සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved]

Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc

Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 393 MARKING

2026/01/07
ictfromabc
Ravindu Bandaranayake

1. Which of the followings are steps in data processing ?
පහත සඳහන් පියවර වලින් දත්ත සැකසීමේ පියවර මොනවාද ?

   1) Data retrieval, encryption, and decryption.
   දත්ත ලබා ගැනීම, කේතනය සහ විකේතනය කිරීම.
   2) Data analysis, visualization, and reporting.
   දත්ත විශ්ලේෂණය, දෘශ්‍යකරණය සහ වාර්තා කිරීම.
   3) Data gathering, validation, processing, output and storage.
   දත්ත රැස්කිරීම, වලංගු කිරීම, සැකසීම, ප්‍රතිදානය සහ ආචයනය.
   4) Data transmission, compression, and decompression.
   දත්ත සම්ප්‍රේෂණය, සම්පීඩනය සහ විසංයෝජනය.
   5) Data formatting, debugging, and execution.
   දත්ත හැඩතල ගැන්වීම, නිදොස්කරණය සහ ක්‍රියාත්මක කිරීම.

Answer : 3)

Fundamental Steps in Data Processing:
දත්ත සැකසීමේ මූලික පියවර:

- **Data Collection දත්ත එකතුව:**
  Gathering raw data from various sources.
  විවිධ මූලාශ්‍රවලින් අමු දත්ත රැස් කිරීම.
- **Data Validation දත්ත වලංගු කිරීම:**
  Ensuring that the data is accurate, complete, and reliable.
  දත්ත නිවැරදි, සම්පූර්ණ සහ විශ්වසනීය බව සහතික කිරීම.
- **Data Processing දත්ත සැකසීම:**
  Converting raw data into meaningful information through operations and transformations.
  මෙහෙයුම් සහ පරිවර්තනයන් හරහා අමු දත්ත අර්ථවත් තොරතුරු බවට පරිවර්තනය කිරීම.
- **Data Output දත්ත ප්‍රතිදානය:**
  Presenting the processed data in a useful format, such as reports or visualizations.
  සකසන ලද දත්ත වාර්තා හෝ දෘශ්‍යකරණය වැනි ප්‍රයෝජනවත් ආකෘතියකින් ඉදිරිපත් කිරීම.
- **Data Storage දත්ත ගබඩාව:**
  Saving the processed data and results for future retrieval and use.
  අනාගත ලබා ගැනීම සහ භාවිතය සඳහා සැකසූ දත්ත සහ ප්‍රතිඵල සුරැකීම.

Now, let's evaluate the given options:
දැන්, අපි ලබා දී ඇති විකල්ප ඇගයීමට ලක් කරමු:

- **Option / විකල්පය 1**
  This option involves data security and management rather than the core steps of data processing. Data retrieval is related to accessing data, encryption/decryption are about securing data, not processing it.
  මෙම විකල්පය දත්ත සැකසීමේ මූලික පියවරයන්ට වඩා දත්ත ආරක්ෂාව සහ කළමනාකරණය ඇතුළත් වේ. දත්ත ලබා ගැනීම දත්ත වෙත ප්‍රවේශ වීම හා සම්බන්ධ වන අතර සංකේතනය/විකේතනය යනු දත්ත සුරක්ෂිත කිරීම මිස එය සැකසීම නොවේ.
- **Option / විකල්පය 2**
  While these are important for understanding and presenting data, they are generally considered part of data analysis and presentation, rather than the core steps of data processing.
  දත්ත අවබෝධ කර ගැනීම සහ ඉදිරිපත් කිරීම සඳහා මේවා වැදගත් වන අතර, ඒවා සාමාන්‍යයෙන් දත්ත සැකසීමේ මූලික පියවරයන් වෙනුවට දත්ත විශ්ලේෂණයේ සහ ඉදිරිපත් කිරීමේ කොටසක් ලෙස සැලකේ.

