

# CUET-UG Information Practices Sample Paper-12

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.** What will be the output of the following SQL query?

```
SELECT ROUND(154.678, -2), TRUNCATE(154.678, -1);
```

- (A) 155, 150
- (B) 200, 150
- (C) 154, 154
- (D) 200, 154.6

**Q2.** Consider a string column Category with the value 'Mathematics'. Which command will return 'ematics'?

- (A) `SELECT SUBSTR('Mathematics', 4);`
- (B) `SELECT MID('Mathematics', -7);`
- (C) `SELECT RIGHT('Mathematics', 7);`
- (D) Both (A) and (C)

**Q3.** A table 'Activity' has a column EventDate with the value '2024-02-29'. What is the result of `SELECT MONTHNAME(ADDDATE(EventDate, INTERVAL 1 YEAR));`?

- (A) February
- (B) March
- (C) NULL



(D) Error

**Q4.** Identify the correct SQL function that returns the position of the first occurrence of a substring 'at' in the string 'Informatics'?

(A) SEARCH('at', 'Informatics')

(B) INSTR('Informatics', 'at')

(C) CHARINDEX('at', 'Informatics')

(D) POSITION('Informatics' IN 'at')

**Q5.** What is the output of `SELECT MOD(LENGTH('CUET2024'), 3);`?

(A) 0

(B) 1

(C) 2

(D) 8

**Q6.** Which function is used to return the current system date and time simultaneously?

(A) CURDATE()

(B) SYSDATE()

(C) DATE()

(D) SYSTEMTIME()

**Q7.** What will be the value of `SELECT LTRIM(' Data Science ');`?

(A) 'Data Science'

(B) 'Data Science '

(C) ' Data Science'

(D) 'DataScience'

**Q8.** The result of `SELECT DAYOFYEAR('2024-03-01');` is (Note: 2024 is a leap year):



- (A) 60
- (B) 61
- (C) 59
- (D) 62

**Q9.** In Relational Algebra, which operation is used to extract specific columns from a relation while removing duplicate rows?

- (A)  $\sigma$  (Selection)
- (B)  $\rho$  (Renaming)
- (C)  $\pi$  (Projection)
- (D)  $\bowtie$  (Join)

**Q10.** A table has attributes (A, B, C, D). If {A, B} and {A, C} can both uniquely identify a tuple, then B and C are parts of:

- (A) Primary Keys
- (B) Candidate Keys
- (C) Foreign Keys
- (D) Alternate Keys

**Q11.** The 'Selection' operator in relational algebra is equivalent to which SQL clause?

- (A) SELECT
- (B) WHERE
- (C) FROM
- (D) GROUP BY

**Q12.** Which constraint prevents the entry of duplicate values in a column but allows a single NULL value?

- (A) PRIMARY KEY
- (B) UNIQUE



- (C) NOT NULL
- (D) CHECK

**Q13.** In which network topology is every node connected to a central switch or hub, making it easy to add new nodes but creating a single point of failure?

- (A) Mesh
- (B) Bus
- (C) Star
- (D) Ring

**Q14.** A device 'X' receives data signals, cleans them, and re-transmits them to cover longer distances without data loss. 'X' is a:

- (A) Gateway
- (B) Repeater
- (C) Router
- (D) Bridge

**Q15.** Identify the correct statement regarding MAC and IP addresses:

- (A) MAC address is logical; IP address is physical.
- (B) MAC address is 48-bit; IP address (IPv4) is 32-bit.
- (C) Both change every time a computer connects to a new network.
- (D) MAC address works at the Network Layer.

**Q16.** Which code snippet will create a Series 'S' with values [10, 20, 30] and custom indices ['x', 'y', 'z']?

- (A) `S = pd.Series([10, 20, 30], index=['x', 'y', 'z'])`
- (B) `S = pd.Series(['x', 'y', 'z'], [10, 20, 30])`
- (C) `S = pd.Series(index=['x', 'y', 'z'], data=[10, 20, 30])`
- (D) Both (A) and (D)



- Q17.** To access the first 3 rows and specific columns 'Name' and 'Score' from a DataFrame `df`, the correct `loc` syntax is:
- (A) `df.loc[0:2, ['Name', 'Score']]`
  - (B) `df.loc[0:3, ['Name', 'Score']]`
  - (C) `df.iloc[0:3, ['Name', 'Score']]`
  - (D) `df.loc[1:3, 'Name':'Score']`
- Q18.** What does the attribute `df.size` return for a DataFrame with 5 rows and 4 columns?
- (A) 5
  - (B) 4
  - (C) 20
  - (D) (5, 4)
- Q19.** Which method is used to read a CSV file where the data is separated by a tab rather than a comma?
- (A) `pd.read_csv(file, sep='t')`
  - (B) `pd.read_table(file)`
  - (C) `pd.read_csv(file, delimiter='tab')`
  - (D) Both (A) and (B)
- Q20.** How can you check if any value in a Series 'S' is NULL?
- (A) `S.empty()`
  - (B) `S.isnull().any()`
  - (C) `S.isna().sum()`
  - (D) `S.count()`
- Q21.** What is the output of `df.columns`?



- (A) A list of all row values.
- (B) An Index object containing the column labels.
- (C) The total count of columns.
- (D) The data types of all columns.

**Q22.** To rename a column 'OldName' to 'NewName' permanently in df:

- (A) `df.rename(columns={'OldName': 'NewName'}, inplace=True)`
- (B) `df.rename({'OldName': 'NewName'})`
- (C) `df.columns['OldName'] = 'NewName'`
- (D) `df.set_index('NewName')`

**Q23.** If S is a Series with index [0, 1, 2, 3], what does S[1:3] return?

- (A) Values at index 1 and 2.
- (B) Values at index 1, 2, and 3.
- (C) Values at index 0, 1, and 2.
- (D) An error.

**Q24.** Which property returns the transpose of a DataFrame?

- (A) `df.transpose`
- (B) `df.T`
- (C) `df.swap()`
- (D) `df.pivot()`

**Q25.** The `df.head(n)` method returns the first n rows. If n is not specified, it returns:

- (A) 1 row
- (B) 10 rows
- (C) 5 rows
- (D) All rows



- Q26.** Which method is used to delete a row with index label 'R1' from a DataFrame?
- (A) `df.remove('R1')`
  - (B) `df.drop('R1', axis=0)`
  - (C) `df.delete('R1')`
  - (D) `df.drop('R1', axis=1)`

- Q27.** What is the data type of `df.values`?
- (A) A Python List
  - (B) A NumPy ndarray
  - (C) A Pandas Series
  - (D) A Dictionary

- Q28.** To find the mean of only the 'Salary' column in `df`, we use:
- (A) `df.mean('Salary')`
  - (B) `df['Salary'].mean()`
  - (C) `df.Salary.avg()`
  - (D) `pd.mean(df['Salary'])`

Which function is used to export a DataFrame to a CSV file without the index column?

