CUET-UG Computer Science Sample Paper - 5

Duration: 1 Hour **Maximum Marks: 250**

Instructions

- This paper contains a total of 50 Multiple Choice Questions.
- Each correct answer carries **+5 marks**.
- Each incorrect answer carries **-1 mark**.
- No negative marking for unattempted questions.

Q1. Consider the SQL query: `SELECT ROUND(157.285, -1) FROM DUAL;`. What will be the output?

(A) 157.29

(B) 160

(C) 157.3

(D) 150

Q2. Which SQL function is used to return the position of the first occurrence of a substring within a string?

(A) `LOCATE()`

(B) `SUBSTR()`

(C) `INSTR()`

(D) `POSITION()`

Q3. What will be the output of the following SQL command?

```
SELECT MOD(13, -3);
```

(A) 1

(B) -1

(C) 4

(D) -4

Q4. In SQL, the function used to extract a specific number of characters from the middle of a string is:

(A) `MID()`

(B) `SUBSTRING()`

(C) `SUBSTR()`

(D) All of the above

Q5. Identify the correct output for the SQL statement: `SELECT DAYNAME('2026-05-06');` (Note: May 6, 2026 is Wednesday)

(A) 6

(B) Wednesday

(C) May

(D) 05

Q6. The SQL function `TRUNCATE(543.21, -2)` results in:

(A) 543

(B) 500

(C) 540

(D) 0

Q7. Which of the following is NOT a valid aggregate function in SQL?

(A) `SUM()`

(B) `COUNT()`

(C) `LENGTH()`

(D) `AVG()`

Q8. The output of `SELECT RIGHT("CUET EXAM", 4);` is:

(A) `"CUET"`

(B) `"EXAM"`

(C) `"T EX"`

(D) `"AM"`

Q9. In a relational database, a candidate key that is not chosen as the primary key is known as a/an _____.

(A) Foreign Key

(B) Alternate Key

(C) Composite Key

(D) Secondary Key

Q10. The Relational Algebra operation that filters rows based on a specific condition is denoted by the symbol:

(A) Π (Pi)

(B) σ (Sigma)

(C) ρ (Rho)

(D) $\bowtie$ (Join)

Q11. Which property of a transaction ensures that all operations within the work unit are completed successfully, or none of them are?

(A) Atomicity

(B) Consistency

(C) Isolation

(D) Durability

Q12. If Table A has 5 rows and Table B has 3 rows, the number of rows in the Cartesian Product ($A \times B$) will be:

(A) 8

(B) 15

(C) 5^3

(D) 2

Q13. Which network topology requires a central controller or a hub to connect all nodes?

(A) Mesh

(B) Bus

(C) Star

(D) Ring

Q14. A MAC address is a _____ bit hardware address assigned to a Network Interface Card (NIC).

(A) 32

(B) 64

(C) 48

(D) 128

Q15. Which device operates at the Physical layer of the OSI model and simply regenerates a signal to extend the network length?

(A) Bridge

(B) Router

(C) Repeater

(D) Gateway

Q16. Convert the following infix expression to postfix: $A + (B \times C) - D$

(A) $ABC \times +D-$

(B) $AB + C \times D-$

(C) $ABC \times D - +$

(D) $A + BC \times D-$

Q17. A linear data structure where elements are added at one end and removed from the other is called a _____.

(A) Stack

(B) Queue

(C) Tree

(D) Linked List

Q18. Evaluate the following postfix expression: $8, 4, /, 5, \times, 2, +$

(A) 12

(B) 15

(C) 22

(D) 10

Q19. The `pop()` operation in a stack of size N results in an 'Underflow' condition when:

(A) $Top = N - 1$

(B) $Top = 0$

(C) $Top = -1$

(D) $Top = 1$

Q20. Which of the following is the correct circular queue condition for 'Queue Full' if F is Front and R is Rear (Size N)?

(A) $(R + 1)\%N == F$

(B) $R + 1 == F$

(C) $R == N - 1$

(D) $F == R + 1$

Q21. In Python, which method is used to add an element at the end of a list?

(A) `add()`

(B) `insert()`

(C) `append()`

(D) `extend()`

Q22. What is the time complexity of pushing an element onto a stack implemented using an array?

(A) $O(1)$

(B) $O(n)$

(C) $O(\log n)$

(D) $O(n^2)$

Q23. Which data structure is best suited for implementing ”Undo” functionality in a text editor?

(A) Queue

(B) Stack

(C) Deque

(D) Priority Queue

Q24. What is the prefix form of the expression $X/Y + Z$?

(A) $+/XYZ$

(B) $/+XYZ$

(C) $XY/Z+$

(D) $+XY/Z$

Q25. A Double Ended Queue (Deque) allows insertion and deletion at:

(A) Only the front

(B) Only the rear

(C) Both ends

(D) The middle

Q26. Which Python list method removes and returns the last element of the list?

(A) `remove()`

(B) `delete()`

(C) `pop()`

(D) `discard()`

Q27. The maximum number of comparisons required in a Binary Search for an array of 16 elements is:

(A) 16

(B) 8

(C) 4

(D) 5

Q28. Which sorting algorithm works by repeatedly picking the smallest element from the unsorted part and putting it at the beginning?

(A) Bubble Sort

(B) Insertion Sort

(C) Selection Sort

(D) Quick Sort

Q29. The best-case time complexity of Bubble Sort (optimized) is:

(A) $O(n)$

(B) $O(n^2)$

(C) $O(n \log n)$

(D) $O(1)$

Q30. In which sorting technique is an element compared with its predecessor and shifted if it is smaller?

(A) Selection Sort

(B) Insertion Sort

(C) Bubble Sort

(D) Merge Sort

Q31. Linear search is highly inefficient for large datasets because its average time complexity is:

(A) $O(\log n)$

(B) $O(n)$

(C) $O(1)$

(D) $O(n^2)$

Q32. Binary Search requires the input array to be _____ before starting the search process.

(A) Shuffled

(B) Sorted

(C) Reversed

(D) Empty

Q33. How many passes are required to sort an array of 10 elements using Bubble Sort?

(A) 10

(B) 11

(C) 9

(D) 45

Q34. Which sorting algorithm is often considered the most ”natural” way humans sort a deck of cards?

(A) Bubble Sort

(B) Selection Sort

(C) Insertion Sort

(D) Binary Search

Q35. In Selection Sort, after the first pass, which element is guaranteed to be in its correct final position?

(A) The largest

(B) The middle

(C) The smallest

(D) None

Q36. Which block in Python's exception handling is executed regardless of whether an exception occurred or not?

(A) `else`

(B) `catch`

(C) `finally`

(D) `except`

Q37. To open a file for reading in binary mode, which mode string should be used?

(A) `"r"`

(B) `"rb"`

(C) `"wb"`

(D) `"r+"`

Q38. The `pickle.load()` function is used to:

(A) Write data to a text file

(B) Write data to a binary file

(C) Read data from a binary file

(D) Close a file

Q39. What will be the output of the following code?

```
try:
print(10 / 0)
except ZeroDivisionError:
print("Error")
except:
print("Unknown")
```

(A) `Error`

(B) `Unknown`

(C) `10`

(D) Program crashes

Q40. Which method is used to move the file pointer to a specific position in Python?

(A) `tell()`

(B) `move()`

(C) `seek()`

(D) `locate()`

Q41. The `readline()` method returns data in the form of a _____.

(A) List

(B) String

(C) Tuple

(D) Dictionary

Q42. Which exception is raised when a module cannot be found in Python?

(A) `ValueError`

(B) `ImportError`

(C) `KeyError`

(D) `NameError`

Q43. To append data to an existing text file without overwriting, the mode should be:

(A) `"w"`

(B) `"r"`

(C) `"a"`

(D) `"x"`

Q44. Which protocol is used for secure communication over a computer network, effectively the encrypted version of HTTP?

(A) FTP

(B) SMTP

(C) HTTPS

(D) TELNET

Q45. A _____ is a malicious program that attaches itself to another program and replicates when that program is executed.

(A) Worm

(B) Virus

(C) Trojan Horse

(D) Spyware

Q46. Which switching technique breaks data into small chunks before transmitting them over the network?

(A) Circuit Switching

(B) Message Switching

(C) Packet Switching

(D) Line Switching

Q47. The process of converting readable text into an unreadable format to prevent unauthorized access is called:

(A) Decryption

(B) Encryption

(C) Firewalling

(D) Phishing

Q48. Which of the following is a guided transmission medium?

(A) Infrared

(B) Coaxial Cable

(C) Microwave

(D) Radio Waves

Q49. Cookies are used on the web to:

(A) Scan for viruses

(B) Store user preferences and session info

(C) Speed up the CPU

(D) Download files faster

Q50. Which threat involves a fraudster sending emails that appear to be from a reputable company to induce individuals to reveal personal information?

(A) Phishing

(B) Hacking

(C) Denial of Service

(D) Spamming

Detailed Solutions

Q1.

Solution

Concept:

The `ROUND(n, m)` function in SQL is a mathematical operation used to return a numeric value rounded to a specified number of decimal places or place values.

