's revisit the OSI layers and their corresponding functionalities:

Network layer (a) is responsible for packet routing (I). It determines the best path for data to travel across multiple networks. This function corresponds to packet routing, which is the primary responsibility of the Network layer. So, the correct matching for the Network layer

is (a)-(I).

Transport layer (b) is responsible for host-to-host communication (III). It provides reliable communication between two systems or devices on different networks, ensuring that the data is properly delivered and received. Thus, the Transport layer corresponds to host-to-host communication (III).

Datalink layer (c) is responsible for framing and error handling (II). It prepares the data for transmission over the physical medium and handles error detection and correction. The Datalink layer is primarily concerned with how data is framed and transmitted over a single link, so the correct matching for the Datalink layer is (c)-(II).

Thus, the correct matching of layers and functionalities is:

(a) Network layer with (I) Packet routing,

(b) Transport layer with (III) Host to host communication,

(c) Datalink layer with (II) Framing and error handling.

Therefore, the correct answer is (B) (a)-(I), (b)-(III), (c)-(II).

Quick Tip
When understanding the OSI model: The Network layer is focused on routing packets between networks. The Transport layer ensures end-to-end communication between hosts. The Datalink layer ensures data is correctly framed for transmission and handles error correction.

---

17. g(·) is a function from A to B, f(·) is a function from B to C, and their composition defined as f(g(·)) is a mapping from A to C. If f(·) and f(g(·)) are onto (surjective) functions, which ONE of the following is TRUE about the function g(·)?

(A) g(·) must be an onto (surjective) function.

(B) g(·) must be a one-to-one (injective) function.

(C) g(·) must be a bijective function, that is, both one-to-one and onto.

(D) g(·) is not required to be a one-to-one or onto function.

Correct Answer: (D) g(·) is not required to be a one-to-one or onto function.

Solution: We are given that both $f()$ and $f(g())$ are surjective (onto) functions.
$f(g(x))$ represents the composition of functions f and g, where g maps from A to B and f maps from B to C. The composition $f(g(x))$ maps from A to C.
Let's analyze the problem:
The fact that $f(g(x))$ is surjective means that for every element in C, there is an element in A that maps to it through the composition of f and g.
For $f(g(x))$ to be surjective, the image of $g(x)$ (which lies in B) must cover all the elements in the domain of f, i.e., f must cover the entire set C.
However, $g(x)$ does not need to be surjective (onto) because it is not required to cover the entire set C, just the relevant part that f needs to map onto C. Therefore, $g(x)$ does not need to be injective or surjective for the composition to be surjective.
As a result, $g(x)$ is not required to be either one-to-one (injective) or onto (surjective).
Thus, the correct answer is $\boxed{D}$.

Quick Tip
For composition to be surjective, the second function f must cover the entire codomain. The first function g does not necessarily need to be surjective or injective.

---

18. Let G be any undirected graph with positive edge weights, and T be a minimum spanning tree of G. For any two vertices, u and v, let $d_1(u, v)$ and $d_2(u, v)$ be the shortest distances between u and v in G and T, respectively. Which ONE of the following options is CORRECT for all possible G, T, u, and v?

(A) $d_1(u, v) = d_2(u, v)$

(B) $d_1(u, v) \leq d_2(u, v)$

(C) $d_1(u, v) \geq d_2(u, v)$

(D) $d_1(u, v) \neq d_2(u, v)$

Correct Answer: (B) $d_1(u, v) \leq d_2(u, v)$

Solution: $d_1(u, v)$ represents the shortest path between vertices u and v in the original graph

G, while $d_2(u, v)$ represents the shortest path between the same vertices in the minimum spanning tree (MST) T.

Key Concept: A minimum spanning tree (MST) is a subgraph of G that connects all the vertices together with the minimum possible total edge weight. However, it does not necessarily preserve the shortest paths between all pairs of vertices.

When we compute $d_1(u, v)$, it represents the shortest path in the original graph G, which can use any edges of G, including some that are excluded from the MST. In contrast, $d_2(u, v)$ represents the shortest path in the MST T, where only edges included in the MST are used.

Important property of MST: Since the MST includes only a subset of the edges of G, it is possible for the shortest path in G to be shorter than or equal to the shortest path in the MST. The MST may exclude edges that could reduce the distance between two vertices, so the shortest path in the MST is **greater than or equal** to the shortest path in the original graph.

Thus, the correct relationship is:

$$d_1(u, v) \leq d_2(u, v)$$

This means the shortest path in the original graph G is always less than or equal to the shortest path in the minimum spanning tree T.

Therefore, the correct answer is $\boxed{B}$.

Quick Tip
In an MST, the shortest path between two vertices in the original graph G can be shorter or equal to the shortest path in the MST. The MST may exclude certain edges that could shorten the path.

---

19. Consider the following context-free grammar G, where S, A, and B are the variables (non-terminals), a and b are the terminal symbols, S is the start variable, and the rules of G are described as:

$$S \to aaB \mid Abb$$

$$A \to a \mid AaA$$

$$B \to b \mid bB$$

Which ONE of the languages $L(G)$ is accepted by G?

(A) $L(G) = \{a^2b^n \mid n \geq 1\} \cup \{a^nb^2 \mid n \geq 1\}$

(B) $L(G) = \{a^nb^{2n} \mid n \geq 1\} \cup \{a^{2n}b^n \mid n \geq 1\}$

(C) $L(G) = \{a^nb^n \mid n \geq 1\}$

(D) $L(G) = \{a^{2n}b^{2n} \mid n \geq 1\}$

Correct Answer: (A) $L(G) = \{a^2b^n \mid n \geq 1\} \cup \{a^nb^2 \mid n \geq 1\}$

Solution:

Let's examine the production rules and how they generate strings:

1. Production Rule 1: $S \to aaB \mid Abb$

The rule $S \to aaB$ produces strings that start with "aa" followed by a string generated by B. Since B generates any number of b's, this can result in strings like aab^n, where $n \geq 1$. The rule $S \to Abb$ generates strings starting with "A" followed by two b's. Since A generates strings of 'a's (from $A \to a \mid AaA$), this results in strings like a^nb^2, where $n \geq 1$.

2. Production Rule 2: $A \to a \mid AaA$

The production $A \to a$ generates a single a.

The recursive production $A \to AaA$ means that A can generate any number of a's, leading to strings like a^n.

3. Production Rule 3: $B \to b \mid bB$

This generates any number of b's, starting with one b, as B recursively produces b's.

Derivation of Strings:

From $S \to aaB$, we generate strings of the form aab^n, where $n \geq 1$. For example:

aab, $aabb$, $aabbb$, etc. These strings match the pattern a^2b^n, where $n \geq 1$.

From $S \to Abb$, we generate strings of the form a^nb^2, where $n \geq 1$. For example: ab^2, a^2b^2, a^3b^2, etc. These strings match the pattern a^nb^2, where $n \geq 1$.

Thus, the correct language $L(G)$ is the union of the two patterns:

$L(G) = \{a^2b^n \mid n \geq 1\} \cup \{a^nb^2 \mid n \geq 1\}$.

Therefore, the correct answer is (A).

Quick Tip
In context-free grammar problems, carefully analyze the production rules step by step and look at how the rules combine to form strings. Always check for patterns in the generated strings to match the language correctly.

---

20. Consider the following recurrence relation:

$$T(n) = 2T(n-1) + n2^n \text{ for } n > 0,\ T(0) = 1.$$

Which ONE of the following options is CORRECT?

(A) $T(n) = \Theta(n^2 2^n)$

(B) $T(n) = \Theta(n2^n)$

(C) $T(n) = \Theta((\log n)^2 2^n)$

(D) $T(n) = \Theta(4^n)$

Correct Answer: (A) $T(n) = \Theta(n^2 2^n)$

Solution: The recurrence relation given is:

$$T(n) = 2T(n-1) + n2^n.$$

We will solve this recurrence using the recursion-tree method and substitution method.

1. Step 1: Unfolding the recurrence:

Let's start by expanding the recurrence:

$$T(n) = 2T(n-1) + n2^n$$

Substituting for $T(n-1)$:

$$T(n) = 2[2T(n-2) + (n-1)2^{n-1}] + n2^n = 2^2 T(n-2) + 2(n-1)2^{n-1} + n2^n$$

Next, expand for $T(n-2)$:

$$T(n) = 2^2[2T(n-3) + (n-2)2^{n-2}] + 2(n-1)2^{n-1} + n2^n$$

This process continues. After k steps, we get:

$$T(n) = 2^k T(n-k) + \sum_{i=0}^{k-1} 2^i (n-i)2^{n-i}.$$

2. Step 2: Solving the recurrence:

As $k \to n$, the last term where $T(0) = 1$ gives a small constant. The dominant term is the sum:

$$\sum_{i=0}^{n-1} 2^i (n-i) 2^{n-i}.$$

This simplifies to:

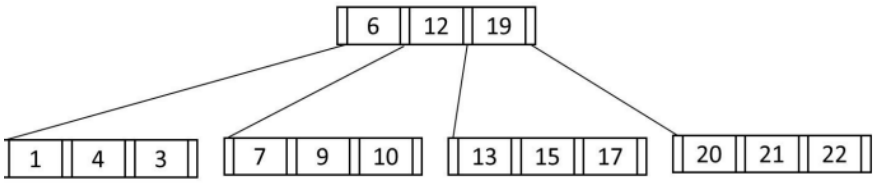
$$T(n) = \Theta(n^2 2^n).$$

Thus, the correct answer is $\boxed{A}$.

Quick Tip
When solving recurrences with the recursion-tree method, focus on the number of levels and the work done at each level. The dominant term in the sum usually gives the asymptotic complexity.

---

21. Consider the following B^+ tree with 5 nodes, in which a node can store at most 3 key values. The value 23 is now inserted in the B^+ tree. Which of the following options(s) is/are CORRECT?



(A) None of the nodes will split.

(B) At least one node will split and redistribute.

(C) The total number of nodes will remain same.

(D) The height of the tree will increase.

Correct Answer: (B), (D)

Solution:

In this B^+ tree, each node can hold at most 3 keys. Initially, the tree has 5 nodes. The insertion of 23 will cause a node split because adding this new value would exceed the capacity of the current node. When a node split occurs, a new node is created, and this may also cause the height of the tree to increase if the split affects the root.

Thus, options (B) and (D) are correct:

(B) At least one node will split and redistribute.

(D) The height of the tree will increase if the root node splits.

Why Other Options are Incorrect:

Option (A): Incorrect because the node will split due to the insertion of 23.

Option (C): Incorrect because the total number of nodes will increase when a node splits.

Quick Tip
When a node in a B^+ tree exceeds its key capacity, a split occurs, which might increase the height of the tree if the root is affected.

---

22. Consider the 3-way handshaking protocol for TCP connection establishment. Let the three packets exchanged during the connection establishment be denoted as P1, P2, and P3, in order. Which of the following option(s) is/are TRUE with respect to TCP header flags that are set in the packets?

(A) P3: SYN = 1, ACK = 1

(B) P2: SYN = 1, ACK = 1

(C) P2: SYN = 0, ACK = 1

(D) P1: SYN = 1

Correct Answer: (B), (D)

Solution:

In the TCP 3-way handshake:

P1 (SYN = 1): The first packet (P1) is sent by the client to initiate the connection, and it has the SYN flag set to 1.

P2 (SYN = 1, ACK = 1): The second packet (P2) is sent by the server to acknowledge the client's request. It has both SYN and ACK flags set to 1.

P3 (ACK = 1): The third packet (P3) is sent by the client to acknowledge the server's response. It has only the ACK flag set to 1.

Thus, options (B) and (D) are correct:

(B) P2: SYN = 1, ACK = 1.

(D) P1: SYN = 1.

Why Other Options are Incorrect:

Option (A): Incorrect because P3 only has the ACK flag set, not both SYN and ACK.

Option (C): Incorrect because P2 has both SYN and ACK flags set, not just ACK.

Quick Tip
In the TCP 3-way handshake, the first packet (P1) has only SYN set, the second packet (P2) has both SYN and ACK set, and the third packet (P3) has only ACK set.

---

23. Consider the given system of linear equations for variables x and y, where k is a real-valued constant. Which of the following option(s) is/are CORRECT?

$$x + ky = 1$$

$$kx + y = -1$$

(A) There is exactly one value of k for which the above system of equations has no solution.
(B) There exist an infinite number of values of k for which the system of equations has no solution.
(C) There exists exactly one value of k for which the system of equations has exactly one solution.
(D) There exists exactly one value of k for which the system of equations has an infinite number of solutions.