- Q29.** (A) `df.to_csv('file.csv', index=False)`  
(B) `df.write_csv('file.csv')`  
(C) `df.save('file.csv', no_index=True)`  
(D) `df.to_csv('file.csv', header=None)`
- Q30.** What happens if you perform `S1 + S2` where `S1` and `S2` have non-matching indices?
- (A) It results in an error.
  - (B) It ignores the non-matching indices.



- (C) It results in NaN for the non-matching indices.
- (D) It concatenates the two Series.

**Q31.** How do you access the element in the 2nd row and 3rd column of df using integer indexing?

- (A) `df.iloc[2, 3]`
- (B) `df.iloc[1, 2]`
- (C) `df.loc[1, 2]`
- (D) `df.iat[2, 3]`

**Q32.** Which Pandas function is used to fill all NaN values in a DataFrame with 0?

- (A) `df.replace(NaN, 0)`
- (B) `df.fillna(0)`
- (C) `df.dropna(0)`
- (D) `df.null(0)`

**Q33.** Which clause is used to filter the results of a GROUP BY query?

- (A) WHERE
- (B) HAVING
- (C) LIKE
- (D) ORDER BY

**Q34.** To arrange the data in descending order of 'Marks' and then ascending order of 'Name', the clause is:

- (A) `ORDER BY Marks DESC, Name ASC`
- (B) `SORT BY Marks, Name`
- (C) `ORDER BY Marks, Name DESC`
- (D) `GROUP BY Marks DESC, Name`

**Q35.** A 'Natural Join' is a type of Join that:



- (A) Join tables based on all columns with the same name.
- (B) Combines all rows of one table with all rows of another.
- (C) Requires an explicit ON clause.
- (D) Returns only unmatched rows.

**Q36.** What is the output of COUNT(\*) if a table has 10 rows and one column has 3 NULL values?

- (A) 7
- (B) 10
- (C) 3
- (D) Error

**Q37.** Which SQL Join returns all rows from the right table, and the matched rows from the left table?

- (A) INNER JOIN
- (B) LEFT JOIN
- (C) RIGHT JOIN
- (D) CROSS JOIN

**Q38.** To find the total number of departments in an 'Employee' table, the correct query is:

- (A) SELECT COUNT(Dept) FROM Employee;
- (B) SELECT COUNT(DISTINCT Dept) FROM Employee;
- (C) SELECT DISTINCT(COUNT Dept) FROM Employee;
- (D) SELECT SUM(Dept) FROM Employee;

**Q39.** The HAVING clause cannot be used without:

- (A) WHERE
- (B) GROUP BY



(C) ORDER BY

(D) JOIN

**Q40.** Which aggregate function ignores NULL values during calculation?

(A) AVG()

(B) SUM()

(C) MAX()

(D) All of the above

**Q41.** If Table A has 4 rows and Table B has 3 rows, how many rows will be in their Cartesian Product?

(A) 7

(B) 12

(C) 1

(D) 4

**Q42.** Which function is used to save a plot as an image file?

(A) `plt.show()`

(B) `plt.save()`

(C) `plt.savefig()`

(D) `plt.print()`

**Q43.** To create a horizontal bar chart, the function used is:

(A) `plt.bar()`

(B) `plt.barh()`

(C) `plt.hbar()`

(D) `plt.plot(kind='barh')`

**Q44.** What is the purpose of the `label` argument in `plt.plot()`?



- (A) To set the X-axis title.
- (B) To provide a name for the data series to be used in the legend.
- (C) To label individual data points.
- (D) To title the entire chart.

**Q45.** Which parameter in `plt.pie()` allows you to highlight or separate a slice from the rest of the chart?

- (A) `shadow`
- (B) `autopct`
- (C) `explode`
- (D) `colors`

**Q46.** To add a grid to the background of a plot, we use:

- (A) `plt.grid(True)`
- (B) `plt.show_grid()`
- (C) `plt.background('grid')`
- (D) `plt.draw(grid=True)`

**Q47.** The 'Right to be Forgotten' is most closely related to which concept?

- (A) Intellectual Property Rights
- (B) Digital Footprint
- (C) E-waste Management
- (D) Open Source Software

**Q48.** Which of the following is an example of 'Phishing'?

- (A) Using a friend's Netflix password with their consent.
- (B) Receiving an email that looks like a legitimate bank notification to capture your PIN.
- (C) Posting a copyrighted video on social media.



(D) Discarding an old smartphone in the trash.

**Q49.** The unauthorized copying and distribution of software is known as:

(A) Software Piracy

(B) Plagiarism

(C) Hacking

(D) Cyber Stalking

**Q50.** Which toxic heavy metal is commonly found in E-waste and can cause damage to the human nervous system?

(A) Iron

(B) Mercury

(C) Aluminum

(D) Magnesium



**Detailed Solutions****Q1.****Solution****Concept:**

- **ROUND**( $n$ ,  $d$ ): Rounds to  $d$  decimal places. If  $d$  is negative, it rounds to the left of the decimal point (tens, hundreds, etc.).
- **TRUNCATE**( $n$ ,  $d$ ): Cuts off the number at  $d$  decimal places without rounding.

**Step-by-Step Calculation:**

- **ROUND(154.678, -2):**
  - $-2$  indicates rounding to the nearest **hundred**.
  - The value at the tens place is 5. Since it is  $\geq 5$ , we round up.
  - Result = 200.
- **TRUNCATE(154.678, -1):**
  - $-1$  indicates removing digits to the left of the decimal point (the units place).
  - It replaces the units digit with 0 without checking the value.
  - Result = 150.

**Explanation Table:**

Function	Position	Result
ROUND(-2)	Hundreds	200
TRUNCATE(-1)	Tens	150

**Final Answer:** 200, 150**Answer: (B)**

Q2.

### Solution

**Concept:** String manipulation functions (SUBSTR, MID, RIGHT) extract parts of a string based on index positions or lengths.