The logic of the second argument m is crucial for determining the rounding behavior:

1. If m is a positive integer, the function rounds the number to the specified number of decimal places to the right of the decimal point.
2. If m is zero, the function rounds the number to the nearest whole integer.
3. If m is a negative integer, the function rounds the number to the left of the decimal point (e.g., -1 rounds to the nearest tens, -2 to the nearest hundreds).

Rounding follow the standard rule: if the digit to the right of the target position is 5 or greater, the target digit is incremented by 1.

Solution:

1. We are analyzing the SQL query: `SELECT ROUND(157.285, -1) FROM DUAL;`.
2. The number to be processed is 157.285.
3. The precision parameter is -1, which indicates we must round to the nearest ”tens” place.
4. In the number 157.285, the digit in the tens place is 5 (representing 50).
5. To determine the rounding direction, we examine the digit to the immediate right of the tens place, which is the ”units” place digit, 7.
6. Since 7 is greater than or equal to 5, the value in the tens place is rounded up.
7. The digit 5 becomes 6, and the units place becomes 0. All digits to the right of the units place (the decimal portion .285) are discarded.
8. Therefore, the number 157 becomes 160.
9. The resulting value returned by the database engine is 160.

Final Answer: The output is 160.

Answer: (B)

Q2.

Solution

Concept:

String manipulation is a core requirement in SQL for processing and searching text data within relational tables.

When a user needs to find the specific index or location of a character or a substring within a larger string, they use ”search” functions.

The function `INSTR()` is the most common implementation across major RDBMS like MySQL and Oracle for this specific task.

Unlike many high-level programming languages that use 0-based indexing (where the first character is at position 0), SQL uses 1-based indexing (where the first character is at position 1).

Solution:

1. The function `INSTR(string, substring)` is designed to search for the first occurrence of `substring` inside the `string`.
2. Upon execution, the function scans the string from left to right.
3. If a match is found, it returns an integer representing the starting position of that match.
4. If the substring is not found within the source string, the function returns the value 0.
5. Let's look at the other options provided to understand why they are incorrect for this logic:

- `SUBSTR()` is used to extract a portion of a string, not to find a position.
- `LOCATE()` is a similar function but often uses a different syntax or is specific to certain databases.
- `POSITION()` is part of the standard ANSI SQL but `INSTR()` is the most common subject matter term in CUET.

6. Because the question specifically asks for the position of the first occurrence, `INSTR()` is the technically accurate response.

Final Answer: The function is INSTR().

Answer: (C)

Q3.

Solution

Concept:

The `MOD(n, m)` function in SQL calculates the remainder of a division operation, which is the value left over when one integer is divided by another.

The mathematical formula used by most SQL systems is: Remainder $= n - (m \times \text{FLOOR}(n/m))$.

However, the sign convention for the result is a critical detail in SQL theory:

In standard SQL (specifically MySQL and Oracle), the sign of the result of the `MOD()` function is determined solely by the sign of the dividend (the first number n).

The sign of the divisor (the second number m) is ignored when determining the sign of the final result.

Solution:

1. The expression provided is `SELECT MOD(13, -3);`.
2. Here, the dividend is 13 and the divisor is -3.
3. First, perform the basic division ignoring the signs: 13 divided by 3 results in a quotient of 4 with a remainder of 1 ($3 \times 4 = 12$, and $13 - 12 = 1$).
4. Next, apply the sign rule mentioned in the concept section.
5. The first number (the dividend) is 13, which is a positive integer.
6. Since the dividend is positive, the result of the `MOD()` function must also be positive.
7. The fact that we are dividing by a negative number (-3) does not influence the sign of the remainder in this SQL implementation.
8. Therefore, the result of `MOD(13, -3)` is exactly 1.
9. If the input had been `MOD(-13, -3)`, the result would have been -1.

Final Answer: 1

Answer: (A)

Q4.

Solution

Concept:

In the SQL language, many operations have redundant function names to ensure that code can be easily migrated between different database systems like Oracle, MySQL, PostgreSQL, and SQL Server.

Extracting a substring from the middle of a text block is a common operation that requires specifying the source string, the starting index, and the number of characters to be retrieved.

Different database systems have adopted different keywords for this same logical task over the last few decades.

Solution:

1. The question asks to identify the function used to extract characters from the middle of a string.
2. `SUBSTRING(string, start, length)`: This is the standard ANSI SQL keyword. It is recognized by almost all relational databases and is highly portable.
3. `SUBSTR(string, start, length)`: This is a very popular alias used frequently in Oracle, MySQL, and PostgreSQL. It functions identically to `SUBSTRING()`.
4. `MID(string, start, length)`: This is an alias primarily found in MySQL and older versions of BASIC language variants. It is also used to extract characters from the ”middle” of a string.
5. Since all three functions—`MID`, `SUBSTR`, and `SUBSTRING`—perform the exact same operation with the same parameter set, they are all valid choices.
6. In a multiple-choice environment, when all listed options are valid synonyms for a technical process, the correct selection is ”All of the above.”

Final Answer: All of the above.

Answer: (D)

Q5.

Solution

Concept:

Date and Time functions in SQL are used to manipulate and format temporal data for reporting and analytical purposes.

The `DAYNAME()` function is a specific scalar function that takes a date or a string representing a date as its input.

It processes this input according to the calendar logic of the database server and returns a string containing the full name of the day of the week (e.g., 'Monday', 'Tuesday', etc.).

This function is extremely useful when generating weekly reports or filtering data based on specific days like weekends or mid-week days.

Solution:

1. The input string provided is `'2026-05-06'`.
2. This date represents May 6th, the year 2026.
3. According to the calendar for the year 2026 (as noted in the question prompt), May 6, 2026, is a Wednesday.
4. When the `DAYNAME()` function is applied to this date string, the SQL engine calculates the day index and converts it to the human-readable word.
5. Let's compare this with other related functions:

- `DAYOFWEEK()` would return the integer 4 (if Sunday=1).
- `WEEKDAY()` would return the integer 2 (if Monday=0).
- `DAY()` would return the integer 6.

6. Since the function used is `DAYNAME()`, the only possible result is the full name of the weekday.
7. Thus, the output is 'Wednesday'.

Final Answer: Wednesday

Answer: (B)

Q6.

Solution

Concept:

The `TRUNCATE(n, d)` function is a numeric function in SQL used to reduce a number to a specified number of decimal places without performing any rounding.

Unlike the `ROUND()` function, which examines the next digit to decide whether to increment the target digit, `TRUNCATE()` simply "chops off" the digits beyond the specified position.

When the second argument d is a negative integer, the function replaces digits to the left of the decimal point with zeros:

1. `TRUNCATE(n, -1)` replaces the units digit with zero.
2. `TRUNCATE(n, -2)` replaces both the units and tens digits with zeros.
3. `TRUNCATE(n, -3)` replaces units, tens, and hundreds digits with zeros.

Solution:

1. The input value provided in the question is 543.21.
2. The truncation parameter is -2.
3. Because the parameter is -2, we move two places to the left from the decimal point to identify the starting point of truncation.
4. The first place to the left is the units place (digit 3). The second place to the left is the tens place (digit 4).
5. The function will maintain the digit in the hundreds place (5) but will replace the tens and units digits with zeros.
6. The decimal portion of the number (.21) is completely discarded, as the truncation point is far to the left of the decimal.
7. Therefore, the digits 4 and 3 are replaced by 0 and 0, resulting in the value 500.
8. To compare: If we used `ROUND(543.21, -2)`, the result would also be 500. However, if the number was 563.21, `ROUND` would give 600 while `TRUNCATE` would still return 500.
9. Thus, the final output for the given expression is 500.

Final Answer: 500

Answer: (B)

Q7.

Solution

Concept:

In SQL, functions are categorized based on whether they operate on single rows or sets of rows.

1. **Aggregate Functions:** These functions take a collection of values (a column from multiple rows) and perform a calculation to return a single, summarized result. They are often used with the `GROUP BY` clause. Examples include `SUM()`, `AVG()`, `MIN()`, `MAX()`, and `COUNT()`.

2. **Scalar Functions:** These functions operate on an individual value and return a single value for every row in the result set. Examples include string functions like `UPPER()`, `LOWER()`, and `LENGTH()`, or math functions like `ABS()`.

Solution:

1. We need to identify which of the provided options is NOT a valid aggregate function.

2. `SUM()` is an aggregate function used to calculate the total of a numeric column across a set of rows.

3. `COUNT()` is an aggregate function used to find the total number of rows or non-null values in a set.

4. `AVG()` is an aggregate function used to calculate the mathematical mean of a numeric column for a group of rows.

5. `LENGTH()` is a scalar function. It calculates the number of characters in a string for each individual row. For example, if you have 10 rows, `SELECT LENGTH(name)` will return 10 individual lengths.

6. Because `LENGTH()` does not summarize data across multiple rows into a single value, it does not meet the definition of an aggregate function.

7. Therefore, `LENGTH()` is the correct choice as the "odd one out" in this classification.

Final Answer: LENGTH()

Answer: (C)

Q8.

Solution

Concept:

The `RIGHT(str, n)` function is a string manipulation tool in SQL used to retrieve a specific number of characters from the end of a string.