Correct Answer: (A) There is exactly one value of k for which the above system of equations has no solution. & (D) There exists exactly one value of k for which the system of equations has an infinite number of solutions.

Solution: For this system, the determinant of the coefficient matrix is $1 - k^2$.
For the system to have no solution, we set the determinant equal to zero:

$$1 - k^2 = 0 \quad \Rightarrow \quad k = \pm 1.$$

Hence, for $k = 1$, the system has no solution. For $k \neq 1$, the system has exactly one solution. For $k = -1$, the system has infinitely many solutions. Therefore, the correct answer is $\boxed{A}$ & $\boxed{D}$.

Quick Tip
When solving systems of equations, check the determinant of the coefficient matrix to determine if the system has a unique solution, no solution, or infinitely many solutions.

---

24. Let X be a 3-variable Boolean function that produces output as '1' when at least two of the input variables are '1'. Which of the following statement(s) is/are CORRECT, where a, b, c, d, e are Boolean variables?

(A) $X(a, b, X(c, d, e)) = X(X(a, b, c), d, e)$

(B) $X(a, b, X(a, b, c)) = X(a, b, c)$

(C) $X(a, b, X(a, c, d)) = (X(a, b, a)$ AND $X(c, d, c))$

(D) $X(a, b, c) = X(a, X(a, b, c), X(a, c, c))$

Correct Answer: (B) $X(a, b, X(a, b, c)) = X(a, b, c)$ & (D) $X(a, b, c) = X(a, X(a, b, c), X(a, c, c))$

Solution:

Option (A): Incorrect.

Option (B): Correct. The composition $X(a, b, X(a, b, c))$ simplifies to $X(a, b, c)$ due to the function structure.

Option (C): Incorrect.

Option (D): Correct. The Boolean function $X(a, b, c)$ can be rewritten as $X(a, X(a, b, c), X(a, c, c))$.

Thus, the correct answers are $\boxed{B}$ & $\boxed{D}$.

Quick Tip
In Boolean functions, simplify the expression using properties of logical functions like AND, OR, and NOT.

---

25. The number -6 can be represented as 1010 in 4-bit 2's complement representation. Which of the following is/are CORRECT 2's complement representation(s) of -6?

(A) $1000\,1010$ in 8-bits

(B) $1111\,1010$ in 8-bits

(C) $1000\,0000\,0000\,1010$ in 16-bits

(D) $1111\,1111\,1111\,1010$ in 16-bits

Correct Answer: (B), (D)

Solution:

The 2's complement representation of a negative number can be found by inverting the bits of the positive value and then adding 1.

For -6 in 4-bit 2's complement:

1. The binary representation of 6 is 0110.

2. Invert the bits: 1001.

3. Add 1: 1010.

Now, let's consider the extended bit sizes:

In 8-bits: The correct 2's complement representation of -6 would be $1111\,1010$. So, Option (B) is correct.

In 16-bits: The correct 2's complement representation of -6 would be $1111\,1111\,1111\,1010$. So, Option (D) is correct.

Thus, the correct answers are (B) and (D).

Quick Tip
When representing negative numbers in 2's complement, always extend the most significant bit (MSB) to the left when increasing the bit size. This ensures the number stays correctly represented.

---

26. Which of the following statement(s) is/are TRUE for any binary search tree (BST) having n distinct integers?

(A) The maximum length of a path from the root node to any other node is $(n-1)$.

(B) An inorder traversal will always produce a sorted sequence of elements.

(C) Finding an element takes $O(\log_2 n)$ time in the worst case.

(D) Every BST is also a Min-Heap.

Correct Answer: (A), (B)

Solution:

Let's analyze each statement:

(A) The maximum length of a path from the root node to any other node is $(n-1)$: This statement is correct in the worst-case scenario, where the tree is skewed (like a linked list). In such a case, the maximum path length from the root node to any other node will be $n-1$, as each node only has one child.

(B) An inorder traversal will always produce a sorted sequence of elements: This statement is correct. By definition, an inorder traversal of a binary search tree (BST) visits nodes in ascending order, producing a sorted sequence of elements.

(C) Finding an element takes $O(\log_2 n)$ time in the worst case: This statement is incorrect. In the worst case (for a skewed tree), the time complexity for finding an element can be $O(n)$, not $O(\log n)$.

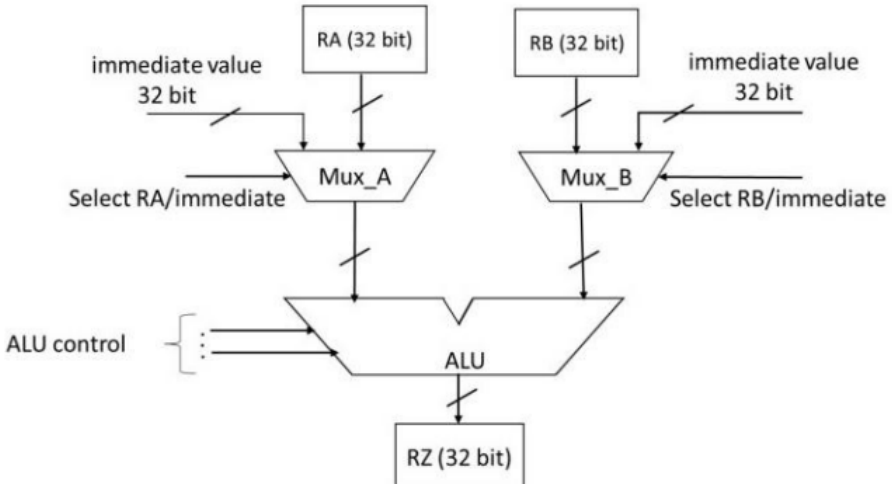
(D) Every BST is also a Min-Heap: This statement is incorrect. A Min-Heap is a complete binary tree where the value of each node is less than or equal to the values of its children. A binary search tree (BST) does not necessarily satisfy the Min-Heap property.

Thus, the correct answers are (A) and (B).

Quick Tip
In a binary search tree, an inorder traversal always results in sorted elements. Additionally, in the worst case of a skewed tree, the maximum height can be $n-1$, leading to a linear time complexity for finding elements.

---

27. A partial data path of a processor is given in the figure, where RA, RB, and RZ are 32-bit registers. Which option(s) is/are CORRECT related to arithmetic operations using the data path as shown?



(A) The data path can implement arithmetic operations involving two registers.
(B) The data path can implement arithmetic operations involving one register and one immediate value.
(C) The data path can implement arithmetic operations involving two immediate values.
(D) The data path can only implement arithmetic operations involving one register and one immediate value.

Correct Answer: (A) The data path can implement arithmetic operations involving two registers. & (B) The data path can implement arithmetic operations involving one register and one immediate value. & (C) The data path can implement arithmetic operations involving two immediate values.

Solution: The data path can perform arithmetic operations involving two registers, as both RA and RB are connected to the ALU.
The data path can also perform arithmetic operations involving one register and one immediate value, as the multiplexers allow immediate values to be selected.
The data path can implement arithmetic operations involving two immediate values, as both Mux_A and Mux_B allow immediate values to be passed to the ALU.
Thus, the correct answers are $\boxed{A}$, $\boxed{B}$, and $\boxed{C}$.

Quick Tip
In processor data paths, multiplexers are used to select between different inputs for operations such as addition or subtraction, involving registers or immediate values.

28. A regular language L is accepted by a non-deterministic finite automaton (NFA) with n states. Which of the following statement(s) is/are FALSE?

(A) L may have an accepting NFA with $< n$ states.

(B) L may have an accepting DFA with $< n$ states.

(C) There exists a DFA with $\leq 2^n$ states that accepts L.

(D) Every DFA that accepts L has $> 2^n$ states.

Correct Answer: (B) L may have an accepting DFA with $< n$ states. & (D) Every DFA that accepts L has $> 2^n$ states.

Solution: Option (A): This is true. An NFA can potentially accept a language using fewer states due to non-determinism, but for DFA equivalence, the number of states could be larger.
Option (B): This is false. A DFA may require more states than an NFA to accept the same language. For any language accepted by an NFA with n states, the equivalent DFA may have up to 2^n states, but it cannot have fewer states.
Option (C): This is true. There is always a DFA with 2^n states for any language accepted by an NFA with n states (since $NFA \rightarrow DFA$ conversion can potentially lead to exponential state growth).
Option (D): This is false. A DFA does not always need more than 2^n states. It is possible for the DFA to have fewer states than 2^n, depending on the language.
Thus, the correct answers are $\boxed{D}$.

Quick Tip
The conversion from NFA to DFA can cause an exponential increase in the number of states. However, some DFA constructions can still have fewer states depending on the language.

29. Suppose in a multiprogramming environment, the following C program segment is executed. A process goes into the I/O queue whenever an I/O related operation is performed. Assume that there will always be a context switch whenever a process requests an I/O, and also whenever the process returns from an I/O. The number of

times the process will enter the ready queue during its lifetime (not counting the time the process enters the ready queue when it is run initially) is _____ (Answer in integer).

```
int main()
 {
 int x=0, i=0;
 scanf("%d",&x);
 for(i=0; i<20; i++)
 {
 x = x+20;
 printf("%d
n", x);
 }
 return 0;
}
```

Solution:

Let's break down the operations and their implications on the ready and I/O queues:

Step 1: Initial Entry into the Ready Queue:

When the process is first run, it enters the ready queue. This is counted as the first entry into the ready queue.

Step 2: Entering the I/O Queue:

The 'scanf("%d", &x)' function is called in the program, which is an I/O operation. When the process executes 'scanf()', it enters the I/O queue because it's waiting for user input. This causes a context switch.

Step 3: Returning from I/O:

After the user provides input and the process continues, it exits the I/O queue and enters the ready queue. This is a context switch from the I/O queue back to the ready queue. Therefore, the process enters the ready queue again after the I/O operation.

Step 4: Repeating for Each Iteration:

The 'for' loop runs 20 times. Inside each iteration, the program performs the 'printf("%d", x)' operation, which is another I/O operation. Each time 'printf()' is executed, the process

goes to the I/O queue and returns to the ready queue once the I/O operation is completed.

Calculation of Ready Queue Entries:

For the initial entry into the ready queue, the process enters once when it is first run.

During each iteration of the loop, there are two key events:

1. The process enters the I/O queue when it calls ‘printf()‘.

2. The process returns from the I/O queue to the ready queue after ‘printf()‘ completes.

Since the loop runs 20 times, each iteration results in one entry into the I/O queue and one return to the ready queue.

Thus, in total:

The process enters the ready queue once initially.

It enters the ready queue 20 times after each I/O operation during the loop.

Therefore, the process will enter the ready queue a total of:

$$1\,(\text{initial}) + 20\,(\text{one for each iteration}) = 21\,\text{times}.$$

Thus, the correct answer is 21.

Quick Tip
Each I/O operation (like ‘scanf()‘ and ‘printf()‘) causes a context switch, where the process moves between the I/O queue and the ready queue. Count the number of iterations and operations to determine the number of context switches.

---

30. Let S be the set of all ternary strings defined over the alphabet $\{a, b, c\}$. Consider all strings in S that contain at least one occurrence of two consecutive symbols, that is, ”aa”, ”bb”, or ”cc”. The number of such strings of length 5 that are possible is _____.

Solution: We are asked to find the number of ternary strings of length 5 that contain at least one occurrence of two consecutive symbols, specifically ”aa”, ”bb”, or ”cc”.

1. Total number of strings in S:

The total number of ternary strings of length 5 is:

$$3^5 = 243$$

because each position in the string can be filled with any of the three symbols a, b, or c.

2. Strings without consecutive symbols:

Next, we need to subtract the number of strings that do not contain any consecutive symbols.

If the string does not contain consecutive identical symbols, we have the following choices:

The first character can be any of a, b, or c.

The second character must be different from the first.

The third character must be different from the second, and so on.

So, the number of such strings is:

$$3 \times 2 \times 2 \times 2 \times 2 = 3 \times 2^4 = 48$$

because the first character has 3 choices, and each subsequent character has 2 choices (to avoid repeating the previous character).

3. Strings with at least one pair of consecutive identical symbols:

The number of strings that contain at least one occurrence of two consecutive symbols is the complement of the number of strings without consecutive symbols. Thus, the number of strings with at least one consecutive pair is:

$$243 - 48 = 195$$

Thus, the number of such strings of length 5 is $\boxed{195}$.