**Target String Analysis:**

- **Input:** 'Mathematics' (Length = 11)
- **Target:** 'ematics' (Length = 7)

**Execution Logic:**

- **Option (A):** SUBSTR('Mathematics', 4) starts at the 4th character ('h') and goes to the end. Result: 'hematics'. (*Incorrect*)
- **Option (B):** MID('Mathematics', -7) uses negative indexing (standard in MySQL). It counts 7 characters from the **right** to find the start position ('e') and extracts to the end. Result: 'ematics'. (*Correct*)
- **Option (C):** RIGHT('Mathematics', 7) extracts exactly 7 characters from the end of the string. Result: 'ematics'. (*Correct*)

**Position Mapping:**

Char	M	a	t	h	e	m	a	t	i	c	s
Pos	1	2	3	4	5	6	7	8	9	10	11
-Idx	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

**Final Answer:** Both (B) and (C) are technically correct in MySQL environments.

**Answer: (D)**



Q3.

**Solution****Concept:** Handling Leap Year dates when adding intervals.

- `ADDDATE(date, INTERVAL)`: Adds time to a date.
- `MONTHNAME(date)`: Returns the full name of the month for a given date.

**Logic Step-by-Step:**

- **Input Date:** 2024-02-29 (Leap Day).
- **Interval Addition:** Adding 1 Year to 2024-02-29 results in 2025-02-28.
- **Reasoning:** Since 2025 is not a leap year, the date 2025-02-29 does not exist. SQL automatically adjusts to the last valid day of the month.
- **Extraction:** `MONTHNAME('2025-02-28')` returns 'February'.

**Comparison:**

Operation	Result
Original Date	2024-02-29
+ 1 Year Interval	2025-02-28
<code>MONTHNAME()</code>	February

**Final Answer:** February**Answer: (A)**

Q4.

**Solution**

**Concept:** SQL provides functions to find the position of a substring within a string. Common functions include:

- `INSTR(string, substring)` – Returns the position of the first occurrence of `substring` in `string` (used in Oracle, MySQL).
- `CHARINDEX(substring, string)` – SQL Server equivalent of `INSTR`.
- `POSITION(substring IN string)` – Standard SQL function.

**Analysis of Options:**

- (a) `SEARCH('at', 'Informatics')` Not a standard SQL function.
- (b) `INSTR('Informatics', 'at')` Correct for MySQL/Oracle.
- (c) `CHARINDEX('at', 'Informatics')` Correct in SQL Server.
- (d) `POSITION('Informatics' IN 'at')` Incorrect syntax; the order is reversed.

**Calculation:**

- String: I n f o r m a t i c s
- Positions (1-indexed): 1 2 3 4 5 6 7 8 9 10 11
- Substring 'at' starts at position 7

**Final Answer:** `INSTR('Informatics', 'at')`

**Answer: (B)**

Q5.

**Solution**

**Concept:** The `LENGTH()` function returns the number of characters in a string. The `MOD(a, b)` function returns the remainder when `a` is divided by `b`.

**Step-by-step Calculation:**

- String: 'CUET2024' → Number of characters = 8
- $\text{MOD}(8, 3) = \text{remainder when } 8 \text{ is divided by } 3$

**Digit-by-digit Division:**

$3 \times 2 = 6$  (remainder 8)  $6 = 2$

**Final Answer:** `2`

**Answer: (C)**



Q6.

**Solution****Concept:** SQL provides different functions to retrieve date and time:

- CURDATE() – Returns only the current date.
- SYSDATE() – Returns the current date and time at the moment of execution.
- DATE() – Extracts the date part from a datetime value.
- SYSTEMTIME() – Not a standard SQL function.

**Analysis:** We need a function that returns both date and time simultaneously, so:

SYSDATE()

**Final Answer:** **Answer: (B)**

Q7.

**Solution****Concept:**

LTRIM(string) removes leading spaces from a string. Trailing spaces remain unaffected.

**Original string:** ' Data Science '**After applying LTRIM():** 'Data Science '**Option Analysis:**

- 'Data Science' Removes trailing spaces too (not LTRIM).
- 'Data Science ' Correct, only leading spaces removed.
- ' Data Science' Leading spaces are removed by LTRIM.
- 'DataScience' Spaces inside the text are untouched.

**Final Answer:** **Answer: (B)**

Q8.

**Solution**

**Concept:** The DAYOFYEAR(date) function returns the day of the year for a given date, ranging from 1 to 366 (for leap years).

**Step-by-Step Calculation for a Leap Year (2024):**

- **January:** 31 days
- **February:** 29 days (Since 2024 is a leap year, February has 29 days instead of 28)
- **March 1st:** 1 day

**Cumulative Count:**

- $31 \text{ (Jan)} + 29 \text{ (Feb)} = 60 \text{ days}$
- $60 + 1 \text{ (March 1st)} = 61$

**Comparison with Non-Leap Year:**

Month	Non-Leap Year (e.g., 2023)	Leap Year (2024)
January	31	31
February	28	29
March 1st	Day 59	<b>Day 61</b>

**Explanation:** In a leap year, every date from March 1st onwards is shifted by +1 compared to a standard year because of the inclusion of February 29th.

**Final Answer:**

**Answer:** (B)



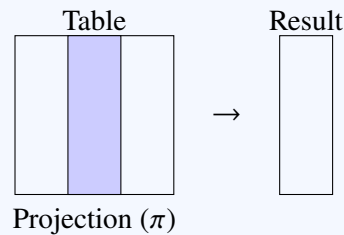
Q9.

**Solution**

**Concept:** Relational algebra operations define how to manipulate relations.

- **Selection ( $\sigma$ ):** Filters rows based on a condition (horizontal).
- **Projection ( $\pi$ ):** Selects specific columns (vertical) and automatically eliminates duplicates.

**Visual Representation:**



**Explanation:** Unlike the SQL SELECT statement (which requires DISTINCT to remove duplicates), the mathematical Projection operator ( $\pi$ ) inherently treats the result as a set, meaning duplicate tuples are removed by definition.

**Final Answer:**  $\pi$  (Projection)

**Answer:** (C)

Q10.

**Solution**

**Concept:** A candidate key is a minimal set of attributes that can uniquely identify each tuple in a relation. A table may have more than one candidate key. If multiple combinations of attributes uniquely identify tuples, each combination is considered a candidate key. One of them may later be selected as the primary key, while the remaining candidate keys become alternate keys.