- **Option / විකල්පය 3**
  This option accurately covers the fundamental steps in data processing.
  මෙම විකල්පය දත්ත සැකසීමේ මූලික පියවර නිවැරදිව ආවරණය කරයි.

- **Option / විකල්පය 4**
  These steps are related to data communication and optimization rather than core data processing. Transmission involves sending data, and compression/decompression involves data size management.
  මෙම පියවර මූලික දත්ත සැකසීමට වඩා දත්ත සන්නිවේදනය සහ ප්‍රශස්තකරණයට සම්බන්ධ වේ. සම්ප්‍රේෂණයට දත්ත යැවීම ඇතුළත් වන අතර සම්පීඩනය/විසංයෝජනය දත්ත ප්‍රමාණය කළමනාකරණය ඇතුළත් වේ.

- **Option / විකල්පය 5**
  Data formatting can be part of data processing, but debugging and execution are more related to programming and software development, not directly to data processing.
  දත්ත හැඩතල ගැන්වීම දත්ත සැකසීමේ කොටසක් විය හැක, නමුත් දෝශ නිරාකරණය සහ ක්‍රියාත්මක කිරීම වඩාත් සම්බන්ධ වන්නේ ක්‍රමලේඛනයට සහ මෘදුකාංග සංවර්ධනයට මිස සෘජුවම දත්ත සැකසීමට නොවේ.

So, the option that aligns with the fundamental steps in data processing is 3) Data gathering, validation, processing, output and storage.
එබැවින්, දත්ත සැකසීමේ මූලික පියවරයන් සමඟ සමපාත වන විකල්පය වන්නේ 3) දත්ත රැස් කිරීම, වලංගු කිරීම, සැකසීම, ප්‍රතිදානය සහ ගබඩා කිරීම වේ.

2. Which of the following hexadecimal number is equivalent to the binary number 110110101011_2 ?
   පහත සඳහන් ෂඩ් දශමය සංඛ්‍යා අතුරින් 110110101011_2 ද්විමය සංඛ්‍යාවට සමාන වන්නේ කුමක්ද?

   1) <u>DAB_{16}</u> 2) $B6B_{16}$ 3) $6DB_{16}$ 4) $AD6_{16}$ 5) BAD_{16}

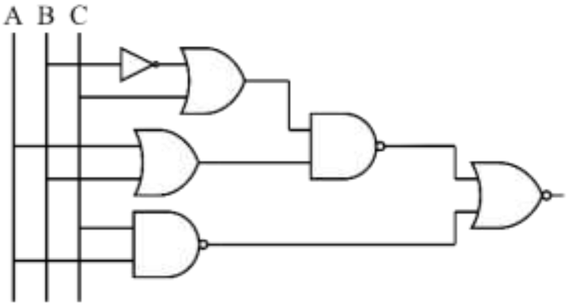
Answer : 1)

Let's convert 110110101011_2 in to Hexadecimal.
110110101011_2 ෂඩ් දශමයට පරිවර්තනය කරමු.

110110101011_2		
1101	1010	1011
13 = D	10 = A	11 = B
DAB_{16}		

Therefore, the correct answer is 1) DAB_{16}.
එබැවින්, නිවැරදි පිළිතුර නම් 1) DAB_{16}.

3. What is the answer after simplifying the logic circuit shown below?
   පහත දක්වා ඇති තාර්කික පරිපථය සුළු කළ පසු පිළිතුර කුමක්ද?



1) 1
2) <u>AC</u>
3) ACB
4) $\overline{B}A$
5) $(A + B)\overline{C}$

Answer : 2)

The expression of the logic circuit given above is $\overline{(\overline{\overline{B} + C . A + B}) + \overline{AC}}$
ඉහත දක්වා ඇති තාර්කික පරිපථයේ ප්‍රකාශය වන්නේ $\overline{(\overline{\overline{B} + C . A + B}) + \overline{AC}}$

The statement is then trivialized as follows.
පසුව එම ප්‍රකාශය පහත ලෙස සුළු කරනු ලැබේ.