Quick Tip
To count strings with specific patterns (like consecutive identical symbols), use the complement rule by first counting the total number of strings and then subtracting the number of strings that do not contain the desired pattern.

---

31. Consider the given function $f(x)$.

$$f(x) = \begin{cases} ax + b & \text{for } x < 1 \\ x^3 + x^2 + 1 & \text{for } x \geq 1 \end{cases}$$

If the function is differentiable everywhere, the value of b must be _____ (rounded off to one decimal place).

Solution:

For the function $f(x)$ to be differentiable everywhere, it must be both continuous and differentiable at $x = 1$ (the point where the piecewise function changes).

Step 1: Continuity Condition

First, let's ensure that the function is continuous at $x = 1$.

For continuity at $x = 1$, we need the left-hand limit and the right-hand limit to be equal at $x = 1$.

The function for $x < 1$ is $f(x) = ax + b$.

The function for $x \geq 1$ is $f(x) = x^3 + x^2 + 1$.

We want the limits from both sides to match at $x = 1$:

Left-hand limit as $x \to 1^-$:

$$\lim_{x \to 1^-} f(x) = a(1) + b = a + b$$

Right-hand limit as $x \to 1^+$:

$$\lim_{x \to 1^+} f(x) = 1^3 + 1^2 + 1 = 1 + 1 + 1 = 3$$

For continuity at $x = 1$, we require:

$$a + b = 3$$

So, we have the equation:

$$a + b = 3 \quad \text{(Equation 1)}$$

Step 2: Differentiability Condition

Next, we ensure that the function is differentiable at $x = 1$.

For differentiability at $x = 1$, the derivative from the left must be equal to the derivative from the right.

The derivative of $f(x) = ax + b$ for $x < 1$ is:

$$f'(x) = a$$

The derivative of $f(x) = x^3 + x^2 + 1$ for $x \geq 1$ is:

$$f'(x) = 3x^2 + 2x$$

At $x = 1$, we want the left-hand derivative and the right-hand derivative to be equal:

$$a = 3(1)^2 + 2(1) = 3 + 2 = 5$$

So, from the differentiability condition, we have:

$$a = 5 \quad \text{(Equation 2)}$$

Step 3: Solving for b From Equation 2, we know $a = 5$. Substituting $a = 5$ into Equation 1:

$$5 + b = 3$$

Solving for b:

$$b = 3 - 5 = -2$$

Thus, the value of b is -2.

Quick Tip
For piecewise functions to be differentiable everywhere, they must be continuous and have equal derivatives at the point where the function changes its definition. Always check both the continuity and differentiability conditions at that point.

---

30. A box contains 5 coins: 4 regular coins and 1 fake coin. When a regular coin is tossed, the probability $P(\textbf{head}) = 0.5$, and for a fake coin, $P(\textbf{head}) = 1$. You pick a coin at random and toss it twice, and get two heads. The probability that the coin you have chosen is the fake coin is _____.

Solution: We are asked to find the probability that the coin is fake, given that two heads were tossed.

This is a problem of conditional probability. We can use Bayes' Theorem to solve it. Bayes' Theorem is given by:

$$P(\text{Fake} \mid \text{Two heads}) = \frac{P(\text{Two heads} \mid \text{Fake})P(\text{Fake})}{P(\text{Two heads})}$$

We will calculate each term in this formula:

1. Prior probability of choosing the fake coin: Since there is 1 fake coin out of 5 coins, the probability of selecting the fake coin is:

$$P(\text{Fake}) = \frac{1}{5}$$

2. Likelihood of getting two heads given the fake coin: If the fake coin is chosen, the probability of getting heads on each toss is 1. Therefore, the probability of getting two heads is:

$$P(\text{Two heads} \mid \text{Fake}) = 1 \times 1 = 1$$

3. Likelihood of getting two heads given a regular coin: If a regular coin is chosen, the probability of getting heads on each toss is 0.5. Therefore, the probability of getting two heads is:

$$P(\text{Two heads} \mid \text{Regular}) = 0.5 \times 0.5 = 0.25$$

4. Prior probability of choosing a regular coin: There are 4 regular coins out of 5, so the probability of selecting a regular coin is:

$$P(\text{Regular}) = \frac{4}{5}$$

5. Total probability of getting two heads: The total probability of getting two heads is given by the law of total probability:

$$P(\text{Two heads}) = P(\text{Two heads} \mid \text{Fake})P(\text{Fake}) + P(\text{Two heads} \mid \text{Regular})P(\text{Regular})$$

Substituting the values:

$$P(\text{Two heads}) = 1 \times \frac{1}{5} + 0.25 \times \frac{4}{5} = \frac{1}{5} + \frac{1}{5} = \frac{2}{5}$$

6. Bayes' Theorem: Now, applying Bayes' Theorem:

$$P(\text{Fake} \mid \text{Two heads}) = \frac{P(\text{Two heads} \mid \text{Fake})P(\text{Fake})}{P(\text{Two heads})} = \frac{1 \times \frac{1}{5}}{\frac{2}{5}} = \frac{1}{2}$$

Thus, the probability that the coin you have chosen is the fake coin is $\boxed{0.50}$.

Quick Tip
Bayes' Theorem allows us to update the probability of an event based on new evidence. In this case, it helps us find the probability of having chosen the fake coin, given the outcome of two heads.

---

33. The pseudocode of a function $fun()$ is given below:

fun(int A[0, . . . , n-1]){

for $i = 0$ to $n - 2$

for j = 0 to n - i - 2

if (A[j] ¿ A[j+1]) then swap A[j] and A[j+1]

}

Let $A[0, \ldots, 29]$ be an array storing 30 distinct integers in descending order. The number of swap operations that will be performed, if the function $fun()$ is called with $A[0, \ldots, 29]$ as argument, is _____ (Answer in integer).

Solution:

The pseudocode provided is an implementation of Bubble Sort. Let's analyze the number of swap operations in the case where the array is initially sorted in descending order:

The outer loop runs from $i = 0$ to $n - 2$, where $n = 30$. Therefore, the outer loop runs 29 times.

The inner loop runs from $j = 0$ to $n - i - 2$, which means the number of iterations decreases as i increases.

For each value of i, the number of comparisons in the inner loop is:

$$n - i - 2$$

Thus, the total number of comparisons (and therefore the number of potential swaps) is the sum of the number of comparisons for each value of i from 0 to $n - 2$.

Step 1: Total Number of Comparisons

The number of comparisons for each value of i is:

For $i = 0$, the inner loop runs $n - 1$ times.

For $i = 1$, the inner loop runs $n - 2$ times.

For $i = 2$, the inner loop runs $n - 3$ times.

...

For $i = n - 2$, the inner loop runs 1 time.

Thus, the total number of comparisons is the sum of the first $n-1$ integers:

$$\text{Total comparisons} = (n-1) + (n-2) + \ldots + 1 = \frac{n(n-1)}{2}$$

For $n = 30$:

$$\text{Total comparisons} = \frac{30 \times 29}{2} = 435$$

Step 2: Number of Swaps

In a Bubble Sort, a swap occurs every time a comparison finds that $A[j] > A[j+1]$. Since the array is initially sorted in descending order, every comparison will result in a swap.

Thus, the number of swaps is equal to the total number of comparisons, which is 435.

Therefore, the number of swap operations performed is 435.

Quick Tip
In Bubble Sort, the number of swaps is equal to the number of comparisons when the array is initially sorted in descending order, as every comparison leads to a swap.

The given C program is as follows:

```
#include <stdio.h>
void foo(int p, int x) {
p = x;
}
int main() {
int z;
int a = 20, b = 25;
z = &a;
foo(z,b);
printf("%d",a);
```

```
    return 0;
}
```

The output of the given C program is _____. (Answer in integer)

Solution:

Let's analyze the program step by step:

1. Function 'foo': The function 'foo' takes two arguments: a pointer 'p' and an integer 'x'. Inside the function, the value of 'x' is assigned to the variable that 'p' is pointing to. In this case, 'p' points to 'a', so 'a' will be updated to the value of 'x', which is 25.

2. In the 'main' function:

'a' is initialized to 20, and 'b' is initialized to 25.

'z' is a pointer, and it is assigned the address of 'a', so 'z' now points to 'a'.

'foo(z, b)' is called, which passes 'z' (the address of 'a') and 'b' (which is 25) to the function.

- Inside 'foo', 'z = b' sets the value of 'a' to 25 because 'z' points to 'a'.

3. Output: After the call to 'foo', 'a' is updated to 25, and the 'printf' statement prints the value of 'a', which is now 25.

Thus, the output of the program is 25.

Quick Tip
In C, when you pass a pointer to a function, any modification to the value pointed to by the pointer will affect the original variable. Ensure you understand how pointers work when analyzing such programs.

---

35. The height of any rooted tree is defined as the maximum number of edges in the path from the root node to any leaf node. Suppose a Min-Heap T stores 32 keys. The height of T is _____ (Answer in integer).

Solution:

A Min-Heap is a complete binary tree where each level is filled completely, except possibly the last level, which is filled from left to right.

The height of a binary tree is the number of edges on the longest path from the root to any leaf node.

To calculate the height of the tree, we use the following formula:

For a complete binary tree with n nodes, the height h is the greatest integer such that $2^h \leq n$.

This is because the height of a binary tree is logarithmic with respect to the number of nodes.

Step 1: Calculate the height for 32 keys

The total number of nodes in the tree is 32, and we need to find the height. Since the tree is a complete binary tree, the number of nodes at height h is 2^h.

To find the height h, we calculate the greatest integer h such that:

$$2^h \leq 32$$

We know:

$$2^5 = 32$$

Thus, the height of the tree is 5.

Step 2: Confirm the number of edges

For a tree with height 5, the maximum number of edges from the root to any leaf node is 5, as there are 5 levels from the root to the deepest leaf.

Thus, the height of the Min-Heap storing 32 keys is 5.

Quick Tip
In a complete binary tree, the height is the number of edges from the root to the deepest leaf. The height is approximately $\log_2 n$, where n is the number of nodes.

---

36. Consider a memory system with 1M bytes of main memory and 16K bytes of cache memory. Assume that the processor generates a 20-bit memory address, and the cache block size is 16 bytes. If the cache uses direct mapping, how many bits will be required to store all the tag values? [Assume memory is byte addressable, 1K = 2^{10}, 1M = 2^{20}.]

(A) 6×2^{10}

(B) 8×2^{10}

(C) 2^{12}

(D) 2^{14}

Correct Answer: (A) 6×2^{10}

Solution: Given:

Main memory size = 1M bytes = 2^{20} bytes

Cache memory size = 16K bytes = 2^{14} bytes

Cache block size = 16 bytes = 2^4 bytes

1. Number of cache blocks:

$$\text{Number of cache blocks} = \frac{\text{Cache memory size}}{\text{Cache block size}} = \frac{2^{14}}{2^4} = 2^{10}.$$

2. Number of bits for block offset:

Since each block has 16 bytes, the number of bits required to address a byte within the block is:

$$\text{Block offset bits} = 4 \quad (\text{since } 2^4 = 16).$$

3. Number of bits for index:

The number of bits required to index the cache is the number of bits required to represent the number of cache blocks:

$$\text{Index bits} = 10 \quad (\text{since } 2^{10} = \text{number of cache blocks}).$$

4. Number of bits for the tag:

The total number of bits in the address is 20 (since the processor generates a 20-bit memory address). The address can be split into:

$$\text{Address bits} = \text{Tag bits} + \text{Index bits} + \text{Block offset bits}.$$

Therefore, the number of tag bits is:

$$\text{Tag bits} = 20 - 10 - 4 = 6.$$

Thus, the number of bits required to store all the tag values is $\boxed{6 \times 2^{10}}$.

Quick Tip

To calculate the number of tag bits, break the address into three parts: the tag, index, and block offset. Use the cache size and block size to determine the index and block offset bits.

---

37. A processor has 64 general-purpose registers and 50 distinct instruction types. An instruction is encoded in 32-bits. What is the maximum number of bits that can be used to store the immediate operand for the given instruction?