**Solution:** In the given table, both {A, B} and {A, C} can uniquely identify a tuple. Since each set independently satisfies the condition of uniqueness and minimality, both combinations are candidate keys. Attribute B is part of the candidate key {A, B}, and attribute C is part of the candidate key {A, C}. A primary key is chosen from candidate keys, but the question only states that the combinations uniquely identify tuples, not which one is selected as primary. Therefore, B and C are parts of candidate keys. Foreign keys are used to establish relationships between tables and are unrelated here.

**Final Answer:** Candidate Keys

**Answer:** (B)



Q11.

**Solution**

**Concept:** In Relational Algebra, the Selection operator retrieves tuples that satisfy a specified condition. SQL provides similar functionality through conditional filtering of rows. The purpose of selection is to reduce the number of rows based on a predicate while keeping all columns unchanged. This operation forms the foundation of query filtering in database management systems.

**Solution:** The Selection operator in Relational Algebra is represented by  $\sigma$  and is used to choose rows that satisfy a condition. In SQL, the equivalent operation is performed using the WHERE clause. For example, the relational algebra expression  $\sigma_{Age>18}(Student)$  corresponds to the SQL query `SELECT * FROM Student WHERE Age > 18;`. The SELECT keyword in SQL specifies columns, not filtering conditions. The FROM clause identifies the table, while GROUP BY is used for grouping rows. Hence, the WHERE clause is the direct equivalent of the Selection operator.

**Final Answer:**

**Answer: (B)**

Q12.

**Solution**

**Concept:** Constraints in SQL are rules applied to table columns to maintain data integrity and accuracy. Different constraints perform different functions such as preventing NULL values, enforcing uniqueness, or validating conditions. The UNIQUE constraint specifically ensures that duplicate values are not entered in a column, while still permitting one NULL value in most database systems.

**Solution:** The UNIQUE constraint prevents duplicate values from being inserted into a column. Unlike the PRIMARY KEY constraint, it allows a single NULL value because NULL represents an unknown value rather than an actual duplicate. The PRIMARY KEY constraint does not allow NULL values and also enforces uniqueness. The NOT NULL constraint only prevents NULL entries without checking duplication, while the CHECK constraint validates conditions specified by the user. Since the question asks for a constraint that blocks duplicates but allows one NULL value, the correct answer is UNIQUE.

**Final Answer:**

**Answer: (B)**



Q13.

**Solution**

**Concept:** Network topology refers to the arrangement of nodes and communication links in a computer network. In a star topology, all devices are connected to a central hub or switch. This structure makes management and expansion easy because devices can be added or removed without affecting others. However, the central device becomes a critical component of the network.

**Solution:** In a star topology, every node communicates through a central hub or switch. This arrangement simplifies troubleshooting and allows new nodes to be connected easily without disturbing the rest of the network. If one peripheral node fails, the rest of the network continues functioning normally. However, if the central hub or switch fails, the entire network becomes nonfunctional because all communication depends on that central device. This characteristic is known as a single point of failure. Other topologies differ significantly: mesh provides multiple paths, bus uses a common cable, and ring connects nodes in a circular manner.

**Final Answer:** Star

Answer: (C)

Q14.

**Solution**

**Concept:** Communication devices are used in computer networks to improve signal transmission and connectivity. Over long distances, signals may weaken because of attenuation. Certain devices regenerate or strengthen these signals so that data can travel farther without corruption or loss. Such devices play an important role in extending the range of a network.

**Solution:** A repeater is a networking device that receives weak or distorted signals, cleans and regenerates them, and then retransmits them with renewed strength. Its main purpose is to extend the communication distance in a network while reducing the chances of signal loss. Repeaters operate at the physical layer of the OSI model and do not analyze or route data packets. A router directs packets between networks, a bridge connects network segments, and a gateway translates between different protocols. Since the described device strengthens and retransmits signals over longer distances, the correct identification of device X is a repeater.

**Final Answer:** Repeater

Answer: (B)



Q15.

**Solution**

**Concept:** MAC and IP addresses are both used for communication in computer networks, but they serve different purposes. A MAC address is a physical hardware address assigned to a network interface card and usually remains fixed. An IP address is a logical address assigned to a device for identification on a network. IPv4 addresses are 32-bit long, whereas MAC addresses are 48-bit long.

**Solution:** The correct statement is that a MAC address is 48-bit and an IPv4 address is 32-bit. A MAC address is permanently assigned by the manufacturer and works at the Data Link Layer of the OSI model. It is used for communication within a local network. On the other hand, an IP address operates at the Network Layer and can change depending on the network connection. Option (A) is incorrect because MAC is physical and IP is logical. Option (C) is incorrect because MAC addresses generally do not change with network connections. Option (D) is wrong because MAC addresses function at the Data Link Layer, not the Network Layer.

**Final Answer:** MAC address is 48-bit; IP address (IPv4) is 32-bit.

**Answer: (B)**

Q16.

**Solution**

**Concept:** In Pandas, a Series is a one-dimensional labeled array.

- `data=` specifies the values.
- `index=` specifies custom labels for the values.

**Visual Representation:**

<b>x</b>	10
<b>y</b>	20
<b>z</b>	30

Pandas Series

**Explanation:** Option (A) correctly creates a Series using values and custom indices. Option (C) is also correct because it uses keyword arguments `data=` and `index=`. Hence, both (A) and (C) are valid ways to create the Series.

**Final Answer:** Both (A) and (C)

**Answer: (D)**



Q17.

**Solution**

**Concept:** The `loc[]` function in Pandas is used for label-based indexing.

- Rows are selected using row labels.
- Columns are selected using column names.
- In `loc`, the ending index is included.

**Visual Representation:**

NameScore			
0			
1			
2			

`df.loc[0:2, ['Name', 'Score']]`

**Explanation:** `df.loc[0:2, ['Name', 'Score']]` selects:

- Rows with labels 0, 1, and 2 (first 3 rows)
- Columns Name and Score

Option (B) selects 4 rows because `loc` includes the ending index. Option (C) is invalid because `iloc[]` uses integer column positions, not column names. Option (D) starts from row 1, not the first row.

**Final Answer:** `df.loc[0:2, ['Name', 'Score']]`

**Answer: (A)**

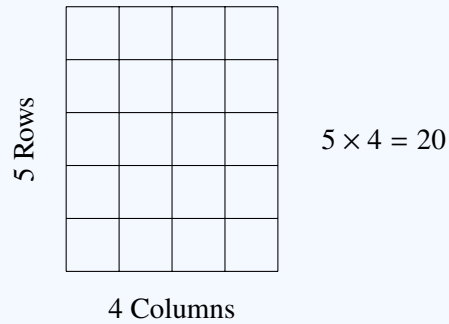
Q18.