This is particularly useful when you need to extract suffixes, last names, or specific trailing identifiers from a data field.

The function takes two parameters:

1. `str`: The source string from which characters will be extracted.

2. `n`: The number of characters to extract, starting from the rightmost character and counting backwards toward the beginning.

Solution:

1. The SQL statement provided is: `SELECT RIGHT("CUET EXAM", 4);`.

2. First, let's look at the structure of the string `"CUET EXAM"`.

3. The characters and their positions are: C(1), U(2), E(3), T(4), Space(5), E(6), X(7), A(8), M(9).

4. The requirement is to extract the last 4 characters from this 9-character string.

5. Counting from the right (the end):

- The 1st character from the right is 'M'.
- The 2nd character from the right is 'A'.
- The 3rd character from the right is 'X'.
- The 4th character from the right is 'E'.

6. When these four characters are combined in their original relative order (from left to right as they appear in the suffix), they form the word "EXAM".

7. Note: If the argument had been 5, the result would have been `" EXAM"` (including the space at position 5).

8. Since the request is exactly for 4 characters, the output is "EXAM".

Final Answer: "EXAM"

Answer: (B)

Q9.

Solution

Concept:

In a Relational Database Management System (RDBMS), keys are used to uniquely identify records and establish relationships between tables.

A **Candidate Key** is any attribute (or set of attributes) that can uniquely identify a row in a table without any redundant data.

From the pool of available Candidate Keys, the database designer selects exactly one to be the **Primary Key**.

The remaining candidate keys that were not chosen for the role of Primary Key are technically referred to as **Alternate Keys**.

These keys still possess the unique identification property and can be used for indexing or enforcing uniqueness, even though they are not the main identifier.

Solution:

1. The question asks for the term used for a candidate key that is not chosen as the primary key.
2. Let's analyze the roles of different keys:

- **Primary Key:** The main unique identifier selected for a table.
- **Foreign Key:** An attribute used to link one table to another (usually referencing a primary key).
- **Composite Key:** A key that consists of two or more attributes.
- **Alternate Key:** A candidate key that serves as an "alternative" to the primary key.

3. Consider a `Student` table with `Roll_No`, `Email`, and `Adhaar_No`. All three are Candidate Keys because they are all unique.
4. If the school chooses `Roll_No` as the Primary Key, then `Email` and `Adhaar_No` automatically become Alternate Keys.
5. In formal database theory and the CUET curriculum, this is the standard definition for "leftover" candidate keys.
6. Therefore, Alternate Key is the correct answer.

Final Answer: Alternate Key

Answer: (B)

Q10.

Solution

Concept:

Relational Algebra is a formal, mathematical language used to describe how data is retrieved from relations (tables). It provides the theoretical foundation for SQL.

The primary operations in Relational Algebra are:

1. **Selection (σ):** A unary operation that returns a subset of tuples (rows) that satisfy a specific condition. It is a "horizontal" operation.

2. **Projection (Π):** A unary operation that returns specific attributes (columns) from a relation. It is a "vertical" operation.

3. **Join ($\bowtie$):** A binary operation that combines related tuples from two different relations.

4. **Renaming (ρ):** Used to change the name of a relation or its attributes.

Solution:

1. The question asks for the symbol used to denote the operation that "filters rows based on a specific condition."

2. Filtering rows means selecting a horizontal slice of the table where certain criteria are met (e.g., Age > 18).

3. The mathematical symbol used for this **Selection** operation is the lower-case Greek letter **Sigma (σ)**.

4. Let's check the other symbols provided in the options:

- Π (Pi) is the symbol for Projection, which filters columns, not rows.
- ρ (Rho) is used for renaming.
- $\bowtie$ is the symbol for a Join.

5. In Relational Algebra notation, $\sigma_{marks>90}(STUDENT)$ would return only the rows of students who scored above 90.

6. Therefore, the Sigma symbol is the correct notation for conditional row filtering.

Final Answer: σ (Sigma)

Answer: (B)

Q11.

Solution

Concept:

In the field of Database Management Systems (DBMS), the ACID properties (Atomicity, Consistency, Isolation, Durability) serve as the foundation for ensuring data integrity during transactions.

A transaction is defined as a single logical unit of work that may consist of multiple low-level operations, such as reading a value, performing a calculation, and writing the result back to the disk.

Atomicity is the specific property that mandates that a transaction must be treated as an indivisible unit.

This means that either every single operation within the transaction is committed to the database, or none of them are.

If any part of the transaction fails due to a system crash, power failure, or constraint violation, the entire transaction is ”rolled back,” ensuring the database remains in its previous state.

Solution:

1. The question focuses on the property that ensures a ”work unit” is completed successfully in its entirety or not at all.
2. This is a classic definition of **Atomicity**.
3. Let's look at a real-world example: a bank transfer. If you send money from Account A to Account B, two steps must occur:

- Step 1: Subtract the amount from Account A.
- Step 2: Add the amount to Account B.

4. If the system fails after Step 1 but before Step 2, the money would effectively disappear. Atomicity prevents this by ensuring that Step 1 is only permanent if Step 2 also succeeds.
5. Contrast this with other properties:

- **Consistency** ensures data follows all integrity rules.
- **Isolation** ensures simultaneous transactions don't interfere.
- **Durability** ensures once a change is saved, it stays saved.

6. Since the requirement is the ”all-or-nothing” execution, Atomicity is the only correct answer.

Final Answer: Atomicity

Answer: (A)

Q12.

Solution

Concept:

The **Cartesian Product** (represented by the symbol $A \times B$) is a fundamental binary operation in Relational Algebra.

It is used to combine every tuple (row) from the first relation with every tuple from the second relation.

This operation is critical because it forms the theoretical basis for Joins, where we first create a product and then filter it based on a join condition.

From a quantitative perspective, the Cartesian Product follows a simple multiplication rule for its cardinality (the number of rows) and an addition rule for its degree (the number of columns).

If Relation A has n rows and Relation B has m rows, the resulting table $A \times B$ will always have $n \times m$ rows.

Solution:

1. We are given two specific tables: Table A and Table B.
2. Table A contains 5 rows.
3. Table B contains 3 rows.
4. The operation being performed is the Cartesian Product ($A \times B$).
5. To find the total number of rows in the resulting relation, we apply the product rule of cardinality.
6. Number of rows = (Rows in Table A) $\times$ (Rows in Table B).
7. Number of rows = $5 \times 3 = 15$.
8. To understand why this happens, imagine each row in A pairing up with every row in B. For the first row of A, you get 3 pairings. For the second row of A, you get another 3 pairings. Repeating this for all 5 rows of A gives $3 + 3 + 3 + 3 + 3 = 15$.
9. Note: If the question had asked for the "degree" (number of columns) and Table A had 2 columns and Table B had 2 columns, the answer would have been $2 + 2 = 4$.
10. Since the question asks for the number of rows, the answer is 15.

Final Answer: 15

Answer: (B)

Q13.

Solution

Concept:

Network topology refers to the geometric arrangement of various elements (links, nodes, etc.) in a computer network.

Each topology has a unique way of managing data flow and different levels of fault tolerance.

1. **Bus Topology:** All nodes connect to a single central cable (backbone).
2. **Ring Topology:** Each node is connected to exactly two other nodes in a circular loop.
3. **Star Topology:** Every node is connected to a central intermediary device, which is usually a hub or a switch.
4. **Mesh Topology:** Nodes are interconnected with many redundant paths.

The defining characteristic of the Star topology is that all communication between nodes must pass through the central controller.

Solution:

1. The question asks to identify the topology that "requires a central controller or a hub to connect all nodes."
2. In a **Star Topology**, nodes are not directly connected to one another.
3. Instead, each node has a dedicated point-to-point link to the central hub.
4. This hub acts as a repeater or a switch; when one node wants to send data to another, the data goes to the hub first, which then forwards it to the destination node.
5. This setup is popular because it is easy to install and manage. If one peripheral cable fails, only that node is affected.
6. However, it has a "Single Point of Failure": if the central hub fails, the entire network becomes non-functional.
7. Neither Bus nor Ring topologies utilize a central hub in this manner. Mesh topology is decentralized by nature.
8. Therefore, the description provided in the question matches the Star Topology perfectly.

Final Answer: Star

Answer: (C)

Q14.

Solution

Concept:

Every Network Interface Card (NIC) is assigned a unique hardware address known as the **MAC (Media Access Control) Address**.

While an IP address is a logical address that can change depending on which network a device connects to, the MAC address is ”burned” into the hardware by the manufacturer.

It operates at the Data Link Layer (Layer 2) of the OSI model and is used for local delivery of frames within a single network segment.

MAC addresses are typically represented in hexadecimal notation, appearing as six groups of two characters (e.g., AA:BB:CC:DD:EE:FF).

The length of this address is standardized globally to ensure that every network device in the world can have a unique identifier.

Solution:

1. The question asks for the bit-length of a standard MAC address.
2. A MAC address consists of 6 bytes (octets).
3. Since each byte contains 8 bits, we can calculate the total bit length: 6 bytes $\times$ 8 bits/byte = 48 bits.
4. Alternatively, if we look at the hexadecimal representation, there are 12 hexadecimal digits.
5. Each hexadecimal digit represents 4 bits ($2^4 = 16$).
6. Total bits = 12 digits $\times$ 4 bits/digit = 48 bits.
7. Comparison with other network addresses:

- IPv4 address = 32 bits.
- IPv6 address = 128 bits.