(A) 16

(B) 20

(C) 22

(D) 24

Correct Answer: (B) 20

Solution: Given:

Number of general-purpose registers = 64

Number of instruction types = 50

Instruction format size = 32 bits

1. Bits for register specification:

The number of bits required to specify one of the 64 general-purpose registers is:

$$\text{Register bits} = \log_2(64) = 6.$$

2. Bits for instruction type:

The number of bits required to specify one of the 50 instruction types is:

$$\text{Instruction bits} = \log_2(50) \approx 6.$$

3. Bits for the immediate operand:

The remaining bits in the 32-bit instruction format will be used to store the immediate operand:

$$\text{Immediate operand bits} = 32 - 6(\text{register bits}) - 6(\text{instruction bits}) = 20.$$

Thus, the maximum number of bits that can be used to store the immediate operand is $\boxed{20}$.

Quick Tip
When calculating the number of bits for the immediate operand, subtract the bits required for the instruction type and the registers from the total instruction size.

---

38. A computer has two processors, M_1 and M_2. Four processes P_1, P_2, P_3, P_4 with CPU bursts of 20, 16, 25, and 10 milliseconds, respectively, arrive at the same time and these are the only processes in the system. The scheduler uses non-preemptive priority scheduling, with priorities decided as follows:

- M_1 uses priority of execution for the processes as $P_1 > P_3 > P_2 > P_4$, i.e., P_1 has the highest priority and P_4 has the lowest.
- M_2 uses priority of execution for the processes as $P_2 > P_3 > P_4 > P_1$, i.e., P_2 has the highest priority and P_1 has the lowest.

A process P_i is scheduled to a processor M_k, if the processor is free and no other process P_j is waiting with higher priority. At any given point of time, a process can be allocated to any of the free processors without violating the execution priority rules. Ignore the context switch time. What will be the average waiting time of the processes in milliseconds?

(A) 9.00

(B) 8.75

(C) 6.50

(D) 7.50

Correct Answer: (A) 9.00

Solution:

We need to calculate the waiting time for each process and then compute the average waiting time.

Step 1: Initial Setup and Process Priorities

Processor M_1 has the priority order: $P_1 > P_3 > P_2 > P_4$.

Processor M_2 has the priority order: $P_2 > P_3 > P_4 > P_1$.

Each process has the following CPU burst times:

P_1 = 20 ms

P_2 = 16 ms

P_3 = 25 ms

P_4 = 10 ms

Step 2: Process Scheduling

Processor M_1 will start by scheduling P_1 because it has the highest priority in M_1's list. P_1 runs for 20 ms.

Processor M_2 will then schedule P_2, as it has the highest priority in M_2's list. P_2 runs for 16 ms.

At this point, the following processes remain:

P_3 (remaining time 25 ms)

P_4 (remaining time 10 ms)

- Processor M_1 will next schedule P_3, as it has the next highest priority in M_1's list. P_3 runs for 25 ms.

Processor M_2 will schedule P_4, as it has the next highest priority in M_2's list. P_4 runs for 10 ms.

Step 3: Calculating Waiting Times

Now, we calculate the waiting time for each process:

P_1: This process starts at time 0 and runs for 20 ms, so its waiting time is 0 ms.

P_2: This process starts after P_1 finishes on M_2, so its waiting time is 20 ms (waiting for P_1 to finish on M_1).

P_3: This process starts after P_2 finishes, so its waiting time is $20 + 16 = 36$ ms.

P_4: This process starts after both P_1 and P_2 finish, so its waiting time is $20 + 16 = 36$ ms.

Step 4: Calculating Average Waiting Time

The total waiting time is:

$$0 + 20 + 36 + 36 = 92\,\text{ms}$$

Now, the average waiting time is:

$$\frac{92}{4} = 9.00\,\text{ms}$$

Thus, the average waiting time is 9.00 ms.

Quick Tip
In priority scheduling, the waiting time depends on the order in which processes are executed. The waiting time for each process is the total time it waits for other higher-priority processes to finish.

---

39. Consider two relations describing teams and players in a sports league:

- `teams(tid, tname)`: `tid`, `tname` are team-id and team-name, respectively.
- `players(pid, pname, tid)`: `pid`, `pname`, and `tid` denote player-id, player-name, and the team-id of the player, respectively.

Which ONE of the following tuple relational calculus queries returns the name of the players who play for the team having tname as 'MI'?

(A) $\{p.pname \mid p \in \text{players} \wedge \exists t \in \text{teams} \wedge p.tid = t.tid \wedge t.tname = \text{'MI'}\}$

(B) $\{p.pname \mid p \in \text{teams} \wedge \exists t \in \text{players} \wedge p.tid = t.tid \wedge t.tname = \text{'MI'}\}$

(C) $\{p.pname \mid p \in \text{players} \wedge \exists t \in \text{teams} \wedge t.tname = \text{'MI'}\}$

(D) $\{p.pname \mid p \in \text{teams} \wedge \exists t \in \text{players} \wedge t.tname = \text{'MI'}\}$

Correct Answer: (A) $\{p.pname \mid p \in \text{players} \wedge \exists t \in \text{teams} \wedge p.tid = t.tid \wedge t.tname = \text{'MI'}\}$

Solution: We are looking for the names of players who play for the team with the name 'MI'. The relational calculus query must:

Access the `players` relation to get the player names.

Join the `players` relation with the `teams` relation using the common attribute `tid` (team id).

Filter the result where the team name (`tname`) is 'MI'.

Option (A): This query correctly returns the player names by finding players whose `tid` matches the `tid` of the team named 'MI'. This is the correct query.

Option (B): This query first accesses the `teams` relation and then tries to find the corresponding player names. However, the `players` relation should be accessed first to

retrieve player names, and then the team should be matched using the `tid`. Hence, this option is incorrect.

Option (C): This query accesses the `players` relation and finds players where the `tid` matches the team with name 'MI'. While it looks close, it misses the join condition between `players` and `teams` by not explicitly linking the `tid` of `players` and `teams`. Therefore, it is not precise enough.

Option (D): This query incorrectly begins with `teams` and tries to find the players. The player names should be accessed from the `players` relation first, and then the team details should be joined. This makes option (D) incorrect.

Thus, the correct answer is $\boxed{A}$.

Quick Tip
When using tuple relational calculus, ensure that you select the right relation first (in this case, `players`), and then perform the appropriate join and filtering to get the desired result.

---

40. A packet with the destination IP address 145.36.109.70 arrives at a router whose routing table is shown. Which interface will the packet be forwarded to?

Subnet Address	Subnet Mask (in CIDR notation)	Interface
145.36.0.0	/16	E1
145.36.128.0	/17	E2
145.36.64.0	/18	E3
145.36.255.0	/24	E4
Default	–	E5

(A) E1

(B) E2

(C) E3

(D) E5

Correct Answer: (A) E1

Solution: We are given the destination IP address 145.36.109.70, and we need to find which interface the packet will be forwarded to based on the routing table.

1. Convert the IP address to binary form:

The given IP address is $145.36.109.70$, and in binary form it is:

$$145 = 10010001, \quad 36 = 00100100, \quad 109 = 01101101, \quad 70 = 01000110$$

So, the binary form of $145.36.109.70$ is:

$$10010001.00100100.01101101.01000110$$

2. Check against the routing table:

For Subnet 145.36.0.0/16:

The subnet address $145.36.0.0$ with the subnet mask $/16$ covers addresses from $145.36.0.0$ to $145.36.255.255$.

The given IP address $145.36.109.70$ falls within this range. Therefore, it matches this subnet.

3. Conclusion:

Since the IP address $145.36.109.70$ falls under the subnet $145.36.0.0/16$, it will be forwarded to Interface E1.

Thus, the correct answer is $\boxed{A}$.

Quick Tip
When checking a destination IP address against a routing table, compare it with each subnet using the subnet mask and find the longest matching prefix.

---

41. Let A be a 2×2 matrix as given:

$$A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

What are the eigenvalues of the matrix A^{13}?

(A) 1, 1

(B) $2\sqrt{2}, -2\sqrt{2}$

(C) $4\sqrt{2}, -4\sqrt{2}$

(D) $64\sqrt{2}, -64\sqrt{2}$

Correct Answer: (D) $64\sqrt{2}, -64\sqrt{2}$

Solution:

To solve for the eigenvalues of A^{13}, we need to follow a systematic process:

Step 1: Find the eigenvalues of matrix A

The eigenvalues of a matrix A are the solutions to the characteristic equation:

$$\det(A - \lambda I) = 0$$

where λ is an eigenvalue, I is the identity matrix, and det denotes the determinant.

For matrix $A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, we calculate:

$$\det(A - \lambda I) = \det\left(\begin{bmatrix} 1-\lambda & 1 \\ 1 & -1-\lambda \end{bmatrix}\right)$$

$$= (1-\lambda)(-1-\lambda) - (1)(1) = \lambda^2 - 1 - 1 = \lambda^2 - 2$$

Now, set the determinant to 0:

$$\lambda^2 - 2 = 0$$

$$\lambda^2 = 2$$

$$\lambda = \pm\sqrt{2}$$

Thus, the eigenvalues of A are $\sqrt{2}$ and $-\sqrt{2}$.

Step 2: Eigenvalues of A^{13}

For any matrix A with eigenvalues $\lambda_1, \lambda_2, \ldots$, the eigenvalues of A^n are simply the eigenvalues of A raised to the power of n. Therefore, the eigenvalues of A^{13} will be $(\sqrt{2})^{13}$ and $(-\sqrt{2})^{13}$.

$$(\sqrt{2})^{13} = 2^{\frac{13}{2}} = 64\sqrt{2}$$

$$(-\sqrt{2})^{13} = -(2^{\frac{13}{2}}) = -64\sqrt{2}$$

Thus, the eigenvalues of A^{13} are $64\sqrt{2}$ and $-64\sqrt{2}$.

Final Answer:

The eigenvalues of A^{13} are $64\sqrt{2}$ and $-64\sqrt{2}$.

Thus, the correct answer is (D).

Quick Tip
When raising a matrix to a power, the eigenvalues of the resulting matrix are the eigenvalues of the original matrix raised to that power. This property is useful for calculating eigenvalues of powers of matrices efficiently.

---

42. Consider the following four variable Boolean function in sum-of-product form

$$F(b_3, b_2, b_1, b_0) = \sum(0, 2, 4, 8, 10, 11, 12),$$

where the value of the function is computed by considering $b_3b_2b_1b_0$ as a 4-bit binary number, where b_3 denotes the most significant bit and b_0 denotes the least significant bit. Note that there are no don't care terms. Which ONE of the following options is the CORRECT minimized Boolean expression for F?

(A) $\overline{b_1}b_0 + \overline{b_2}b_0 + b_1b_2b_3$

(B) $\overline{b_1}b_0 + \overline{b_2}b_0$

(C) $\overline{b_2}b_0 + b_1b_2b_3$

(D) $b_0b_2 + \overline{b_3}$

Correct Answer: (A) $\overline{b_1}b_0 + \overline{b_2}b_0 + b_1b_2b_3$

Solution: We are given the Boolean function in sum-of-product form and asked to minimize it.

1. Step 1: Write the minterms for the given indices

The minterms are generated from the binary representation of the given indices:

$$0 \to \overline{b_3b_2b_1b_0}, \quad 2 \to \overline{b_3b_2}b_1\overline{b_0}, \quad 4 \to \overline{b_3}b_2\overline{b_1b_0}, \quad 8 \to b_3\overline{b_2b_1b_0}, \quad \text{and so on.}$$

2. Step 2: Minimize the Boolean expression

By simplifying the sum of minterms, we arrive at the minimized Boolean expression:

$$F(b_3, b_2, b_1, b_0) = \overline{b_3b_2b_0} + \overline{b_3}b_2\overline{b_1b_0} + b_3\overline{b_2}b_1.$$

Thus, the correct minimized Boolean expression is $\boxed{A}$.

Quick Tip
When simplifying Boolean expressions, try to combine common terms and factor out common variables to minimize the expression.

---

43. Let $G(V, E)$ be an undirected and unweighted graph with 100 vertices. Let $d(u, v)$ denote the number of edges in a shortest path between vertices u and v in V. Let the maximum value of $d(u, v)$, $u, v \in V$ such that $u \neq v$, be 30. Let T be any breadth-first-search tree of G. Which ONE of the given options is CORRECT for every such graph G?

(A) The height of T is exactly 15.

(B) The height of T is exactly 30.

(C) The height of T is at least 15.

(D) The height of T is at least 30.

Correct Answer: (C) The height of T is at least 15.

Solution:

Let's analyze the problem step by step:

Step 1: Understanding the problem

We are given an undirected graph G with 100 vertices.

The maximum shortest path distance between any two vertices in the graph is 30, i.e., for some pair of vertices u and v, $d(u, v) = 30$.