**Solution**

**Concept:** In Pandas, the attribute `df.size` returns the total number of elements in a DataFrame.

$$\text{size} = \text{rows} \times \text{columns}$$

**Visual Representation:**



**Explanation:** The DataFrame contains:

- 5 rows
- 4 columns

So, the total number of elements is:

$$5 \times 4 = 20$$

Therefore, `df.size` returns 20.

**Final Answer:**

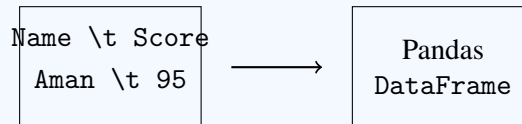
**Answer:** (C)



Q19.

**Solution****Concept:** Pandas can read tab-separated data using either:

- `pd.read_csv()` with the separator set to `\t`
- `pd.read_table()`, which is designed for tab-separated values by default

**Visual Representation:****Explanation:**

- `pd.read_csv(file, sep='\t')` correctly specifies tab as the separator.
- `pd.read_table(file)` also works because it assumes tab-separated data by default.
- `delimiter='tab'` is incorrect because the delimiter should be the tab character `\t`, not the word "tab".

**Final Answer:** Both (A) and (B)**Answer: (D)**

Q20.

**Solution****Concept:** In Pandas, NULL or missing values can be detected using:

- `isnull()` or `isna()` to identify missing values
- `any()` to check whether at least one missing value exists

**Visual Representation:**

10
NULL
25
40

`S.isnull().any()` True

**Explanation:**

- `S.isnull().any()` returns True if at least one NULL value exists in the Series.
- `S.empty()` checks whether the Series is empty, not whether it contains NULL values.
- `S.isna().sum()` counts NULL values but does not directly return a True/False check.
- `S.count()` counts non-NULL values.

**Final Answer:** `S.isnull().any()`**Answer: (B)**

Q21.

**Solution**

**Concept:** In Pandas, a DataFrame stores data in rows and columns similar to a table. The `columns` attribute is used to access the labels of all columns present in the DataFrame. Instead of returning raw data or counts, it provides an Index object containing the names of the columns. This attribute is frequently used for inspecting structure, renaming columns, and performing column-based operations in data analysis tasks.

**Solution:** The expression `df.columns` returns an Index object that contains all column labels of the DataFrame. For example, if a DataFrame has columns named Name, Age, and Marks, then `df.columns` will display something like `Index(['Name', 'Age', 'Marks'], dtype='object')`. It does not return row values, which are accessed differently using methods such as `values` or row indexing. It also does not provide the number of columns directly; that can be obtained using `len(df.columns)` or `df.shape`. Similarly, column data types are accessed through `df.dtypes`. Therefore, the correct description of `df.columns` is that it returns an Index object containing the column labels.

**Final Answer:** An Index object containing the column labels.

**Answer: (B)**

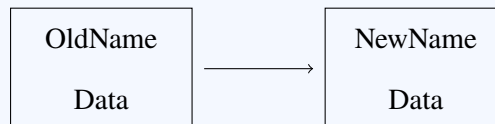


Q22.

**Solution**

**Concept:** In Pandas, column names can be changed using the `rename()` method. To make changes permanent, `inplace=True` is used.

- `rename(columns={})` maps old column names to new ones.
- `inplace=True` modifies the original DataFrame.

**Visual Representation:**

```
rename(columns=..., inplace=True)
```

**Explanation:**

- Option (A) correctly renames the column and updates the DataFrame permanently.
- Option (B) is incomplete because it does not specify `columns=` or `inplace=True`.
- Option (C) is invalid syntax for renaming columns.
- Option (D) changes the index, not column names.

**Final Answer:** `rename(inplace=True)`

**Answer: (A)**



Q23.

**Solution**

**Concept:** Slicing in a Pandas Series follows Python slicing rules.

- The starting index is included.
- The ending index is excluded.

**Visual Representation:**



`S[1:3]`

**Explanation:** `S[1:3]` starts from index 1 and stops before index 3. Therefore, it returns the values at:

1 and 2

Index 3 is not included because slicing excludes the ending position.

**Final Answer:** Values at index 1 and 2

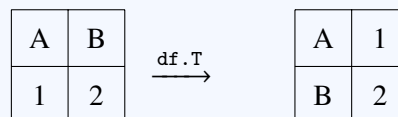
Answer: (A)

Q24.

**Solution**

**Concept:** Transpose of a DataFrame means converting rows into columns and columns into rows. In Pandas, the property `df.T` is commonly used for this operation.

**Visual Representation:**



**Explanation:**

- `df.T` directly returns the transpose of the DataFrame.
- `df.transpose` refers to the transpose method but is usually called as `df.transpose()`.
- `df.swap()` is not a valid transpose method.
- `df.pivot()` reshapes data differently and is not used for transpose.

**Final Answer:** `df.T`

Answer: (B)



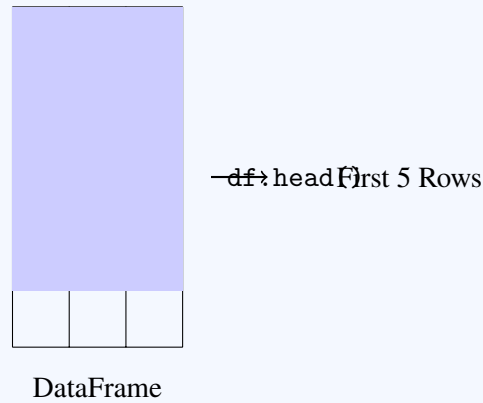
Q25.

**Solution**

**Concept:** The `head()` method in Pandas is used to display the top rows of a DataFrame.

- `df.head(n)` returns the first `n` rows.
- If no value of `n` is given, the default is 5 rows.

**Visual Representation:**



**Explanation:**

- `df.head()` without any argument automatically displays the first 5 rows.
- This default behavior helps quickly preview the dataset.

**Final Answer:** 5 rows

**Answer:** (C)



Q26.

**Solution**

**Concept:** In Pandas, rows and columns of a DataFrame can be removed using the `drop()` method. The parameter `axis` determines whether rows or columns are affected. By convention, `axis=0` refers to rows and `axis=1` refers to columns. Deleting rows by index labels is a common operation during data cleaning and preprocessing when unnecessary or duplicate records must be removed from a dataset.