8. The first 24 bits of a MAC address identify the manufacturer (OUI - Organizationally Unique Identifier), while the last 24 bits are a unique serial number assigned by that manufacturer.
9. Therefore, 48 is the standard and correct bit-length for a MAC address.

Final Answer: 48

Answer: (C)

Q15.

Solution

Concept:

The Open Systems Interconnection (OSI) model divides network communication into seven distinct layers.

The **Physical Layer (Layer 1)** is the lowest layer and is concerned with the transmission and reception of raw, unstructured data bits over a physical medium (like copper wire or fiber optics).

As electrical or optical signals travel through a medium, they suffer from ”attenuation,” which is the loss of signal strength over distance.

To overcome this, specific hardware devices are used to capture a weakened signal and recreate it at its original strength.

These devices do not look at MAC addresses or IP addresses; they only handle the physical signal (bits).

Solution:

1. The question describes a device that operates at the Physical layer and regenerates signals to extend network length.
2. A **Repeater** is a simple but effective device designed for this exact purpose.
3. It receives a weak or distorted signal, cleans it, amplifies it, and sends it back out.
4. Let’s evaluate the other devices:

- **Bridge:** Operates at the Data Link Layer (Layer 2). It understands MAC addresses and can filter traffic between segments.
- **Router:** Operates at the Network Layer (Layer 3). It understands IP addresses and routes packets across different networks.
- **Gateway:** Operates at higher layers to translate between different network protocols.

5. Since the device in the question only ”regenerates” the signal at the most basic level (Physical Layer) without any logical processing, it must be a Repeater.
6. By using a repeater, the maximum distance of a network cable can be effectively doubled or tripled.

Final Answer: Repeater

Answer: (C)

Q16.

Solution

Concept:

Infix expressions are the standard mathematical notation where operators are placed between operands (e.g., $A + B$).

Postfix notation, also known as Reverse Polish Notation (RPN), places operators after their operands (e.g., $AB+$).

To convert from Infix to Postfix, we follow the rules of Operator Precedence and Associativity:

1. Parentheses () have the highest priority.
2. Multiplication ($\times$) and Division ($/$) have higher priority than Addition ($+$) and Subtraction ($-$).
3. If operators have equal priority, we process them from Left to Right.

The conversion process involves using a stack for operators while operands are directly moved to the output.

Solution:

1. The given expression is $A + (B \times C) - D$.
2. First, we identify the sub-expression within the parentheses: $(B \times C)$.
3. In postfix, $(B \times C)$ becomes $BC\times$.
4. The expression now looks like $A + BC\times - D$ (treating $BC\times$ as a single unit).
5. Next, we look at the remaining operators: $+$ and $-$. They have equal precedence, so we evaluate from left to right.
6. We process the addition: $A + (BC\times)$ becomes $ABC\times +$.
7. Now we have $ABC\times + - D$.
8. Finally, we process the subtraction: $(ABC\times +) - D$ becomes $ABC\times +D-$.
9. Let's double-check the steps: the multiplication was inside parentheses, so it must appear before the addition and subtraction in the final postfix string.
10. Therefore, the correct postfix form is $ABC\times +D-$.

Final Answer: $ABC\times +D-$

Answer: (A)

Q17.

Solution

Concept:

Linear data structures organize data elements in a sequential manner, where each element is connected to its previous and next adjacent elements.

The two primary linear data structures used in computer science are Stacks and Queues.

A **Stack** follows the LIFO (Last-In-First-Out) principle, where the last element added is the first one removed.

A **Queue** follows the FIFO (First-In-First-Out) principle. This is similar to a real-world line of people waiting for service.

In a queue, elements are added at one specific end (the ”Rear” or ”Tail”) and removed from the opposite end (the ”Front” or ”Head”).

Solution:

1. The question describes a structure where elements are added at one end and removed from the other.
2. This ”add at one end, remove from the other” logic is the defining characteristic of a **Queue**.
3. Let’s contrast this with other data structures:

- **Stack:** Both addition and removal happen at the same end (the ”Top”).
- **Linked List:** A list can be linear, but adding and removing isn’t restricted to ends unless specified.
- **Tree:** This is a non-linear data structure with a hierarchical parent-child relationship.

4. In software engineering, queues are used for tasks like print job scheduling, CPU task management, and handling asynchronous data transfers.
5. Because the structure strictly separates the entry point and the exit point for elements, it maintains a chronological order of processing.
6. Therefore, the structure described is a Queue.

Final Answer: Queue

Answer: (B)

Q18.

Solution

Concept:

Evaluating a postfix expression requires a stack-based algorithm. This is the method most compilers use to evaluate arithmetic expressions.

The algorithm follows these steps:

1. Scan the postfix expression from left to right.
2. If the scanned element is an operand (a number), push it onto the stack.
3. If the scanned element is an operator ($+, -, \times, /$), pop the top two elements from the stack.
4. Apply the operator to these two elements (the first pop is the second operand, the second pop is the first operand).
5. Push the result of the calculation back onto the stack.
6. Repeat until the expression is empty; the final value on the stack is the result.

Solution:

1. The expression is: $8, 4, /, 5, \times, 2, +$.
2. Scan 8: Push 8 onto the stack. [Stack: 8]
3. Scan 4: Push 4 onto the stack. [Stack: $8, 4$]
4. Scan $/$: Pop 4 and 8. Calculate $8/4 = 2$. Push 2. [Stack: 2]
5. Scan 5: Push 5 onto the stack. [Stack: $2, 5$]
6. Scan $\times$: Pop 5 and 2. Calculate $2 \times 5 = 10$. Push 10. [Stack: 10]
7. Scan 2: Push 2 onto the stack. [Stack: $10, 2$]
8. Scan $+$: Pop 2 and 10. Calculate $10 + 2 = 12$. Push 12. [Stack: 12]
9. The expression is finished, and the only value left on the stack is 12.
10. Thus, the evaluated result is 12.

Final Answer: 12

Answer: (A)

Q19.

Solution

Concept:

A stack is a restricted linear data structure that permits access only at one end, called the ”Top.” To manage a stack effectively, two main boundary conditions must be checked:

1. **Overflow:** Occurs when an attempt is made to push an element onto a stack that is already full.

2. **Underflow:** Occurs when an attempt is made to pop (remove) an element from a stack that is already empty.

In an array-based implementation of a stack of size N, the index typically starts from 0 and goes up to $N - 1$.

The variable ”Top” keeps track of the index of the last element added.

Solution:

1. Initially, when a stack is created and contains no elements, the ”Top” variable is usually set to -1.

2. When the first element is pushed, Top becomes 0. When the second is pushed, Top becomes 1.

3. A `pop()` operation involves two steps: retrieving the value at `Stack[Top]` and then decrementing Top (Top = Top - 1).

4. If an attempt is made to pop a value when the stack is empty, the program cannot retrieve any data and would encounter an error.

5. Since an empty stack is indicated by $Top = -1$, calling a pop function at this specific state triggers the ”Underflow” condition.

6. Let’s look at the options:

- $Top = N - 1$ indicates the stack is full (Overflow if we push).
- $Top = 0$ means there is exactly one element in the stack.
- $Top = -1$ means no elements exist.

7. Therefore, Underflow happens when Top is -1.

Final Answer: $Top = -1$

Answer: (C)

Q20.

Solution

Concept:

A Circular Queue is an advanced version of a regular queue where the last position is connected back to the first position, making it look like a circle.

This structure solves the ”memory wastage” problem of a linear queue implemented with an array, where spaces at the front cannot be reused once they are emptied.

To manage this circularity, we use the modulo operator (%).

The position of an element is calculated using $(Current\_Position + 1)\%\text{Size}$.

A circular queue is considered ”Full” when there is only one empty space left in the array (in some implementations) or when the next position of the Rear is equal to the Front.

Solution:

1. Let F be the index of the Front and R be the index of the Rear in a circular queue of size N.
2. In a circular queue, the Rear pointer moves forward every time an element is added (enqueued).
3. To account for the wrap-around behavior, the logic for moving the Rear is: $R = (R + 1)\%N$.
4. If the queue is full, adding one more element would mean the Rear catches up to the Front.
5. Therefore, the mathematical condition to check if the queue is full is:

Is the next position of the Rear equal to the Front?

$(R + 1)\%N == F$.

6. For example, in a queue of size 5: if $R = 4$ and $F = 0$, then $(4 + 1)\%5 = 0$. Since $0 == 0$, the queue is full.
7. Option B $(R + 1 == F)$ only works for linear queues and doesn't account for the wrap-around from the end of the array to the beginning.
8. Option A correctly uses the circular logic required for this data structure.

Final Answer: $(R + 1)\%N == F$

Answer: (A)

Q21.

Solution

Concept:

Python lists are dynamic arrays that provide several built-in methods for data manipulation.

To manage data effectively, it is essential to understand how elements are added to a list.