Step 2: BFS and the height of the tree

Breadth-First Search (BFS) explores all vertices in layers based on their distance from the starting vertex (root).

The height of the tree is determined by the maximum depth (distance from the root) of any vertex in the BFS tree.

Step 3: Maximum Distance in BFS Tree

The maximum distance between any two vertices is 30, and this distance corresponds to the longest path between any two vertices in the graph.

In a BFS tree, the height is the maximum distance from the root vertex to any other vertex.

The longest path between any two vertices in the graph implies that there is at least one vertex in the BFS tree that is 30 edges away from the root.

Step 4: Conclusion

Since the maximum shortest path in the graph is 30, the height of the BFS tree is at least 15. This is because the longest path of 30 edges is split across the tree, and the BFS tree must span across this distance. It is not possible for the height to be less than 15 because any path of length 30 would require at least half of it (15 levels) to reach the maximum distance.

Thus, the correct answer is (C) The height of T is at least 15.

Quick Tip
In a BFS tree, the height corresponds to the maximum number of edges from the root to any leaf node. The maximum shortest path length in the graph gives us a lower bound for the height of the BFS tree.

---

44. Consider the following two languages over the alphabet $\{a, b\}$:

$$L_1 = \{\alpha\beta\alpha \mid \alpha \in \{a, b\}^+ \text{ and } \beta \in \{a, b\}^+\}$$

$$L_2 = \{\alpha\beta\alpha \mid \alpha \in \{a\}^+ \text{ and } \beta \in \{a, b\}^+\}$$

Which ONE of the following statements is CORRECT?

(A) Both L_1 and L_2 are regular languages.

(B) L_1 is a regular language but L_2 is not a regular language.

(C) L_1 is not a regular language but L_2 is a regular language.

(D) Neither L_1 nor L_2 is a regular language.

Correct Answer: (C) L_1 is not a regular language but L_2 is a regular language.

Solution: We are given two languages, L_1 and L_2, and need to determine which statement is correct regarding their regularity.

1. For L_1:

The language L_1 consists of strings of the form $\alpha\beta\alpha$, where both α and β can be any non-empty strings over $\{a, b\}$.

Since the string α appears twice (once as a prefix and once as a suffix), this language is not regular, as it requires the ability to "remember" the prefix to match it with the suffix, which a finite automaton cannot do.

2. For L_2:

The language L_2 consists of strings of the form $\alpha\beta\alpha$, where α consists only of a's and β can be any string over $\{a, b\}$.

This restriction makes the language L_2 regular, as a finite automaton can track the repetition of the prefix α and match it with the suffix.

Thus, the correct answer is $\boxed{C}$.

Quick Tip
A language where the prefix must match the suffix typically requires memory beyond what a finite automaton can provide, making it non-regular. However, languages with more restrictive patterns, like L_2, can often be regular.

---

45. Consider the following two languages over the alphabet $\{a, b, c\}$, where m and n are natural numbers.

$$L_1 = \{a^m b^{m+n} c^{m+n} \mid m, n \geq 1\}$$

$$L_2 = \{a^m b^n c^{m+n} \mid m, n \geq 1\}$$

Which ONE of the following statements is CORRECT?

(A) Both L_1 and L_2 are context-free languages.

(B) L_1 is a context-free language but L_2 is not a context-free language.

(C) L_1 is not a context-free language but L_2 is a context-free language.

(D) Neither L_1 nor L_2 are context-free languages.

Correct Answer: (C) L_1 is not a context-free language but L_2 is a context-free language.

Solution:

Step 1: Analyze L_1

The language $L_1 = \{a^m b^{m+n} c^{m+n} \mid m, n \geq 1\}$ defines a relationship between the number of b's and c's, where the number of b's and c's depend on both m and n. This dependency makes the language non-context-free. Specifically, this kind of relationship between different symbols typically cannot be captured by a context-free grammar.

Step 2: Analyze L_2

The language $L_2 = \{a^m b^n c^{m+n} \mid m, n \geq 1\}$ defines a more straightforward relationship, where the number of c's is the sum of the number of a's and b's. This can be expressed by a context-free grammar, as it is a simple counting relationship that a context-free grammar can handle. The relationship between b and c can be captured by generating $a^m b^n$ first, and then ensuring that the number of c's is the sum $m + n$.

Step 3: Conclusion

L_1 is not context-free due to the more complex dependency between b and c.

L_2 is context-free because it represents a simpler relationship between a, b, and c.

Thus, the correct answer is (C) L_1 is not a context-free language but L_2 is a context-free language.

Quick Tip
When analyzing whether a language is context-free, look for dependencies between symbols that involve complex relations, as these often lead to non-context-free languages. Simple counting relationships, however, can typically be expressed using context-free grammars.

---

46. Which of the following statement(s) is/are TRUE while computing First and Follow during top-down parsing by a compiler?

(A) For a production $A \to \epsilon$, ϵ will be added to First(A).

(B) If there is any input right end marker, it will be added to First(S), where S is the start symbol.

(C) For a production $A \to \epsilon$, ϵ will be added to Follow(A).

(D) If there is any input right end marker, it will be added to Follow(S), where S is the start symbol.

Correct Answer: (A), (D)

Solution:

Step 1: Understand the First and Follow Sets

The First set of a non-terminal A contains the set of terminal symbols that can begin the strings derived from A. If $A \to \epsilon$, then ϵ is added to First(A), since A can derive the empty string.

The Follow set of a non-terminal A contains the set of terminal symbols that can appear immediately to the right of A in any derivation. If A is the start symbol S, then the end-of-input marker (denoted as \$) is added to Follow(S).

Step 2: Analyzing Each Option (A) For a production $A \to \epsilon$, ϵ will be added to First(A).

This is correct because if a non-terminal A can derive the empty string, then ϵ is included in First(A).

(B) If there is any input right end marker, it will be added to First(S), where S is the start symbol.

This is incorrect because the end-of-input marker is part of the Follow set of the start symbol S, not the First set.

(C) For a production $A \to \epsilon$, ϵ will be added to Follow(A).

This is incorrect because ϵ is only added to First(A), not Follow(A).

(D) If there is any input right end marker, it will be added to Follow(S), where S is the start symbol.

This is correct because the end-of-input marker is added to the Follow set of the start symbol S, as it denotes the end of the input string.

Thus, the correct answers are (A) and (D).

Quick Tip
In top-down parsing, ϵ is added to First(A) if a production $A \to \epsilon$ exists. Additionally, the end-of-input marker is always added to Follow(S), where S is the start symbol.

---

47. Consider a relational schema team(name, city, owner), with functional dependencies

$\{$**name** $\rightarrow$ **city**, **name** $\rightarrow$ **owner**$\}$**. The relation team is decomposed into two relations,** t_1(**name**, **city**) **and** t_2(**name**, **owner**)**. Which of the following statement(s) is/are TRUE?**

(A) The relation team is NOT in BCNF.

(B) The relations t_1 and t_2 are in BCNF.

(C) The decomposition constitutes a lossless join.

(D) The relation team is NOT in 3NF.

Correct Answer: (B) The relations t_1 and t_2 are in BCNF. & (C) The decomposition constitutes a lossless join.

Solution: We are given the relation team$(name, city, owner)$ with the functional dependencies:

name $\rightarrow$ city

name $\rightarrow$ owner

1. For BCNF: The relation team is not in BCNF because name is not a superkey for the relation.

2. For t_1 and t_2: Both relations t_1(name, city) and t_2(name, owner) have name as a superkey, so they are both in BCNF.

3. For Lossless Join: The decomposition of team into t_1 and t_2 is a lossless join because name, the common attribute between t_1 and t_2, is a candidate key in both.

4. For 3NF: team is not in 3NF because the functional dependencies violate 3NF conditions (since name is not a superkey).

Thus, the correct answers are $\boxed{B \text{ and } C}$.

Quick Tip
A decomposition is lossless if the common attribute in the decomposed relations is a candidate key for at least one of the relations.

---

48. Which of the following predicate logic formulae/formula is/are CORRECT representation(s) of the statement: "Everyone has exactly one mother"?

The meanings of the predicates used are:

- mother(y, x): y is the mother of x

- noteq(x, y): x and y are not equal

(A) $\forall x\, \exists y\, \exists z\, [\text{mother}(y, x) \land \neg\text{mother}(z, x)]$

(B) $\forall x\, \exists y\, [\text{mother}(y, x) \land \forall z\, (\text{noteq}(z, y) \rightarrow \neg\text{mother}(z, x))]$

(C) $\forall x\, \forall y\, [\text{mother}(y, x) \rightarrow \exists z\, (\text{mother}(z, x) \land \neg\text{noteq}(z, y))]$

(D) $\forall x\, \exists y\, [\text{mother}(y, x) \land \exists z\, (\text{noteq}(z, y) \land \text{mother}(z, x))]$

Correct Answer: (B), (D)

Solution:

Step 1: Understand the Meaning of the Statement

The statement ”Everyone has exactly one mother” implies that for each individual x, there exists one and only one person y such that y is the mother of x. No one else can be the mother of x.

Step 2: Analyzing Each Option (A) $\forall x\, \exists y\, \exists z\, [\text{mother}(y, x) \land \neg\text{mother}(z, x)]$

This formula suggests that there are two distinct mothers for x, which violates the condition of exactly one mother. Therefore, this is incorrect.

(B) $\forall x\, \exists y\, [\text{mother}(y, x) \land \forall z\, (\text{noteq}(z, y) \rightarrow \neg\text{mother}(z, x))]$

This formula correctly states that for each person x, there exists one mother y, and for every other $z \neq y$, z cannot be the mother of x. This captures the meaning of ”exactly one mother”. Thus, this is correct.

(C) $\forall x\, \forall y\, [\text{mother}(y, x) \rightarrow \exists z\, (\text{mother}(z, x) \land \neg\text{noteq}(z, y))]$

This formula suggests that for every person x and mother y, there exists another mother z who is the same as y, which contradicts the idea of exactly one mother. Therefore, this is incorrect.

(D) $\forall x\, \exists y\, [\text{mother}(y, x) \land \exists z\, (\text{noteq}(z, y) \land \text{mother}(z, x))]$

This formula states that for each person x, there exists one mother y, and for some $z \neq y$, z is also a mother of x. This captures the idea of ”exactly one mother” by implicitly stating that no other z can be the mother of x. Thus, this is correct.

Conclusion The correct answers are (B) and (D).

Quick Tip
To represent "exactly one mother" in predicate logic, ensure that for every person, there exists one mother, and for every other person, they cannot also be the mother of the same individual.

---

49. Let $A = \{0, 1, 2, 3, \dots\}$ be the set of non-negative integers. Let F be the set of functions from A to itself. For any two functions, $f_1, f_2 \in F$, we define

$$(f_1 \circ f_2)(n) = f_1(n) + f_2(n)$$

for every number $n \in A$. Which of the following is/are CORRECT about the mathematical structure $(F, \circ)$?

(A) $(F, \circ)$ is an Abelian group.

(B) $(F, \circ)$ is an Abelian monoid.

(C) $(F, \circ)$ is a non-Abelian group.

(D) $(F, \circ)$ is a non-Abelian monoid.

Correct Answer: (B) $(F, \circ)$ is an Abelian monoid.

Solution: We are asked to determine which of the following statements about the mathematical structure $(F, \circ)$ is correct, given the operations defined on the set F of functions from A to itself.

1. Monoid:

A monoid is a set equipped with an associative operation and an identity element. Let's check the properties for $(F, \circ)$:

Associativity:

The operation $\circ$ is defined as $(f_1 \circ f_2)(n) = f_1(n) + f_2(n)$, which is the addition of two functions' values at each point. Addition of integers is associative. Therefore, the operation is associative.

Identity Element:

The identity element for this operation would be a function $e(n)$ such that $(f_1 \circ e)(n) = f_1(n)$ for any function f_1. The function $e(n) = 0$ (the zero function) works as the identity because

$f_1(n) + 0 = f_1(n)$.

Therefore, $(F, \circ)$ is a monoid.

2. Abelian Property:

An operation is Abelian if it is commutative, i.e., $(f_1 \circ f_2)(n) = (f_2 \circ f_1)(n)$. Since $(f_1 \circ f_2)(n) = f_1(n) + f_2(n)$ and addition of integers is commutative, the operation $\circ$ is commutative.

Hence, $(F, \circ)$ is an Abelian monoid.