**Solution:** To remove a row with the index label 'R1', the correct method is `df.drop('R1', axis=0)`. The `drop()` function deletes labels from a DataFrame, and `axis=0` specifies that the operation should target rows. If `axis=1` were used, Pandas would attempt to remove a column instead. Option (A) is incorrect because DataFrames do not have a `remove()` method. Option (C) is also invalid because there is no `delete()` method for DataFrames. Therefore, the second option is the valid and standard approach for deleting a row by its label. The operation can also be made permanent by adding `inplace=True` if required.

**Final Answer:** `df.drop('R1', axis=0)`

**Answer: (B)**

Q27.

**Solution**

**Concept:** In Pandas, the `values` attribute of a DataFrame is used to access the underlying data stored inside it. Pandas internally relies heavily on NumPy arrays for efficient computation and storage. Therefore, when `df.values` is called, the DataFrame data is returned in the form of a NumPy array rather than a Python list or another Pandas object.

**Solution:** The expression `df.values` returns the contents of the DataFrame as a NumPy ndarray. This array contains all row and column values without labels such as index names or column names. For example, if a DataFrame contains numerical data, `df.values` produces a two-dimensional NumPy array representation of that data. Option (A) is incorrect because the returned object is not a standard Python list. Option (C) is wrong because a Series is one-dimensional, while `df.values` returns the complete dataset. Option (D) is also incorrect because dictionaries store key-value pairs and are unrelated here. Hence, the correct data type of `df.values` is a NumPy ndarray.

**Final Answer:** A NumPy ndarray

**Answer: (B)**



Q28.

**Solution**

**Concept:** Pandas provides built-in statistical methods for analyzing data stored in DataFrames and Series. To calculate the average of a specific column, the column must first be selected from the DataFrame. The `mean()` method is then applied to compute the arithmetic mean of the values present in that column. This approach is widely used in data analysis and reporting tasks.

**Solution:** To calculate the mean of only the 'Salary' column, the correct syntax is `df['Salary'].mean()`. Here, `df['Salary']` selects the required column as a Pandas Series, and the `mean()` method computes its average value. Option (A) is incorrect because the `mean()` method does not take a column name as an argument in this way. Option (C) is invalid because Pandas does not provide an `avg()` method for Series objects. Option (D) is not the standard or recommended syntax in Pandas. Therefore, selecting the column first and then applying `mean()` is the correct and commonly used approach.

**Final Answer:** `df['Salary'].mean()`

**Answer: (B)**

Q29.

**Solution**

**Concept:** Exporting data from a Pandas DataFrame to a CSV file is commonly done using the `to_csv()` method. By default, Pandas also writes the index column into the CSV file. However, in many practical situations, the index is unnecessary and should be excluded. The parameter `index=False` is used to prevent the index column from being exported.

**Solution:** The correct function for exporting a DataFrame to a CSV file without including the index column is `df.to_csv('file.csv', index=False)`. The `to_csv()` method converts the DataFrame into CSV format and saves it to the specified file. Setting `index=False` ensures that row index labels are not written into the output file. Option (B) is incorrect because there is no `write_csv()` method in Pandas. Option (C) uses invalid syntax and unsupported parameters. Option (D) removes column headers instead of the index. Therefore, the first option correctly exports the DataFrame while excluding the index column.

**Final Answer:** `df.to_csv('file.csv', index=False)`

**Answer: (A)**



Q30.

**Solution**

**Concept:** Pandas Series operations are index-based, meaning arithmetic calculations are performed according to matching index labels rather than positions alone. When two Series objects have different indices, Pandas aligns the indices automatically before performing the operation. If a value is missing for a particular index in either Series, the result for that index becomes NaN to indicate missing data.

**Solution:** When  $S1 + S2$  is performed on two Series with non-matching indices, Pandas first aligns the indices and then adds corresponding values. For indices present in one Series but absent in the other, there is no matching value available for computation. In such cases, Pandas assigns NaN to those positions. This behavior helps maintain consistency and avoids incorrect calculations. Option (A) is incorrect because Pandas does not raise an error for non-matching indices. Option (B) is wrong because indices are not ignored. Option (D) is incorrect because addition does not concatenate Series objects. Therefore, non-matching indices produce NaN values in the result.

**Final Answer:** It results in NaN for the non-matching indices.

**Answer: (C)**

Q31.

**Solution**

**Concept:** In Pandas, integer-based indexing uses:

- `iloc []` for selecting rows and columns by position (0-based indexing)
- The first row has index 0, second row has index 1, and so on

**Visual Representation:**

0,0	0,1	0,2	0,3	
1,0	1,1	1,2	1,3	<code>df.iloc[1,2]</code>
2,0	2,1	2,2	2,3	

**Explanation:**

- 2nd row → index 1
- 3rd column → index 2

So the correct integer indexing is:

`df.iloc[1, 2]`

Option (A) and (D) are incorrect because they use index (2,3), which refers to 3rd row and 4th column. Option (C) is incorrect because `loc []` uses labels, not integer positions.

**Final Answer:** `df.iloc[1, 2]`

**Answer: (B)**



Q32.

**Solution**

**Concept:** In Pandas, missing values (NaN) can be replaced using the `fillna()` method. It allows you to fill all NaN values with a specified value.

**Visual Representation:**

10		10
NaN		0
25	$\xrightarrow{\text{fillna}(0)}$	25
NaN		0

**Explanation:**

- `df.fillna(0)` replaces all NaN values with 0.
- `df.replace(NaN, 0)` is not valid unless NaN is properly defined.
- `df.dropna(0)` removes rows or columns with NaN, not fills them.
- `df.null(0)` is not a valid Pandas function.

**Final Answer:** `df.fillna(0)`

**Answer: (B)**



Q33.

**Solution**

**Concept:** In SQL, filtering can happen at different stages of query execution:

- WHERE filters rows before grouping.
- HAVING filters groups after GROUP BY.

**Visual Representation:**



**Explanation:**

- WHERE filters individual rows before aggregation.
- HAVING is used after grouping to filter aggregated results.
- LIKE is used for pattern matching, not grouping filters.
- ORDER BY sorts the result set.

**Final Answer:**

**Answer: (B)**

Q34.

**Solution**

**Concept:** In SQL, sorting of query results is performed using the ORDER BY clause. It allows multiple columns to be sorted in different orders such as ascending (ASC) or descending (DESC). When more than one column is specified, the first column is used as the primary sort key, and the subsequent columns are used to break ties in the order defined.