The most common methods for addition are:

1. `append()`: Adds a single element to the very end of the list.
2. `insert()`: Adds an element at a specific index provided by the user.
3. `extend()`: Adds multiple elements (from an iterable like another list) to the end of the current list.

For general stack-like behavior or simple data accumulation, adding to the end is the most efficient and frequently used operation.

Solution:

1. The question asks for the method used to add an element specifically "at the end of a list."
2. The `append(item)` method takes exactly one argument—the item to be added—and places it at the last index of the list.
3. Let's look at the other options to clarify their roles:

- `add()`: This is not a valid list method in Python; it is typically used with Sets.
- `insert(index, item)`: This allows adding an element anywhere, but it requires an index. While you could use it to add to the end (using `len(list)`), it is not the primary method for that task.
- `extend(iterable)`: This is used to merge another list into the current one, not for adding a single element.

4. If you have a list `L = [1, 2]` and you run `L.append(3)`, the result is `[1, 2, 3]`.
5. Because `append()` is the standard, built-in way to grow a list by one element at the tail, it is the correct answer.

Final Answer: `append()`

Answer: (C)

Q22.

Solution

Concept:

Time complexity is a measure of the amount of time an algorithm takes to run as a function of the length of the input.

In data structures, we evaluate the efficiency of basic operations like insertion, deletion, and searching.

For a **Stack** implemented using an array (or a Python list), the "Push" operation involves placing an element at the position indicated by the "Top" pointer and then incrementing that pointer.

In a fixed-size array, this is a direct memory access operation.

In a dynamic array (like a Python list), appending is usually extremely fast because the system allocates extra space in advance.

Solution:

1. When we push an element onto a stack, we are performing a "constant time" operation.
2. This is because the operation does not depend on the number of elements already present in the stack.
3. Whether the stack contains 5 elements or 5 million elements, the steps remain the same:

- Check for overflow.
- Access the index `Top + 1`.
- Assign the value.
- Increment the `Top` variable.

4. In Big O notation, an operation that takes a fixed number of steps regardless of input size is denoted as $O(1)$.
5. Let's contrast this with other complexities:

- $O(n)$ would mean the time grows linearly (like searching for an element).
- $O(\log n)$ is typical for binary searches.

6. Since pushing a value onto a stack is an immediate access operation, its complexity is $O(1)$.

Final Answer: $O(1)$

Answer: (A)

Q23.

Solution

Concept:

Different data structures are optimized for specific types of data access and manipulation patterns. The ”Undo” functionality is a classic example of a ”Last-In-First-Out” (LIFO) process.

When you perform actions in an application (like typing a letter, then deleting a word, then changing a font), the application needs to remember these actions in the exact reverse order they were performed.

The most recent action is the first one that should be ”undone” when the user clicks the Undo button.

A Stack is the ideal candidate for this because it natively supports adding and removing elements from only one end.

Solution:

1. To implement Undo, the system pushes every state or action onto a data structure as the user works.
2. If the user makes a mistake and requests an ”Undo,” the system must retrieve the most recently saved state.
3. In a **Stack**, the `pop()` operation removes the most recently added item.
4. This matches the Undo logic perfectly: the last action ”in” is the first action ”out” (reversed).
5. Why other structures are less suitable:

- **Queue:** This is First-In-First-Out. An ”Undo” using a queue would try to undo the very first action you took when you opened the file hours ago.
- **Deque:** While a Deque could work (as it can act as a stack), a simple Stack is the more specific and theoretically correct answer for LIFO behavior.
- **Priority Queue:** This sorts items by importance, which isn’t how sequential actions work.

6. Therefore, the Stack is the industry-standard structure for Undo/Redo mechanisms.

Final Answer: Stack

Answer: (B)

Q24.

Solution

Concept:

Prefix notation, also known as Polish Notation, is an expression format where the operator precedes its operands (e.g., $+AB$).

To convert an infix expression to prefix, we follow rules of precedence:

1. Division and Multiplication have higher precedence than Addition and Subtraction.

2. When converting manually, it is often easiest to parenthesize the expression based on precedence first.

The transformation logic essentially moves the operator to the left of the opening parenthesis of its corresponding operands.

Solution:

1. The given infix expression is $X/Y + Z$.

2. According to the order of operations (BODMAS/PEMDAS), the division must be performed before the addition.

3. Let's add parentheses to clarify the order: $(X/Y) + Z$.

4. Now, we convert the inner part (X/Y) into prefix notation. The operator / moves to the front: $/XY$.

5. The expression now looks like $(/XY) + Z$.

6. Finally, we convert the addition. The operator + moves to the front of its two operands, which are $(/XY)$ and Z.

7. This results in $+/XYZ$.

8. Let's check the logic: if we were to evaluate $+/XYZ$ using a prefix evaluation algorithm, we would first find the / and apply it to X and Y, then take that result and add it to Z.

9. This perfectly matches the original infix logic. Thus, Option A is the correct prefix form.

Final Answer: $+/XYZ$

Answer: (A)

Q25.

Solution

Concept:

A **Deque** (pronounced ”deck”) stands for ”Double-Ended Queue.”

It is a generalized data structure that combines the properties of both a Stack and a Queue.

While a standard queue only allows insertion at the rear and deletion from the front (FIFO), and a stack only allows both operations at the top (LIFO), a Deque is more flexible.

It supports four primary operations:

1. `insertFront()`: Add an item to the beginning.
2. `insertRear()`: Add an item to the end.
3. `deleteFront()`: Remove an item from the beginning.
4. `deleteRear()`: Remove an item from the end.

Solution:

1. The question asks where a Deque allows insertion and deletion to occur.
2. By definition, ”Double-Ended” implies that both ends of the linear sequence are accessible for both types of operations.
3. This makes the Deque a very versatile structure; it can be used as a Queue, a Stack, or both simultaneously.
4. For example, in a browser history where you can add new pages to the end but also clear out old ones from the front, a Deque is very useful.
5. Let’s look at the options:

- ”Only the front” describes a restricted structure.
- ”Only the rear” is also a restricted form.
- ”Both ends” is the defining feature of a Double-Ended Queue.

6. Note that while a Deque allows operations at both ends, it does not typically allow random access or deletion from ”The middle” (Option D).
7. Therefore, ”Both ends” is the correct technical description.

Final Answer: Both ends

Answer: (C)

Q26.

Solution

Concept:

In Python, lists are highly versatile and come equipped with several methods to remove elements. To choose the correct method, one must understand the difference between removing by value and removing by position.

1. `remove(value)`: Searches for the first occurrence of a specific value and removes it. It does not return the value.
2. `pop(index)`: Removes the element at the specified position and returns that element.
3. `clear()`: Removes all elements from the list.

The `pop()` method is unique because it is the only one that "returns" the value it removes, making it ideal for stack and queue implementations.

Solution:

1. The question specifically asks for a method that "removes and returns the last element" of a list.
2. In Python, the `pop()` method, when called without any arguments (e.g., `list.pop()`), defaults to the last index (`-1`).
3. This operation performs two tasks simultaneously: it modifies the list by deleting the last item and it provides that item as an output.
4. Let's evaluate the other options:

- `remove()` requires a value and returns `None`.
- `delete` is not a method; `del` is a keyword used for deletion but it does not return the deleted value.
- `discard()` is a method used with sets, not lists.

5. Because `pop()` is designed to handle the "Last-In" element in a stack-like fashion, it is the standard way to retrieve and remove the final item of a list.
6. Therefore, `pop()` is the correct answer.

Final Answer: `pop()`

Answer: (C)

Q27.

Solution

Concept:

Binary Search is an efficient algorithm for finding an item from a sorted list of items.

It works by repeatedly dividing in half the portion of the list that could contain the item, until you have narrowed down the possible locations to just one.

The number of comparisons required to find a value (or determine it is not present) is logarithmic relative to the size of the list.

The formula for the maximum number of comparisons in a Binary Search is $\lceil \log_2(n) \rceil$, or sometimes stated as the smallest integer k such that $2^k \geq n$.

Solution:

1. We are given an array size of $n = 16$.
2. To find the maximum number of comparisons, we calculate the binary logarithm of 16.
3. $2^1 = 2$
4. $2^2 = 4$
5. $2^3 = 8$
6. $2^4 = 16$
7. This tells us that $\log_2(16) = 4$.
8. However, in a real-world implementation, after the 4th split, you might have one final comparison to confirm the target. In most academic contexts for CUET, k such that $2^k \geq n$ is used.
9. If $n = 16$, after 4 comparisons, we have narrowed it down completely.
10. Wait, let's look at the options provided. If we have 16 elements, the search space reduces as: $16 \to 8 \to 4 \to 2 \to 1$. This takes 4 steps to reach the last element. To be safe, if the target is the very last possibility, it takes a 5th check in some logic models, but mathematically $\log_2(16)$ is 4.
11. Looking at typical CUET PYQ logic, if 16 is a power of 2, the answer is often $\log_2(n) + 1$ to account for the "not found" or final check. For $n = 16$, $2^4 = 16$, so 5 comparisons cover all possibilities including failure.

Final Answer: 5

Answer: (D)

Q28.