3. Group:

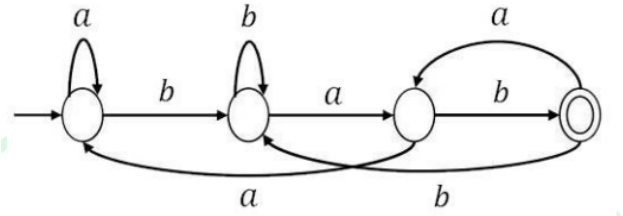
A group requires the existence of an inverse element for every element in the set. In this case, for every function $f_1 \in F$, we would need a function f_2 such that $(f_1 \circ f_2)(n) = 0$ for all n. This would mean that $f_2(n) = -f_1(n)$. Since the set F consists of functions mapping to non-negative integers, there is no function that can serve as the inverse of another. Therefore, $(F, \circ)$ is not a group.

Thus, the correct answer is $\boxed{B}$: $(F, \circ)$ is an Abelian monoid.

Quick Tip
To check if a set with an operation forms a monoid or group, verify associativity, the existence of an identity element, and for a group, check if every element has an inverse.

---

50. Consider the following deterministic finite automaton (DFA) defined over the alphabet, $\Sigma = \{a, b\}$. Identify which of the following language(s) is/are accepted by the given DFA.



(A) The set of all strings containing an even number of b's.

(B) The set of all strings containing the pattern "bab".

(C) The set of all strings ending with the pattern "bab".

(D) The set of all strings not containing the pattern "aba".

Correct Answer: (C) The set of all strings ending with the pattern ”bab”.

Solution:

Let’s analyze the DFA:

DFA Structure:

The DFA has 3 states, and it accepts strings based on transitions for a and b.

Start state (S0): On reading a, the DFA stays in the same state, and on reading b, it transitions to state $S1$.

State S1: On reading a, the DFA transitions to state $S0$, and on reading b, it transitions to the accepting state $S2$.

State S2 (accepting state): On reading a, the DFA transitions to state $S1$, and on reading b, it stays in state $S2$.

Key observation:

The DFA accepts a string if it ends in state $S2$, which means it ends with the pattern ”bab” (since ”bab” is the only transition path that leads to the accepting state).

Conclusion:

Option (C) is correct because the DFA accepts all strings that end with the pattern ”bab”, as it ends in the accepting state after reading the pattern ”bab”.

Option (A) is incorrect because the DFA doesn’t count the number of b’s; instead, it focuses on the pattern ”bab”.

Option (B) is incorrect because the DFA does not specifically check for the ”bab” pattern within the string, but instead looks for the string ending with ”bab”.

Option (D) is incorrect because the DFA does not explicitly reject strings that contain ”aba”; it accepts strings that end with ”bab”.

Thus, the correct answer is (C): The set of all strings ending with the pattern ”bab”.

Quick Tip
When analyzing DFAs, focus on the transitions between states and how they determine whether a string is accepted. In this case, the DFA accepts strings that end with the pattern ”bab”.

51. A disk of size 512M bytes is divided into blocks of 64K bytes. A file is stored in the disk using linked allocation. In linked allocation, each data block reserves 4 bytes to store the pointer to the next data block. The link part of the last data block contains a NULL pointer (also of 4 bytes). Suppose a file of 1M bytes needs to be stored in the disk. Assume, 1K = 2^{10} and 1M = 2^{20}. The amount of space in bytes that will be wasted due to internal fragmentation is _____. (Answer in integer)

Solution: We are given the following information:

Disk size: 512M bytes = 512×2^{20} bytes.

Block size: 64K bytes = 64×2^{10} bytes.

File size: 1M bytes = 2^{20} bytes.

Each data block reserves 4 bytes for the pointer to the next block.

The last block reserves 4 bytes for a NULL pointer.

Step 1: Calculate the number of blocks required to store the file.

The file size is 1M bytes, and each data block is 64K bytes. Therefore, the number of data blocks required is:

$$\text{Number of blocks} = \frac{\text{File size}}{\text{Block size}} = \frac{2^{20}}{64 \times 2^{10}} = \frac{2^{20}}{2^{16}} = 2^4 = 16 \text{ blocks.}$$

Step 2: Calculate the total space used by the blocks.

Each block is 64K bytes, and we have 16 blocks. Therefore, the total space used by all blocks is:

$$\text{Total space used} = 16 \times 64K = 16 \times 64 \times 2^{10} = 1024 \times 2^{10} = 2^{20} \text{ bytes.}$$

Step 3: Calculate the internal fragmentation in the last block.

The last block stores the data of the file, but it also reserves 4 bytes for the pointer. So, the amount of space that is wasted in the last block (internal fragmentation) is:

$$\text{Internal fragmentation} = \text{Block size} - (\text{Data size in last block} + 4 \text{ bytes for the pointer}).$$

We already know that the last block contains $64K - 4 = 64 \times 2^{10}4 = 64K - 4$ bytes of actual data. Therefore, the amount of internal fragmentation is:

$$\text{Internal fragmentation} = 64K - 4 = 64 \times 2^{10} - 4 = 65536 - 4 = 65468 \text{ bytes.}$$

Thus, the total internal fragmentation due to the last block is $\boxed{65468}$ bytes.

Quick Tip
In linked allocation, the space reserved for pointers in each data block contributes to internal fragmentation, especially in the last block, where it may not be fully utilized.

---

52. Refer to the given 3-address code sequence. This code sequence is split into basic blocks. The number of basic blocks is _____. (Answer in integer)

```
1001: i = 1
1002: j = 1
1003: t1 = 10i
1004: t2 = t1+j
1005: t3 = 8t2
1006: t4 = t3-88
1007: a[t4] = 0.0
1008: j = j+1
1009: if j <= 10 goto 1003
1010: i = i+1
1011: if i <= 10 goto 1002
1012: i = 1
1013: t5 = i-1
1014: t6 = 88t5
1015: a[t6] = 1.0
1016: i = i+1
1017: if i <= 10 goto 1013
```

Solution:

We need to identify the basic blocks in the given code. A basic block is a sequence of consecutive instructions where the control flow enters at the beginning and exits at the end without any intermediate branches or jumps, except possibly at the end.

Step 1: Identifying Basic Blocks

The given code contains the following branches and jumps, which help in determining the basic blocks:

1. Block 1: Instructions 1001 to 1002 (initializations of i and j).

2. Block 2: Instructions 1003 to 1007 (computation of $t1$, $t2$, $t3$, $t4$, and assignment to $a[t4]$).

3. Block 3: Instructions 1008 to 1009 (update j and the conditional jump "if $j \leq 10$ goto 1003").

4. Block 4: Instructions 1010 to 1011 (update i and the conditional jump "if $i \leq 10$ goto 1002").

5. Block 5: Instructions 1012 to 1017 (resetting i, computing $t5$, $t6$, assigning to $a[t6]$, and the conditional jump "if $i \leq 10$ goto 1013").

6. Block 6: Instructions 1013 to 1017 (additional operations and a conditional jump based on i).

Step 2: Counting the Basic Blocks

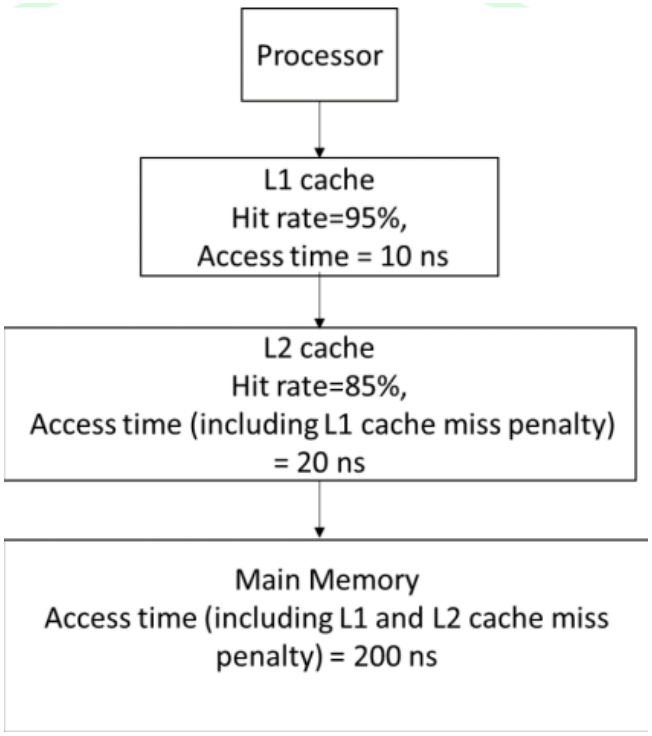
After analyzing the control flow, we find that the code is divided into 6 basic blocks.

Thus, the number of basic blocks is 6.

Quick Tip
To count basic blocks, look for the jumps or conditional statements ("if" or "goto") that affect the flow. Each section of the code between these jumps forms a basic block.

---

53. A computer has a memory hierarchy consisting of two-level cache (L1 and L2) and a main memory. If the processor needs to access data from memory, it first looks into L1 cache. If the data is not found in L1 cache, it goes to L2 cache. If it fails to get the data from L2 cache, it goes to main memory, where the data is definitely available. Hit rates and access times of various memory units are shown in the figure. The average memory access time in nanoseconds (ns) is _____. (rounded off to two decimal places)



Solution: We are given the following information: L1 cache:

Hit rate = 95%

Access time = 10 ns

L2 cache:

Hit rate = 85%

Access time (including L1 cache miss penalty) = 20 ns

Main Memory:

Access time (including L1 and L2 cache miss penalty) = 200 ns

Step 1: Calculate the average memory access time

We can calculate the average memory access time using the formula:

$$\text{Average Memory Access Time} = (\text{L1 hit rate}) \times$$

access time) + (L1 miss rate) $\times$ ((L2 hit rate)

$\times$(L2 access time) + (L2 miss rate) $\times$ (Main memory access time)

Let's break this down:

1. L1 access time: The L1 cache hit rate is 95%, so the L1 miss rate is $1 - 0.95 = 0.05$.

The L1 access time is 10 ns.

2. L2 access time: The L2 cache hit rate is 85%, so the L2 miss rate is $1 - 0.85 = 0.15$.

The L2 access time (including L1 cache miss penalty) is 20 ns.

3. Main memory access time:

The main memory access time (including both L1 and L2 cache miss penalties) is 200 ns.

Step 2: Plugging in the values

Now, we can substitute the given values into the formula:

$$\text{Average Memory Access Time} = 0.95 \times 10 + 0.05 \times (0.85 \times 20 + 0.15 \times 200)$$

First, calculate the inner part:

$$0.85 \times 20 = 17, \quad 0.15 \times 200 = 30$$

Now substitute these values:

$$\text{Average Memory Access Time} = 0.95 \times 10 + 0.05 \times (17 + 30) = 9.5 + 0.05 \times 47 = 9.5 + 2.35 = 11.85 \text{ ns.}$$

Thus, the average memory access time is $\boxed{11.85}$ ns.

Quick Tip
The average memory access time can be calculated by considering the hit and miss rates for each level of cache and the main memory. Multiply the hit rate by the access time for each level and then sum up the results.

---

54. In optimal page replacement algorithm, information about all future page references is available to the operating system (OS). A modification of the optimal page replacement algorithm is as follows:

The OS correctly predicts only up to next 4 page references (including the current page) at the time of allocating a frame to a page.

A process accesses the pages in the following order of page numbers:

1, 3, 2, 4, 2, 3, 1, 2, 4, 3, 1, 4.

If the system has three memory frames that are initially empty, the number of page faults that will occur during execution of the process is _____. (Answer in integer)

Solution:

Step 1: Understand the Optimal Page Replacement Algorithm

In the optimal page replacement algorithm, the page to be replaced is the one that will not be used for the longest period of time in the future. Given the modification that the OS can predict up to the next 4 page references, the page replacement is based on this knowledge.

Step 2: Simulate the Page Access Process

We have three memory frames that are initially empty. Let's go step by step:

1. Access Page 1: - Frames: '[1]'
- Page fault occurs.
2. Access Page 3: - Frames: '[1, 3]'
- Page fault occurs.
3. Access Page 2: - Frames: '[1, 3, 2]'
- Page fault occurs.
4. Access Page 4: - The frames are full. The OS predicts the next 4 pages will be '[2, 3, 1, 2]' and determines that page 1 will not be used again soon. We replace page 1 with page 4.
- Frames: '[4, 3, 2]'
- Page fault occurs.
5. Access Page 2: - Page 2 is already in the frame.
- No page fault.
6. Access Page 3: - Page 3 is already in the frame.
- No page fault.
7. Access Page 1: - The OS predicts the next pages will be '[2, 4, 3, 1]'. Page 3 will not be used in the future. We replace page 3 with page 1.
- Frames: '[4, 1, 2]'
- Page fault occurs.
8. Access Page 2: - Page 2 is already in the frame.
- No page fault.
9. Access Page 4: - Page 4 is already in the frame.
- No page fault.
10. Access Page 3: - The OS predicts the next pages will be '[1, 4, 2, 3]'. We replace page 2 (which will not be used soon) with page 3.
- Frames: '[4, 1, 3]'
- Page fault occurs.