**Solution:** To arrange data in descending order of Marks and then ascending order of Name, we use the ORDER BY clause with multiple sorting conditions. The correct syntax is ORDER BY Marks DESC, Name ASC. Here, Marks DESC ensures that higher marks appear first, and Name ASC ensures that if two or more students have the same marks, their names are arranged in alphabetical order. Option (B) is incorrect because SORT BY is not standard SQL syntax for this operation. Option (C) reverses the required order of sorting for both columns. Option (D) is incorrect because GROUP BY is used for aggregation, not sorting. Therefore, the correct clause is the one that explicitly specifies both descending and ascending order using ORDER BY.

**Final Answer:**

**Answer: (A)**



Q35.

**Solution**

**Concept:** A Natural Join in relational databases is a join operation that automatically combines two tables based on columns with the same name and compatible data types. It eliminates the need for an explicit join condition because the matching is performed implicitly on all common attributes between the relations. The result includes only one copy of the common columns.

**Solution:** A Natural Join works by identifying columns that have the same name in both tables and using them as the basis for joining. It then returns rows where the values in these common columns match. Unlike other join types, it does not require an explicit ON clause because the matching condition is implied. Option (B) describes a Cartesian product, where every row of one table is combined with every row of another. Option (C) is incorrect because Natural Join does not require manual condition specification. Option (D) is also wrong because it does not return only unmatched rows; that behavior is associated with outer joins. Therefore, a Natural Join is defined as joining tables based on all columns with the same name.

**Final Answer:** Join tables based on all columns with the same name.

**Answer:** (A)

Q36.

**Solution**

**Concept:** In SQL, the function COUNT(\*) counts all rows in a table, regardless of NULL values.

**Key Point:**

- COUNT(\*) counts every row.
- NULL values do not affect COUNT(\*).
- COUNT(column) would ignore NULLs in that column.

**Visual Representation:**

Row 1	COUNT(*) = 4 rows (all counted)
Row 2	
Row 3	
Row 4	

**Explanation:** Even though one column contains 3 NULL values, COUNT(\*) still counts all rows in the table. Since the table has 10 rows, all 10 rows are counted.

**Final Answer:** 10

**Answer:** (B)



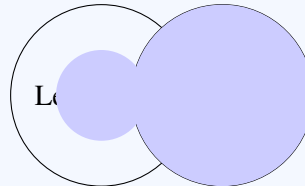
Q37.

**Solution**

**Concept:** SQL Joins are used to combine rows from two tables based on a related column.

- **RIGHT JOIN** returns all rows from the right table and matching rows from the left table.
- If there is no match, NULL values are returned for columns of the left table.

**Visual Representation:**



RIGHT JOIN = Entire Right + Matching Left

**Explanation:**

- **INNER JOIN** returns only matching rows from both tables.
- **LEFT JOIN** returns all rows from the left table.
- **RIGHT JOIN** returns all rows from the right table and matched rows from the left.
- **CROSS JOIN** returns Cartesian product of both tables.

**Final Answer:** RIGHT JOIN

**Answer:** (C)



Q38.

**Solution**

**Concept:** In SQL, when we need to count unique values in a column, we use `COUNT(DISTINCT column_name)`.

- `COUNT(Dept)` counts all non-NULL department entries (including duplicates).
- `COUNT(DISTINCT Dept)` counts only unique departments.

**Visual Representation:**

HR
IT
HR
Finance

Distinct Departments: HR, IT, Finance  
⇒ `COUNT(DISTINCT Dept) = 3`

**Explanation:**

- Option (A) counts all rows including duplicates, so it is incorrect.
- Option (B) correctly counts unique departments.
- Option (C) is syntactically incorrect SQL.
- Option (D) is invalid because SUM is not applicable to text data.

**Final Answer:** `SELECT COUNT(DISTINCT Dept) FROM Employee;`

**Answer: (B)**



Q39.

**Solution**

**Concept:** The HAVING clause in SQL is used to filter grouped results, and it works on aggregated data.

- WHERE filters rows before grouping.
- GROUP BY creates groups on which aggregation is performed.
- HAVING filters those groups after aggregation.

**Visual Representation:****Explanation:**

- HAVING is used after grouping data.
- Without GROUP BY, there are no grouped results to filter, so HAVING cannot function properly in standard SQL usage.
- WHERE, ORDER BY, and JOIN are not required for HAVING.

**Final Answer:**

**Answer: (B)**



Q40.

**Solution**

**Concept:** In SQL, aggregate functions perform calculations on multiple rows and generally ignore NULL values (except COUNT(\*)).

- AVG() ignores NULLs while computing average.
- SUM() ignores NULLs while adding values.
- MAX() ignores NULLs while finding the maximum value.

**Visual Representation:**

10
NULL
30
20

$$\text{AVG} = (10+30+20)/3$$

NULL ignored

**Explanation:**

- All listed aggregate functions ignore NULL values during computation.
- Therefore, each of AVG(), SUM(), and MAX() works by skipping NULL entries.

**Final Answer:** All of the above

**Answer: (D)**



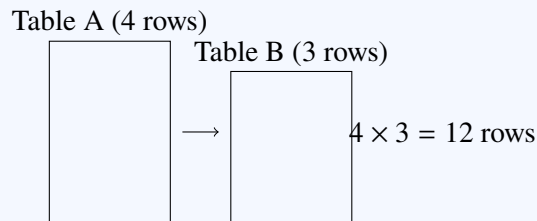
Q41.

**Solution**

**Concept:** The Cartesian Product in SQL returns all possible combinations of rows from two tables.

If Table A has  $m$  rows and Table B has  $n$  rows, then:

$$\text{Cartesian Product} = m \times n$$

**Visual Representation:****Explanation:**

- Each row of Table A pairs with every row of Table B.
- So total combinations =  $4 \times 3 = 12$ .

**Final Answer:**

**Answer:** (B)



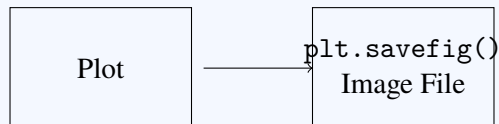
Q42.

**Solution**

**Concept:** In Matplotlib, plots can be saved as image files using a specific function.

- `plt.show()` displays the plot on screen.
- `plt.savefig()` saves the plot to a file (e.g., PNG, JPG, PDF).

**Visual Representation:**



**Explanation:**

- `plt.savefig()` is used to store the plot as an image file.
- `plt.show()` only displays the plot.
- `plt.save()` and `plt.print()` are not valid Matplotlib functions.