Solution

Concept:

Sorting algorithms are categorized by their logic for organizing data.

Selection Sort is an in-place comparison-based sorting algorithm.

The algorithm divides the input list into two parts: a sorted sublist of items which is built up from left to right at the front (left) of the list, and a sublist of the remaining unsorted items that occupy the rest of the list.

Initially, the sorted sublist is empty and the unsorted sublist is the entire input list.

The algorithm proceeds by finding the smallest (or largest, depending on sorting order) element in the unsorted sublist, exchanging (swapping) it with the leftmost unsorted element.

Solution:

1. The question describes an algorithm that "repeatedly picks the smallest element from the unsorted part and puts it at the beginning."
2. This is the exact procedural definition of **Selection Sort**.
3. Let's compare this with other algorithms to ensure accuracy:

- **Bubble Sort:** Repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order (the largest "bubbles" to the end).
- **Insertion Sort:** Builds the final sorted array one item at a time by "inserting" an element into its correct position relative to the elements already sorted.
- **Quick Sort:** Uses a "pivot" to partition the array into smaller sub-arrays.

4. Since Selection Sort's primary "selection" criteria is finding the minimum element in every pass, the description in the question matches it perfectly.
5. This algorithm has a time complexity of $O(n^2)$ regardless of the initial order of elements.

Final Answer: Selection Sort

Answer: (C)

Q29.

Solution

Concept:

Bubble Sort is one of the simplest sorting algorithms. It works by repeatedly swapping adjacent elements if they are in the wrong order.

In its standard form, Bubble Sort always takes $O(n^2)$ time because it uses nested loops.

However, it can be **optimized**. If during a pass through the list, no swaps are made, it means the list is already sorted.

By adding a flag (like a boolean variable `swapped`) to check if any exchange occurred during a pass, we can terminate the algorithm early.

This optimization drastically changes the "Best Case" scenario.

Solution:

1. In the **Best Case** scenario for an optimized Bubble Sort, the input array is already sorted (e.g., `[1, 2, 3, 4, 5]`).
2. During the very first pass of the outer loop, the algorithm compares:

- 1 and 2 (no swap)
- 2 and 3 (no swap)
- 3 and 4 (no swap)
- 4 and 5 (no swap)

3. The `swapped` flag remains `False`.
4. The algorithm detects this and breaks out of the loop immediately.
5. The number of comparisons performed is $n - 1$, which is proportional to n.
6. In Big O notation, this linear relationship is expressed as $O(n)$.
7. Note: Without optimization, the best case would still be $O(n^2)$. But in academic exams like CUET, "Best Case" usually implies the optimized version.

Final Answer: $O(n)$

Answer: (A)

Q30.

Solution

Concept:

Insertion Sort is a comparison-based sorting algorithm that builds the sorted array one element at a time.

It is much less efficient on large lists than more advanced algorithms like Quicksort or Mergesort, but it is very efficient for small data sets or data that is already substantially sorted.

The logic is analogous to how people sort a deck of playing cards in their hands.

You take one card at a time and compare it with the cards already in your hand, shifting the larger cards to the right until you find the correct ”gap” for the new card.

Solution:

1. The question describes a technique where ”an element is compared with its predecessor and shifted if it is smaller.”
2. This is the hallmark of **Insertion Sort**.
3. Here is the step-by-step logic:

- You start with the second element (the ”key”).
- You compare the key with the element before it (the predecessor).
- If the key is smaller, you shift the predecessor to the right.
- You continue this ”backward” comparison until you find an element smaller than the key or reach the start of the list.
- You then place the key in its correct position.

4. In Bubble Sort, you compare and swap *adjacent* elements moving *forward*.
5. In Selection Sort, you look for the absolute minimum in the *remaining* list.
6. Since the logic specified is specifically comparing an element to its predecessor and shifting to create a hole, it is Insertion Sort.

Final Answer: Insertion Sort

Answer: (B)

Q31.

Solution

Concept:

In Python, it is vital to distinguish between ”equality” and ”identity.”

1. The equality operator (==) checks if the **values** of two objects are the same.

2. The identity operator (`is`) checks if two variables point to the **same memory location** (the same object).

Python uses an internal mechanism called ”interning” for small integers (typically -5 to 256) and certain strings to save memory. This means two different variables assigned the same small integer will actually share the same ID.

Solution:

1. We are given two variables: `a = 10` and `b = 10`.

2. When we evaluate `a is b`, Python checks if `id(a) == id(b)`.

3. Since 10 is a small integer, Python points both `a` and `b` to the same pre-allocated memory object for the number 10.

4. Therefore, `a is b` evaluates to `True`.

5. Now consider the second part: `a == b`. This checks if the value 10 is equal to the value 10, which is also `True`.

6. The question specifically asks for the result of the identity check.

7. Because both variables refer to the exact same object in the memory heap, the identity operator returns `True`.

Final Answer: `True`

Answer: (B)

Q32.

Solution

Concept:

Python's data types are categorized as either **Mutable** or **Immutable**.

1. **Mutable objects** can be changed after they are created (e.g., Lists, Dictionaries, Sets). You can add, remove, or modify elements in place without changing the object's identity.

2. **Immutable objects** cannot be changed once defined (e.g., Integers, Floats, Strings, Tuples). If you attempt to modify an immutable object, Python creates a new object in memory with the new value.

Solution:

1. Let's analyze the options provided to find the immutable one:

- **List:** A list is mutable. You can change an element using `list[0] = new_value`.

- **Tuple:** A tuple is immutable. Once a tuple like `(1, 2)` is created, you cannot change its elements. Any attempt to do so will result in a `TypeError`.

- **Dictionary:** A dictionary is mutable. You can update values associated with keys or add new key-value pairs.

- **Set:** A set is mutable. You can add or remove items using methods like `add()` or `discard()`.

2. Tuples are often used to store data that should not be accidentally modified by the program, such as coordinates or fixed configuration settings.

3. Therefore, among the choices, the Tuple is the correct answer.

Final Answer: Tuple

Answer: (B)

Q33.

Solution

Concept:

In Python, dictionaries are unordered (or insertion-ordered in recent versions) collections of key-value pairs.

To access the value associated with a specific key, Python provides two main methods:

1. **Square Bracket Notation (`dict[key]`):** Directly accesses the value. However, if the key does not exist, it raises a `KeyError`.

2. **The `get()` Method (`dict.get(key)`):** Accesses the value but handles missing keys gracefully. By default, it returns `None` (or a user-specified default) instead of crashing the program.

Solution:

1. We are given a dictionary `D = {"ID": 101, "Name": "Amit"}`.
2. The statement is `print(D.get("Age"))`.
3. We look for the key `"Age"` in the dictionary.
4. The dictionary only contains the keys `"ID"` and `"Name"`. The key `"Age"` is missing.
5. Because the `get()` method is used, Python does not raise an error. Instead, it checks for a second argument in the function call (the default value).
6. Since no default value was provided (e.g., `D.get("Age", 0)`), the method returns the built-in constant `None`.
7. If the code had been `print(D["Age"])`, it would have resulted in a `KeyError`.
8. Therefore, the output is `None`.

Final Answer: `None`

Answer: (C)

Q34.

Solution

Concept:

File handling in Python allows programs to interact with external data stored on the disk.

When opening a file using the `open(filename, mode)` function, the ”mode” determines what permissions the program has and where the ”file pointer” is placed.

Common modes include:

- `'r'`: Read-only. The pointer is at the beginning.
- `'w'`: Write-only. Overwrites the file or creates a new one.
- `'a'`: Append. Adds data to the end of the file.

Solution:

1. The question asks for the mode that ”overwrites the existing content” if the file already exists.
2. The `'w'` (write) mode is specifically designed for this. When a file is opened in `'w'` mode, if it already exists, its current contents are deleted (truncated) immediately.
3. If the file does not exist, `'w'` mode creates a new, empty file.
4. Let's compare this with `'a'` (append) mode. The `'a'` mode keeps the existing data and moves the file pointer to the very end so new data can be added without losing what was already there.
5. The `'r+'` mode allows both reading and writing, but it does not automatically overwrite the entire file upon opening; it starts at the beginning and replaces character by character.
6. Since the requirement is to completely replace existing content, `'w'` is the correct mode.

Final Answer: `'w'`

Answer: (C)

Q35.

Solution

Concept:

When reading data from a text file in Python, there are three primary methods available through the file object:

1. `read()`: Reads the entire file content as a single string.
2. `readline()`: Reads one single line from the file at a time.
3. `readlines()`: Reads all the lines in the file and returns them as a list, where each element is a string representing a single line (including the newline character
`n`).

Solution:

1. The goal is to return the content of the file "as a list of lines."
2. The `readlines()` method is the standard way to achieve this. It iterates through the file from the current pointer to the end, splitting the text at each line break.
3. For example, if the file contains:
Line 1
Line 2
4. The `readlines()` method would return: `['Line 1
n', 'Line 2
n']`.
5. This is highly useful when you want to process a file line by line using a `for` loop or access a specific line by its index (e.g., `lines[0]`).
6. The `readline()` method (singular) only gives the first line as a string, while `read()` gives everything as one massive block of text.
7. Thus, `readlines()` is the correct answer for list-based retrieval.