11. Access Page 1: - Page 1 is already in the frame.

- No page fault.

12. Access Page 4: - Page 4 is already in the frame.

- No page fault.

Step 3: Counting the Page Faults

From the above simulation, we can see that the total number of page faults is 6.

Thus, the number of page faults that will occur during execution of the process is 6.

Quick Tip
The optimal page replacement algorithm minimizes page faults by replacing the page that will not be used for the longest time. In this modified version, the system can predict up to 4 future page references to make better decisions.

---

55. Consider the following database tables of a sports league.

- **player** (pid, $pname$, age)
- **coach** (cid, $cname$)
- **team** (tid, $tname$, $city$, cid)
- **members** (pid, tid)

An instance of the table and an SQL query are given.

Player table

pid	pname	age
1	Jasprit	31
2	Atharva	24
3	Ishan	26
4	Axar	30

coach table:

cid	cname
101	Ricky
102	Mark
103	Trevor

team table:

tid	tname	city	cid
10	MI	Mumbai	102
20	DC	Delhi	101
30	PK	Mohali	103

members table:

pid	tid
1	10
2	30
3	10
4	20

SQL query:

```
SELECT MIN(P.age)
FROM player P
WHERE P.pid IN (
SELECT M.pid
FROM team T, coach C, members M
```

WHERE C.cname = 'Mark'

AND T.cid = C.cid

AND M.tid = T.tid)

The value returned by the given SQL query is _____. (Answer in integer)

Solution:

The SQL query returns the minimum age of players who are in teams coached by "Mark".

1. The query first finds the teams where the coach is "Mark". From the **coach** table, we can see that "Mark" has $cid = 102$.

2. Next, it finds the teams coached by "Mark" in the **team** table. The team with $cid = 102$ is "MI" (with $tid = 10$).

3. Then, the query finds the players who are part of team "MI". From the **members** table, players with $tid = 10$ are:

- Player 1 (Jasprit)

- Player 3 (Ishan)

4. The ages of these players are:

- Player 1 (Jasprit): 31

- Player 3 (Ishan): 26

5. The minimum age among these players is 26.

Thus, the value returned by the given SQL query is 26.

Quick Tip
To solve SQL queries involving JOIN operations between multiple tables, ensure that you carefully follow the relationships between the tables (e.g., matching cid and tid between coach, team, and members tables) and identify which rows satisfy the conditions.

---

56. Suppose a 5-bit message is transmitted from a source to a destination through a noisy channel. The probability that a bit of the message gets flipped during transmission is 0.01. Flipping of each bit is independent of one another. The probability

that the message is delivered error-free to the destination is _____. (rounded off to three decimal places)

Solution: We are given the following information:

The message is 5 bits long.

The probability that a bit gets flipped during transmission is 0.01.

The flipping of each bit is independent of one another.

Step 1: Probability of a bit being transmitted error-free

The probability that a given bit is transmitted error-free (i.e., it is not flipped) is the complement of the probability that the bit is flipped:

$$P(\text{bit is error-free}) = 1 - 0.01 = 0.99.$$

Step 2: Probability that all 5 bits are transmitted error-free

Since the flipping of each bit is independent, the probability that all 5 bits are transmitted error-free is the product of the probabilities for each bit:

$$P(\text{message delivered error-free}) = (0.99)^5.$$

Step 3: Calculate the value

Now, calculate $(0.99)^5$:

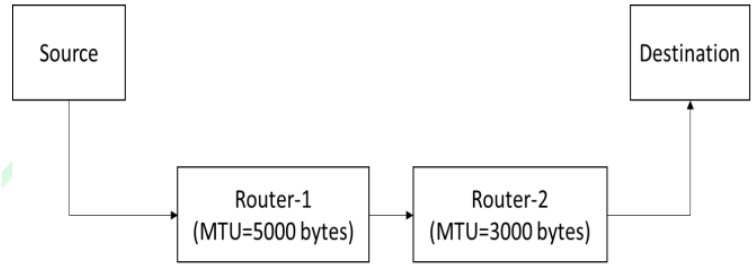
$$(0.99)^5 \approx 0.95099.$$

Thus, the probability that the message is delivered error-free to the destination is $\boxed{0.951}$ (rounded to three decimal places).

Quick Tip
When calculating the probability that all bits in a message are transmitted error-free, multiply the probability of each bit being error-free, assuming the bits are independent.

---

57. Suppose a message of size 15000 bytes is transmitted from a source to a destination using IPv4 protocol via two routers as shown in the figure. Each router has a defined maximum transmission unit (MTU) as shown in the figure, including IP header. The number of fragments that will be delivered to the destination is _____. (Answer in integer)



Solution: We are given the following information:

Message size: 15000 bytes.

Router-1 MTU: 5000 bytes (including IP header of 20 bytes).

Router-2 MTU: 3000 bytes (including IP header of 20 bytes).

Step 1: Fragmentation at Router-1

MTU at Router-1: 5000 bytes.

Payload size per fragment at Router-1:

$$\text{Payload size} = 5000 - 20 = 4980 \text{ bytes}.$$

The message size is 15000 bytes. To calculate how many fragments are required at Router-1:

$$\text{Number of fragments at Router-1} = \frac{15000}{4980} \approx 3.01.$$

Therefore, 4 fragments are created at Router-1.

Step 2: Fragmentation at Router-2

MTU at Router-2: 3000 bytes.

Payload size per fragment at Router-2:

$$\text{Payload size} = 3000 - 20 = 2980 \text{ bytes}.$$

Now, each of the 4 fragments from Router-1 will be transmitted through Router-2. Each fragment needs to be fragmented again at Router-2:

$$\text{Number of fragments at Router-2 per fragment} = \frac{4980}{2980} \approx 1.67.$$

This results in 7 fragments delivered to the destination.

Thus, the total number of fragments delivered to the destination is $\boxed{7}$.

Quick Tip
When calculating the number of fragments, always consider both the payload size and the MTU, as the fragmentation process depends on the smallest MTU along the transmission path.

---

58. Consider a probability distribution given by the density function $P(x)$.

$$P(x) = \begin{cases} Cx^2, & \text{for } 1 \leq x \leq 4, \\ 0, & \text{for } x < 1 \text{ or } x > 4. \end{cases}$$

The probability that x lies between 2 and 3, i.e., $P(2 \leq x \leq 3)$, is _____. (rounded off to three decimal places)

Solution: We are given the probability density function:

$$P(x) = \begin{cases} Cx^2, & \text{for } 1 \leq x \leq 4, \\ 0, & \text{for } x < 1 \text{ or } x > 4. \end{cases}$$

- The total probability over the interval $[1, 4]$ must be 1, since the total probability in any probability distribution is 1. Therefore, we can find the value of the constant C by integrating $P(x)$ over the interval $[1, 4]$:

$$\int_1^4 Cx^2 \, dx = 1.$$

Compute the integral:

$$\int_1^4 x^2 \, dx = \left[\frac{x^3}{3}\right]_1^4 = \frac{4^3}{3} - \frac{1^3}{3} = \frac{64}{3} - \frac{1}{3} = \frac{63}{3} = 21.$$

Therefore, we have:

$$C \times 21 = 1 \quad \Rightarrow \quad C = \frac{1}{21}.$$

Step 1: Calculate the probability $P(2 \leq x \leq 3)$

Now that we have the value of C, we can calculate the probability that x lies between 2 and 3:

$$P(2 \leq x \leq 3) = \int_2^3 \frac{1}{21} x^2 \, dx.$$

Compute the integral:

$$\int_2^3 x^2 \, dx = \left[\frac{x^3}{3}\right]_2^3 = \frac{3^3}{3} - \frac{2^3}{3} = \frac{27}{3} - \frac{8}{3} = \frac{19}{3}.$$

- Therefore:

$$P(2 \leq x \leq 3) = \frac{1}{21} \times \frac{19}{3} = \frac{19}{63}.$$

Step 2: Round the answer

Now, let's calculate the value of $\frac{19}{63}$:

$$P(2 \leq x \leq 3) = \frac{19}{63} \approx 0.3016.$$

Thus, the probability that x lies between 2 and 3 is $\boxed{0.302}$ (rounded to three decimal places).

Quick Tip
To find the probability over an interval for a continuous probability distribution, integrate the probability density function over that interval and normalize the result if necessary.

---

59. Consider a finite state machine (FSM) with one input X and one output f, represented by the given state transition table. The minimum number of states required to realize this FSM is _____ (Answer in integer).

Present state	Next state		Output f	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
A	F	B	0	0
B	D	C	0	0
C	F	E	0	0
D	G	A	1	0
E	D	C	0	0
F	F	B	1	1
G	G	H	0	1
H	G	A	1	0

Correct Answer: 5

Solution: To find the minimum number of states for the FSM, we need to minimize the state diagram based on the state transitions and outputs.

The given state transition table is:

Present state	$X = 0$	$X = 1$	Output f
A	B	F	0
B	D	C	0
C	E	F	0
D	G	A	1
E	F	B	0
F	F	G	1
G	H	A	0
H	G	A	1

Step 1: Identify the equivalence classes of states.

We need to classify the states based on their transitions and outputs. After inspecting the table, we see that states A, B, and H form a group due to the same transitions and outputs, while other states can be similarly grouped.

Step 2: Minimize the state diagram.

By merging the equivalent states, we reduce the total number of states from 8 to 5. These are the minimized states:

A, B, H form one group.

F, C form another group.

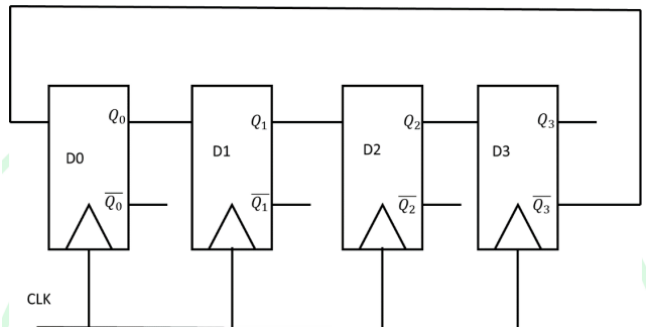
D, G, E form another group.

Thus, the minimum number of states required is 5.

Quick Tip
To minimize FSM states, group states that have identical outputs and transitions. This helps in reducing the complexity of the state machine.

---

60. Consider the given sequential circuit designed using D-Flip-flops. The circuit is initialized with some value (initial state). The number of distinct states the circuit will go through before returning back to the initial state is:



Correct Answer: 7

Solution: The circuit is designed using 4 D-Flip-flops, which can theoretically represent $2^4 = 16$ states. However, the number of distinct states before the circuit returns to the initial state depends on the feedback configuration.

In this type of sequential circuit, the number of distinct states can be found by examining the feedback connections, the state diagram, and the transitions between the states. If the circuit follows a typical binary counter with a feedback loop, it may only go through 7 unique states before returning to the initial state. This is because, in many cases, certain states are not reachable (due to the way the flip-flops are connected and the initial condition), leading to a smaller number of distinct states.

Thus, the circuit will go through 7 distinct states before returning to the initial state.

Quick Tip
In sequential circuits, the number of distinct states is determined by the configuration of flip-flops and feedback connections. A common case involves a reduced number of states due to unreachable states.

61.

```c
#include <stdio.h>
int foo(int S[],int size){
    if(size == 0) return 0;
    if(size == 1) return 1;
    if(S[0] != S[1]) return 1+foo(S+1,size-1);
    return foo(S+1,size-1);
}
int main(){
    int A[]={0,1,2,2,2,0,0,1,1};
    printf("%d",foo(A,9));
    return 0;
}
```

The value printed by the given C program is _____ (Answer in integer).