**Final Answer:** `plt.savefig()`

**Answer:** (C)



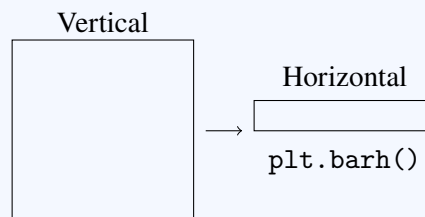
Q43.

**Solution**

**Concept:** In Matplotlib, different functions are used for different types of bar charts:

- `plt.bar()` creates a vertical bar chart.
- `plt.barh()` creates a horizontal bar chart.

**Visual Representation:**



**Explanation:**

- `plt.bar()` is used for vertical bars.
- `plt.barh()` is specifically designed for horizontal bar charts.
- `plt.hbar()` is not a valid Matplotlib function.
- `plt.plot(kind='barh')` is not used in standard Matplotlib plotting syntax.

**Final Answer:** `plt.barh()`

**Answer: (B)**

Q44.

**Solution**

**Concept:** In Matplotlib, the `pyplot` module is used for data visualization, where different parameters help customize plots. The `label` parameter in `plt.plot()` is specifically used to assign a name to a plotted data series, which is later displayed in the legend using `plt.legend()`.

**Solution:** The `label` argument in `plt.plot()` is used to provide a name for the data series so that it can be identified in the legend of the graph. When multiple lines are plotted on the same axes, each line can be given a distinct label, making the visualization easier to interpret. This label does not affect axis titles, individual data points, or the overall chart title. Option (A) is incorrect because axis titles are set using `plt.xlabel()` and `plt.ylabel()`. Option (C) is wrong because labeling individual data points requires annotations. Option (D) is incorrect because chart titles are set using `plt.title()`. Therefore, the correct purpose of the `label` parameter is to name the data series for use in the legend.

**Final Answer:** To label a data series for the legend

**Answer: (B)**



Q45.

**Solution**

**Concept:** In Matplotlib, pie charts are created using the `plt.pie()` function, which provides several parameters to customize the appearance of slices. One important feature is the ability to emphasize or separate a specific slice from the rest of the chart. This is useful for highlighting important categories in data visualization.

**Solution:** The parameter used to highlight or separate a slice in a pie chart is `explode`. This parameter takes a list of values that define how far each slice is offset from the center of the pie chart. A non-zero value causes the corresponding slice to be pulled out, making it visually distinct from the others. Option (A), `shadow`, only adds a shadow effect but does not separate slices. Option (B), `autopct`, is used to display percentage values. Option (D), `colors`, is used to assign colors to slices. Therefore, the correct parameter for separating a slice is `explode`.

**Final Answer:** `explode`

**Answer:** (C)

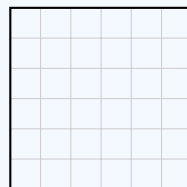
Q46.

**Solution**

**Concept:** In Matplotlib, a grid is used to improve readability of plots by adding horizontal and vertical reference lines.

- `plt.grid(True)` enables the grid on the current plot.

**Visual Representation:**



Plot with Grid

— `plt.grid(True)`

**Explanation:**

- `plt.grid(True)` activates grid lines in the plot.
- `plt.show_grid()`, `plt.background('grid')`, and `plt.draw(grid=True)` are not valid Matplotlib functions.

**Final Answer:** `plt.grid(True)`

**Answer:** (A)



Q47.

**Solution**

**Concept:** The 'Right to be Forgotten' is a legal principle in data privacy that allows individuals to request the removal of their personal data from online platforms under certain conditions. It is closely associated with the concept of a digital footprint, which refers to the trace of data that individuals leave behind while using the internet.

**Solution:** The 'Right to be Forgotten' is most closely related to the concept of a digital footprint. This right enables users to control or delete personal information that exists online, especially when it is no longer relevant or is misused. It aims to protect user privacy in the digital environment by reducing unnecessary exposure of personal data. Option (A), Intellectual Property Rights, deals with ownership of creative works. Option (C), E-waste Management, is related to electronic waste disposal. Option (D), Open Source Software, refers to software with publicly available source code. Therefore, the correct association is with the digital footprint, as it directly involves managing personal data presence on the internet.

**Final Answer:** Digital Footprint

**Answer: (B)**

Q48.

**Solution**

**Concept:** Phishing is a type of cybercrime in which attackers attempt to trick individuals into revealing sensitive information such as passwords, PINs, or bank details. This is typically done by disguising fraudulent messages as legitimate communications from trusted organizations like banks or service providers.

**Solution:** An example of phishing is receiving an email that appears to come from a legitimate bank asking for sensitive information such as a PIN or password. In reality, such emails are designed to deceive users into providing confidential data that can be misused. Option (A) is incorrect because using a password with consent is not malicious. Option (C) refers to copyright violation, not phishing. Option (D) is related to improper disposal of electronic devices. Therefore, the correct example of phishing is a fraudulent email pretending to be from a bank to steal personal information.

**Final Answer:** Fake bank email to steal PIN

**Answer: (B)**



Q49.

**Solution**

**Concept:** Software piracy refers to the illegal copying, distribution, or use of software without proper authorization or licensing. It violates intellectual property laws and can lead to legal consequences as well as financial losses for software developers and companies.

**Solution:** The unauthorized copying and distribution of software is known as software piracy. This includes activities such as installing software without a valid license, sharing paid software illegally, or distributing cracked versions. Option (B), plagiarism, refers to copying someone else's written or creative work without credit. Option (C), hacking, involves unauthorized access to systems. Option (D), cyber stalking, refers to harassment using digital platforms. Therefore, the correct term for illegal copying and distribution of software is software piracy.

**Final Answer:** Software Piracy

**Answer:** (A)

Q50.

**Solution**

**Concept:** E-waste contains various hazardous materials, including heavy metals that are harmful to both the environment and human health. Some of these metals can cause serious neurological and physiological damage if they enter the body through improper disposal or exposure.

**Solution:** Mercury is a toxic heavy metal commonly found in electronic waste such as batteries, switches, and fluorescent components. Exposure to mercury can severely damage the human nervous system, affecting brain function, coordination, and overall neurological health. Option (A), iron, is not highly toxic in this context. Option (C), aluminum, is generally less harmful in typical exposure levels. Option (D), magnesium, is also not associated with severe nervous system damage. Therefore, mercury is the correct answer as it is a well-known hazardous component of e-waste.

**Final Answer:** Mercury

**Answer:** (B)

“



**Answer Key**

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