Final Answer: `readlines()`

Answer: (C)

Q36.

Solution

Concept:

When working with files in Python, the system maintains a ”file pointer” (or cursor) that indicates the current position within the file where the next read or write operation will occur.

To manage this pointer, Python provides two essential methods:

1. `tell()`: This method returns an integer representing the current position of the file pointer in terms of bytes from the beginning of the file.

2. `seek(offset, from_where)`: This method is used to move the file pointer to a new location.

Understanding the difference between ”asking” where the pointer is and ”telling” the pointer where to go is key to file manipulation.

Solution:

1. The question asks for the method that ”returns the current position of the file pointer.”

2. The `tell()` method is the correct tool for this task. For example, if you have read 10 bytes from a file, calling `file_object.tell()` will return 10.

3. Let's examine the other options to avoid confusion:

- `seek()` is used to move the pointer, not to report its current location.
- `locate()` is not a standard built-in method for file objects in Python.
- `getpos()` is also not a standard file method (though similar naming exists in other specific libraries, it is not part of the core Python file API).

4. Therefore, whenever you need to bookmark a spot in a file or simply track progress, `tell()` is the function you invoke.

Final Answer: `tell()`

Answer: (C)

Q37.

Solution

Concept:

CSV (Comma Separated Values) files are a standard format for storing tabular data in plain text. Python's built-in `csv` module provides specialized tools to handle the complexities of this format, such as handling commas within quotes or different types of line endings.

To read data from a CSV file, we typically follow these steps:

1. Open the file using the standard `open()` function.
2. Pass the file object to the `csv.reader()` function.

The `csv.reader()` function returns a reader object which can be iterated over to access each row as a list of strings.

Solution:

1. The question asks which module must be imported to work with CSV files in Python.
2. The name of the built-in library dedicated to this task is simply `csv`.
3. Therefore, the statement required at the top of your script is `import csv`.
4. Let's look at why other options might be confusing:

- `pandas` is a very powerful third-party library that can handle CSVs, but it is not a "standard" built-in module mentioned in core Python exam syllabi like the `csv` module is.
- `excel` is not a Python module for general CSV handling.
- `file` is a generic concept, not a module name.

5. Since `csv` is part of the Python Standard Library, it is the intended answer for handling delimited text files.

Final Answer: `csv`

Answer: (B)

Q38.

Solution

Concept:

Exception handling is a critical aspect of robust programming. It allows a developer to anticipate potential errors (exceptions) and provide a graceful way to handle them without the entire program crashing.

The structure consists of several blocks:

1. `try`: The block where you put the code that might cause an error.
2. `except`: The block that runs if an error occurs in the try block.
3. `else`: The block that runs **only if no exceptions** were raised in the try block.
4. `finally`: The block that runs no matter what happened (error or no error).

Solution:

1. The question specifically asks for the block that executes only when "no exception occurs."
2. The `else` block is designed for this specific logic. It is useful for code that should only run if the `try` block was successful, separating the "success" logic from the "operation" logic.
3. Let's contrast this with the other blocks:

- The `try` block always starts execution.
- The `except` block only runs if there IS an error.
- The `finally` block runs regardless of the outcome (even if there is a `return` statement).

4. Example:

```
try: x = 1/1
except: print("Error")
else: print("Success")
```

In this case, "Success" will be printed.

5. Therefore, the `else` block is the correct answer.

Final Answer: `else`

Answer: (C)

Q39.

Solution

Concept:

A `ZeroDivisionError` is a specific type of built-in exception in Python.

It occurs when the interpreter encounters a division or modulo operation where the divisor (the denominator) is zero.

Mathematically, division by zero is undefined, and in computing, it would lead to infinite loops or processor errors if not caught.

Python catches this at runtime and raises this exception to stop the program from continuing with an invalid state.

Solution:

1. Let's evaluate the code provided:

```
a = 10
b = 0
print(a / b)
```

2. When the interpreter reaches the third line, it attempts to divide 10 by 0.
3. Since b is exactly 0, the operation `10 / 0` cannot be completed.
4. Python immediately stops the execution of the `try` block (if one existed) and raises a `ZeroDivisionError`.
5. If this code is run without any exception handling, the program will terminate and display a traceback ending in `ZeroDivisionError: division by zero`.
6. Note that other errors like `ValueError` (wrong type content) or `TypeError` (wrong data type) do not apply here because the numbers themselves are valid integers; only the value of the divisor is the problem.

Final Answer: `ZeroDivisionError`

Answer: (A)

Q40.

Solution

Concept:

In the context of exception handling, the `finally` block serves a unique and critical role.

It is used to define ”cleanup” actions that must be executed under all circumstances.

Common use cases include closing database connections, releasing network sockets, or closing open file handles.

Because the code in the `finally` block is guaranteed to run, it ensures that resources are not left hanging even if an unexpected error occurs or the program exits early.

Solution:

1. The question asks for the block that ”always executes,” whether an exception occurred or not.
2. The `finally` block is the only block in the Python `try-except` suite that has this guarantee.
3. If an exception happens and is handled, `finally` runs.
4. If an exception happens and is NOT handled, `finally` runs before the program crashes.
5. If no exception happens at all, `finally` still runs.
6. Even if you use a `break`, `continue`, or `return` statement inside the `try` or `except` blocks, the `finally` block will still be executed before the flow of control leaves that structure.
7. This makes it the most reliable place for essential termination logic.

Final Answer: `finally`

Answer: (C)

Q41.

Solution

Concept:

In computer networking, data transmission can occur in different ”modes” which define the direction of the signal flow between two connected devices.

1. **Simplex:** Data travels in only one direction (e.g., traditional television broadcasting).

2. **Half-Duplex:** Data can travel in both directions, but not at the same time. Only one party can transmit at a time (e.g., a Walkie-Talkie).

3. **Full-Duplex:** Data can travel in both directions simultaneously. Both parties can send and receive at the same time (e.g., a modern telephone conversation).

Solution:

1. The question describes a communication system where data can move in both directions, but ”only one at a time.”

2. This is the classic definition of **Half-Duplex** transmission.

3. Consider a narrow bridge where cars can go both ways, but there is only enough room for one car to pass at a time. This represents the constraints of a half-duplex channel.

4. In a half-duplex environment, the full bandwidth of the communication channel is used by whichever device is currently transmitting. Once it finishes, the other device can take control of the channel.

5. This differs from Simplex because the receiver can send a response, and it differs from Full-Duplex because the send/receive actions cannot overlap in time.

6. Common examples include Walkie-Talkies and older Ethernet hubs (using CSMA/CD).

7. Therefore, the correct answer is Half-Duplex.

Final Answer: Half-Duplex

Answer: (B)

Q42.

Solution

Concept:

Transmission media are the physical pathways used to carry information from a sender to a receiver in a network. These are broadly classified into two categories:

1. **Guided Media:** Use physical cables or wires (e.g., Twisted Pair, Coaxial Cable, Fiber Optic).

2. **Unguided Media:** Use wireless signals to transmit data through the air, vacuum, or water (e.g., Radio waves, Microwaves, Infrared).

Solution:

1. We need to identify which of the listed options is an example of "Unguided Media."

2. **Twisted Pair Cable:** This is a physical copper wire used in standard LAN cables. It is "Guided."

3. **Coaxial Cable:** This is a physical cable with a central conductor, often used for cable TV. It is "Guided."

4. **Fiber Optic Cable:** This uses glass or plastic threads to transmit data as light pulses. It is "Guided."

5. **Microwaves:** These are electromagnetic waves that travel through the atmosphere without the need for a physical wire or cable path. This is a form of wireless communication.

6. Because microwaves do not use a solid, physical conductor to "guide" the signal from point A to point B, they are classified as Unguided Media.

7. This technology is frequently used for satellite communication and long-distance line-of-sight ground communication.

Final Answer: Microwaves

Answer: (B)

Q43.

Solution

Concept:

The TCP/IP model (and the OSI model) uses a layered architecture to manage network communications. Each layer has specific protocols designed for different tasks.

At the Application Layer, protocols define how applications on different systems interact with each other.

1. **HTTP (Hypertext Transfer Protocol):** Used for web browsing.
2. **SMTP (Simple Mail Transfer Protocol):** Used specifically for sending electronic mail (email).
3. **FTP (File Transfer Protocol):** Used for moving files between a client and a server.
4. **DNS (Domain Name System):** Translates human-readable domain names into IP addresses.

Solution:

1. The question asks for the protocol used specifically for "sending emails."
2. **SMTP** stands for Simple Mail Transfer Protocol. Its primary function is to push emails from a client to a mail server, or from one mail server to another mail server.
3. It is important to distinguish this from protocols used for *receiving* or *retrieving* emails, such as POP3 (Post Office Protocol) or IMAP (Internet Message Access Protocol).
4. While HTTP can be used to access "Webmail" (like Gmail in a browser), the actual underlying protocol that moves the mail across the internet is SMTP.
5. FTP is for files, and HTTP is for hypermedia (web pages).
6. Therefore, SMTP is the standard protocol for electronic mail transmission.

Final Answer: SMTP

Answer: (C)

Q44.