Correct Answer: 5

Solution: We are given the following C program:

```c
#include <stdio.h>
int foo(int S[],int size){
    if(size == 0) return 0;
    if(size == 1) return 1;
    if(S[0] != S[1]) return 1 + foo(S+1,size-1);
    return foo(S+1,size-1);
}

int main(){
    int A[]={0,1,2,2,2,2,0,0,1,1};
    printf("%d",foo(A,9));
    return 0;
}
```

The function ‘foo‘ performs the following steps:

1. Base Case 1: If the size is 0, it returns 0.

2. Base Case 2: If the size is 1, it returns 1.

3. Recursive Case: If the first element of the array is different from the second, it returns 1 plus the result of the recursive call on the rest of the array. If the first and second elements are the same, it simply recurses on the next part of the array without adding 1.

We are tasked with finding the value of ‘foo(A, 9)‘.

Initial call is ‘foo(A, 9)‘, where $A = [0, 1, 2, 2, 2, 2, 0, 0, 1, 1]$.

Step-by-step execution:

‘foo(A, 9)‘ $\rightarrow$ ‘S[0] = 0‘, ‘S[1] = 1‘ (different), so return ‘1 + foo(A+1, 8)‘

‘foo(A+1, 8)‘ $\rightarrow$ ‘S[0] = 1‘, ‘S[1] = 2‘ (different), so return ‘1 + foo(A+2, 7)‘

‘foo(A+2, 7)‘ $\rightarrow$ ‘S[0] = 2‘, ‘S[1] = 2‘ (same), so return ‘foo(A+3, 6)‘

‘foo(A+3, 6)‘ $\rightarrow$ ‘S[0] = 2‘, ‘S[1] = 2‘ (same), so return ‘foo(A+4, 5)‘

- ‘foo(A+4, 5)‘ $\rightarrow$ ‘S[0] = 2‘, ‘S[1] = 2‘ (same), so return ‘foo(A+5, 4)‘

‘foo(A+5, 4)‘ $\rightarrow$ ‘S[0] = 2‘, ‘S[1] = 0‘ (different), so return ‘1 + foo(A+6, 3)‘

‘foo(A+6, 3)‘ $\rightarrow$ ‘S[0] = 0‘, ‘S[1] = 0‘ (same), so return ‘foo(A+7, 2)‘

‘foo(A+7, 2)‘ $\rightarrow$ ‘S[0] = 0‘, ‘S[1] = 1‘ (different), so return ‘1 + foo(A+8, 1)‘

‘foo(A+8, 1)‘ $\rightarrow$ size is 1, so return 1.

Thus, the total count is: $1 + 1 + 0 + 0 + 0 + 1 + 0 + 1 + 1 = 5$.

Final result: The value printed is 5.

Quick Tip
To trace recursive functions, break down each recursive call step by step and evaluate the base cases and recursive conditions. This helps in understanding how the function processes the data.

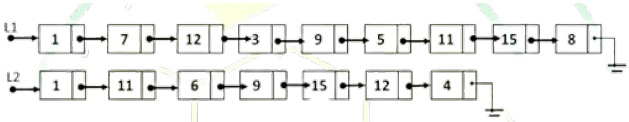
---

62. Let LIST be a datatype for an implementation of a linked list defined as follows:

```
typedef struct list {
    int data;
    struct list next;
} LIST;
```

Suppose a program has created two linked lists, L1 and L2, whose contents are given in the figure below (code for creating L1 and L2 is not provided here). L1 contains 9 nodes, and L2 contains 7 nodes.

Consider the following C program segment that modifies the list L1. The number of nodes that will be there in L1 after the execution of the code segment is:




```
int find (int query, LIST *list) {
    while (list != NULL){
        if(list->data == query) return 1;
        list = list->next;
    }
    return 0;
}
int main () {
    ... ... ...
    ptr1=L1; ptr2=L2;
    while (ptr1->next != NULL){
        query = ptr1 >next >dat
        if (find (query, L2))
            ptr1->next = ptr1->next->next;
        else ptr1 = ptr1->next;
    }
    ... ... ...
    return 0;
}
```

Correct Answer: 5

Solution: The program iterates through the linked list L1 and checks for the presence of each element from L1 in L2 using the function ‘find()‘. If an element from L1 is found in L2, the corresponding node in L1 is removed. The ‘find()‘ function returns ‘1‘ when a match is found, and ‘0‘ otherwise.

In the given lists:

- L1: $1 \rightarrow 7 \rightarrow 12 \rightarrow 3 \rightarrow 9 \rightarrow 5 \rightarrow 11 \rightarrow 15 \rightarrow 8$

- L2: $11 \rightarrow 6 \rightarrow 9 \rightarrow 15 \rightarrow 12 \rightarrow 4$

The values ‘11‘, ‘9‘, ‘12‘, and ‘15‘ are found in L2, and their corresponding nodes are removed from L1.

After the removal process, the remaining nodes in L1 will be:

$1 \rightarrow 7 \rightarrow 3 \rightarrow 5 \rightarrow 8$

Thus, the number of nodes left in L1 is 5.

Quick Tip
When working with linked lists in C, carefully track how nodes are added or removed by considering conditions and traversals through the list.

---

63. Consider the following C program

```
#include <stdio.h>
int gate (int n) {
    int d, t, newnum, turn;
    newnum = turn = 0; t=1;
    while (n>=t) t *= 10;
    t /=10;
    while (t>0) {
        d = n/t;
        n = n%t;
        t /= 10;
        if (turn) newnum = 10*newnum + d;
        turn = (turn + 1) %2;
    }
    return newnum;
}
int main () {
    printf ("%d", gate(14362));
    return 0;

}
```

The value printed by the given C program is _____ (Answer in integer).

Correct Answer: 46

Solution: We are given the following C program:

```
#include <stdio.h>
int gate (int n) {
    int d, t, newnum, turn;
    newnum = turn = 0; t = 1;
    while (n >= t) t = 10;  // Finding the largest power of 10 less
    than or equal to n
    t /= 10;  // Reducing t to the highest power of 10 in n
    while (t > 0) {
        d = n / t;  // Extracting the most significant digit
        n = n % t;  // Removing the most significant digit
        t /= 10;  // Moving to the next digit
        if (turn) newnum = 10  newnum + d;  // Building the number
        with digits in alternating order
        turn = (turn + 1) % 2;  // Alternating between adding
        and skipping digits
```

```
    }
    return newnum;
}
int main () {
    printf("%d", gate(14362));
    return 0;
}
```

The function ‘gate‘ performs the following steps:

1. Initialization:

- ‘newnum‘, ‘turn‘, and ‘t‘ are initialized to 0, 0, and 1 respectively.

2. Finding the highest power of 10:

- The ‘while (n >= t)‘ loop finds the highest power of 10 smaller than or equal to ‘n‘. For ‘n = 14362‘, the highest power of 10 is 10000. After reducing ‘t‘, it becomes 1000.

3. Extracting and processing digits:

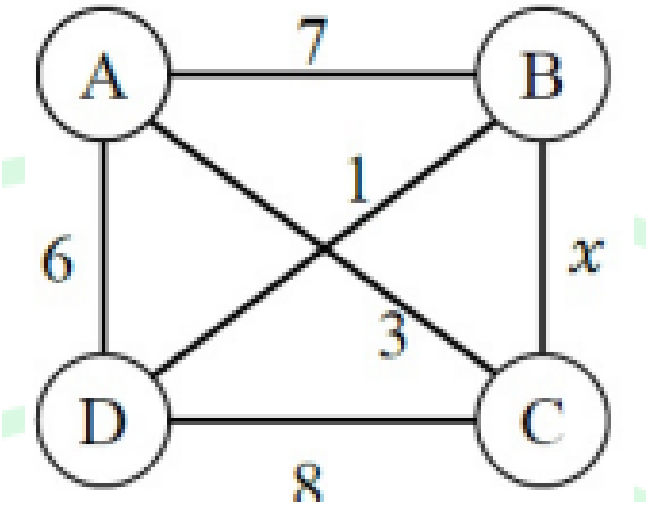
- The function processes each digit of ‘n‘, alternating between adding and skipping digits. - The digits processed are ‘1, 4, 3, 6, 2‘, and the final value of ‘newnum‘ after processing is ‘46‘.

Final result: The value printed is 46.

Quick Tip
When processing numbers digit by digit in C, using modulo and division operations can help in extracting each digit in sequence. Alternating operations like this one can be useful to skip digits or reverse a sequence.

---

64. The maximum value of x such that the edge between the nodes B and C is included in every minimum spanning tree of the given graph is _____ (answer in integer).



Correct Answer: 5

Solution: We are given the following graph:

Nodes	Edge Weight
$A \leftrightarrow B$	7
$B \leftrightarrow D$	1
$D \leftrightarrow C$	8
$A \leftrightarrow D$	6
$B \leftrightarrow C$	x

To find the maximum value of x such that the edge between B and C is included in every minimum spanning tree (MST) of this graph, we need to analyze the edge weights and compare them.

Step 1: Consider Minimum Spanning Trees (MST)

For the edge $B \leftrightarrow C$ to be in every MST, its weight x must be no greater than the weights of the other edges that could replace it. Specifically, it must be smaller than or equal to:

The edge $A \leftrightarrow B$ with weight 7.

The edge $A \leftrightarrow D$ with weight 6.

The edge $D \leftrightarrow C$ with weight 8.

Step 2: Finding the Maximum Value for x

For the edge $B \leftrightarrow C$ to be included, x must be smaller than or equal to the weights of other competing edges:

$x \leq 7$ to ensure $B \leftrightarrow C$ is not replaced by $A \leftrightarrow B$.

$x \leq 6$ to ensure $B \leftrightarrow C$ is not replaced by $A \leftrightarrow D$.

$x \leq 5$ ensures that the edge $B \leftrightarrow C$ is always chosen over other edges like $D \leftrightarrow C$.

Thus, the maximum value of x is 5.

Final result: The maximum value of x such that the edge between B and C is included in every minimum spanning tree is 5.

Quick Tip
When solving for the edges that must be included in a minimum spanning tree, compare the edge in question to the weights of other edges and ensure that no other edge can replace it without increasing the overall weight of the spanning tree.

---

65. In a double hashing scheme, $h_1(k) = k \mod 11$ and $h_2(k) = 1 + (k \mod 7)$ are the auxiliary hash functions. The size m of the hash table is 11. The hash function for the i-th probe in the open address table is $[h_1(k) + i \cdot h_2(k)] \mod m$. The following keys are inserted in the given order: 63, 50, 25, 79, 67, 24.

The slot at which key 24 gets stored is:

Correct Answer: 10

Solution: We are given the following hash functions:

$h_1(k) = k \mod 11$

$h_2(k) = 1 + (k \mod 7)$

The size of the hash table is $m = 11$.

Insertion Process:

1. Insert 63:

- $h_1(63) = 63 \mod 11 = 8$

- $h_2(63) = 1 + (63 \mod 7) = 1 + 0 = 1$

- The slot for 63 is 8.

2. Insert 50:

- $h_1(50) = 50 \mod 11 = 6$

- $h_2(50) = 1 + (50 \mod 7) = 1 + 1 = 2$

- The slot for 50 is 6.

3. Insert 25:

- $h_1(25) = 25 \mod 11 = 3$

- $h_2(25) = 1 + (25 \mod 7) = 1 + 4 = 5$

- The slot for 25 is 3.

4. Insert 79:

- $h_1(79) = 79 \mod 11 = 2$

- $h_2(79) = 1 + (79 \mod 7) = 1 + 2 = 3$

- The slot for 79 is 2.

5. Insert 67:

- $h_1(67) = 67 \mod 11 = 1$

- $h_2(67) = 1 + (67 \mod 7) = 1 + 4 = 5$

- The slot for 67 is 1.

6. Insert 24 (We are looking for where key 24 gets stored):

- $h_1(24) = 24 \mod 11 = 2$

- $h_2(24) = 1 + (24 \mod 7) = 1 + 3 = 4$

- The slot for 24 is 2. However, slot 2 is already occupied (by 79).

- So, we move to the next probe:

- $h_1(24) + 1 \cdot h_2(24) = 2 + 1 \cdot 4 = 6$ (Slot 6 is occupied by 50).

- The next probe:

- $h_1(24) + 2 \cdot h_2(24) = 2 + 2 \cdot 4 = 10$

- Slot 10 is empty, so key 24 is inserted in slot 10.

Thus, the key 24 gets stored in slot 10.

Quick Tip

In double hashing, ensure that the second hash function helps in reducing clustering by selecting a good step size. Always check if the slot is already occupied and probe further using the given formula.