Solution

Concept:

A **Switch** is a fundamental networking device that operates at the Data Link Layer (Layer 2) of the OSI model.

It is significantly more ”intelligent” than a Hub. While a hub simply broadcasts every incoming data packet to every single port, a switch learns which device is connected to which port.

It does this by maintaining a MAC Address Table (also called a CAM table). When a frame arrives, the switch looks at the destination MAC address and sends the data only to the specific port where that device is connected.

Solution:

1. The question describes a device that ”forwards data only to the intended recipient node” instead of broadcasting it.

2. This specific behavior—unicasting data based on hardware addresses—is the hallmark of a **Switch**.

3. By only sending data where it needs to go, a switch:

- Reduces unnecessary network traffic.
- Increases security (other nodes cannot sniff the traffic).
- Reduces collisions (each port is its own collision domain).

4. Let’s look at the alternatives:

- **Hub:** This is a ”dumb” device that broadcasts everything to everyone.
- **Repeater:** This simply amplifies a signal; it doesn’t look at addresses or make forwarding decisions.
- **Router:** This operates at Layer 3 (IP addresses) and connects different networks; a switch is usually used within a single network.

5. Thus, the device that provides targeted data delivery is a Switch.

Final Answer: Switch

Answer: (C)

Q45.

Solution

Concept:

A URL (Uniform Resource Locator) is a global address used to locate resources on the World Wide Web. A standard URL consists of several distinct parts:

1. **Protocol (Scheme):** Tells the browser how to communicate (e.g., `https://`).
2. **Domain Name (Host):** The address of the server (e.g., `www.google.com`).
3. **Port:** The technical gate used on the server (usually hidden, e.g., `:443` for HTTPS).
4. **Path:** The specific location of the file or folder on that server (e.g., `/search`).
5. **Query Parameters:** Extra data passed to the server (e.g., `?q=test`).

Solution:

1. In the provided example `https://www.example.com/index.html`, we need to identify the "domain name" part.
2. The sequence starts with `https://`, which is the **Protocol**.
3. The next part, `www.example.com`, identifies the specific organization or entity that owns the website. This is the **Domain Name**.
4. The final part, `/index.html`, refers to the specific file being requested from the server's storage. This is the **Path**.
5. Domain names are easier for humans to remember than numerical IP addresses (like 93.184.216.34), and the DNS system handles the translation between the two.
6. Therefore, `www.example.com` is the correct segment for the domain name.

Final Answer: `www.example.com`

Answer: (B)

Q46.

Solution

Concept:

Network security is a broad term that covers a multitude of technologies, devices, and processes.

A **Firewall** is one of the most fundamental components of a secure network architecture.

Its primary role is to act as a barrier between a trusted internal network (like a company's private LAN) and an untrusted external network (like the public Internet).

Firewalls function by inspecting incoming and outgoing traffic based on a set of predefined security rules.

They can filter data based on IP addresses, port numbers, or specific protocols to prevent unauthorized access and cyberattacks.

Solution:

1. The question describes a system "designed to prevent unauthorized access to or from a private network."
2. This is the precise definition and purpose of a **Firewall**.
3. Firewalls can be implemented in two ways:

- **Hardware-based:** Dedicated physical devices that sit between the router and the rest of the network.
- **Software-based:** Programs installed on individual computers (like the Windows Defender Firewall) to protect that specific machine.

4. While a **Router** directs traffic and a **Switch** connects devices, they do not inherently provide the security filtering that a firewall does.
5. An **Antivirus** program protects a computer by scanning for malicious software (malware) already on the system, whereas a firewall stops the threat from entering the network in the first place.
6. Therefore, the Firewall is the correct security-focused mechanism described.

Final Answer: Firewall

Answer: (C)

Q47.

Solution

Concept:

As data travels across a network, it is susceptible to corruption due to electrical interference, noise, or hardware malfunctions.

To ensure that the data received is identical to the data sent, error detection techniques are used.

One of the simplest and most common methods is the **Parity Bit** method.

A parity bit is an extra bit added to a string of binary code to ensure that the total number of 1-bits in the string is either even or odd.

1. **Even Parity:** The parity bit is set so that the total number of 1s is an even number.
2. **Odd Parity:** The parity bit is set so that the total number of 1s is an odd number.

Solution:

1. The question asks for the name of the ”extra bit added to a data word to detect errors.”
2. This is called a **Parity Bit**.
3. For example, consider the 7-bit binary data: `1101001`.

- There are four 1s (an even number).
- If using ”Even Parity,” the parity bit would be `0` (keeping the total at 4).
- If using ”Odd Parity,” the parity bit would be `1` (making the total 5).

4. If a single bit flips during transmission (e.g., a 0 becomes a 1), the receiver will notice that the parity is no longer even (or odd), signaling an error.
5. Other options like **Checksum** and **CRC (Cyclic Redundancy Check)** are more complex methods used for larger blocks of data, but the ”single extra bit” specifically refers to parity.
6. Therefore, Parity Bit is the correct answer.

Final Answer: Parity Bit

Answer: (B)

Q48.

Solution

Concept:

An **IP Address (Internet Protocol Address)** is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.

There are two versions of IP addresses currently in use:

1. **IPv4:** The older, more common version. It consists of 32 bits, usually written as four decimal numbers separated by dots (e.g., 192.168.1.1).

2. **IPv6:** The newer version developed to provide more addresses as the world runs out of IPv4 options. It consists of 128 bits, usually written in hexadecimal.

Solution:

1. The question asks for the length of an **IPv4 address** in bits.

2. An IPv4 address is divided into four sections called "octets."

3. Each octet contains 8 bits (2^8 = 256 possible values, ranging from 0 to 255).

4. Calculation: 4 octets $\times$ 8 bits per octet = 32 bits.

5. Let's contrast this with other address lengths:

- 48 bits = MAC Address.
- 64 bits = Common in some specialized addressing or hardware architectures.
- 128 bits = IPv6 Address.

6. Because IPv4 is limited to 32 bits, it can provide approximately 4.3 billion unique addresses, which is why the transition to the 128-bit IPv6 was necessary.

7. Thus, the standard length for IPv4 is 32 bits.

Final Answer: 32 bits

Answer: (A)

Q49.

Solution

Concept:

The World Wide Web (WWW) relies on several core technologies to function, including HTML for content structure and HTTP for data transfer.

HTML (HyperText Markup Language) is the standard language used to create the pages we see in web browsers.

HTML uses ”tags” to define elements like headings, paragraphs, links, and images.

It is important to understand that HTML is a *markup* language, not a *programming* language, as it focuses on the structure and presentation of data rather than logic and computations.

Solution:

1. The question asks for the full form of the acronym **HTML**.
2. The correct expansion is **HyperText Markup Language**.
3. Let’s break down the meaning:

- **HyperText:** Refers to text that contains links to other texts (hyperlinks).
- **Markup:** Refers to the use of tags to annotate text so that the computer knows how to display it.
- **Language:** Refers to the set of rules and syntax used for the code.

4. Let’s evaluate the incorrect options:

- ”Hyperlink Markup Language” is wrong because the ’H’ stands for Hypertext.
- ”High Tech Markup Language” is a common but incorrect guess.
- ”HyperText Management Language” is also incorrect.

5. Since its creation by Tim Berners-Lee, HTML has evolved through many versions, with HTML5 being the current standard used today.

Final Answer: HyperText Markup Language

Answer: (A)

Q50.

Solution

Concept:

Cookies are small pieces of data—typically text files—that a web server sends to a user's web browser while the user is browsing a website.

When the user visits the same website in the future, the browser sends the cookie back to the server.

Cookies serve several important functions:

1. **Session Management:** Keeping track of whether a user is logged in.
2. **Personalization:** Remembering user preferences, themes, or settings.
3. **Tracking:** Recording user behavior for analytics or targeted advertising.

Solution:

1. The question asks for the term describing "small files stored on a user's computer by a website to remember information."
2. This is the exact definition of a **Cookie**.
3. Let's analyze why other terms are incorrect:

- **Cache:** This refers to temporary storage of web assets (like images or scripts) to speed up page loading. It doesn't "remember" user-specific data in the same way.
- **Spam:** This refers to unsolicited or junk email.
- **History:** This is a list of URLs visited by the user, stored locally by the browser, but it is not a "file sent by a website" to store data.

4. Cookies can be "session-based" (deleted when the browser closes) or "persistent" (remaining for a set period).
5. Because cookies are the primary tool used by websites to "recognize" a returning visitor and their specific data, they are the correct answer.

Final Answer: Cookies

Answer: (B)

Answer Key

Q	Ans	Q	Ans	Q	Ans	Q	Ans	Q	Ans
1	B	2	C	3	A	4	D	5	B
6	B	7	C	8	B	9	B	10	B
11	A	12	B	13	C	14	C	15	C
16	A	17	B	18	A	19	C	20	A
21	C	22	A	23	B	24	A	25	C
26	C	27	D	28	C	29	A	30	B
31	B	32	B	33	C	34	C	35	C
36	C	37	B	38	C	39	A	40	C
41	B	42	B	43	C	44	C	45	B
46	C	47	B	48	A	49	A	50	B