$= \overline{(\overline{\overline{B} + C.A + B}) + \overline{AC}}$	
$= \overline{\overline{(\overline{\overline{B} + C.A + B})}}.\overline{\overline{AC}}$	De morgan's law ද මුවාර් ප්‍රමේයය
$= (\overline{B} + CA + B)\overline{\overline{AC}}$	Double inverse law ද්විත්ව ප්‍රතිලෝම න්‍යාය
$= (\overline{B} + CA + B)AC$	Double inverse law ද්විත්ව ප්‍රතිලෝම න්‍යාය
$= (CA + 1)AC$	Complement law ප්‍රතිලෝම න්‍යාය
$= 1AC$	Identity law සර්වසාම්‍ය න්‍යාය
$= AC$	Identity law සර්වසාම්‍ය න්‍යාය

Therefore the correct answer is **2).**
එබැවින් නිවැරදි පිළිතුර **2)** වේ.

4. What is the correct SQL code to insert the name Supun in the FirstName field of the Student table?
Student යන වගුවේ FirstName යන ක්ෂේත්‍රයට Supun යන නම ඇතුළත් කිරීමට අදාළ නිවැරදි SQL කේතය කුමක්ද?

1) `INSERT INTO Student('Supun')INTO'FirstName';`
2) `INSERT('Supun')INTO Student FirstName;`
3) <u>`INSERT INTO Student(FirstName)VALUES('Supun');`</u>
4) `INSERT INTO Student FirstName VALUES (Supun);`
5) `INSERT(FirstName)INTO Student Supun;`

Answer: 3)

SQL Syntax for INSERT query / INSERT විමසුම සඳහා SQL වාක්‍ය ඛණ්ඩය:

```
INSERT INTO TableName (ColumnName1, …) VALUES (Value);
```

- `TableName:`
Specifies the table where the record will be inserted.
වාර්තාව ඇතුළත් කරන වගුව නියම කරයි.

- `(ColumnName1, …):`
Specifies the column(s) where the value(s) will be inserted.
අගය ඇතුළත් කරන තීරුව නියම කරයි.

- `(Value):`
Provides the actual value(s) to insert.
ඇතුළු කිරීමට සත්‍ය අගය සපයයි.

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

1) **`INSERT INTO Student('Supun') INTO 'FirstName';`**
Incorrect syntax; 'Supun' and 'FirstName' are incorrectly placed.
වැරදි වාක්‍ය ඛණ්ඩයකි Supun සහ FirstName වැරදි ලෙස තබා ඇත.

2) **`INSERT ('Supun') INTO Student FirstName;`**
Incorrect syntax; column name placement is invalid.
වැරදි වාක්‍ය ඛණ්ඩයකි තීරු නම් ස්ථානගත කිරීම වලංගු නොවේ.

3) **`INSERT INTO Student(FirstName) VALUES ('Supun');`**
Correct syntax; inserts 'Supun' into the FirstName column of the Student table.
නිවැරදි වාක්‍ය ඛණ්ඩය වේ Student වගුවේ FirstName තීරුවට Supun ඇතුළු කරයි.

4) **`INSERT INTO Student FirstName VALUES (Supun);`**
Incorrect syntax; column name placement and value quoting are incorrect.
වැරදි වාක්‍ය ඛණ්ඩයකි තීරු නම් ස්ථානගත කිරීම සහ අගය උපුටා දැක්වීම වැරදිය.

5) **`INSERT (FirstName) INTO Student Supun;`**
Incorrect syntax; lacks proper column and value placement.
වැරදි වාක්‍ය ඛණ්ඩයකි නිසි තීරුව සහ අගය ස්ථානගත කිරීමක් නොමැත.

The correct answer is / නිවැරදි පිළිතුර වන්නේ,

3) `INSERT INTO Student (FirstName) VALUES ('Supun');`

5. What is a key advantage of denormalization in a data warehouse?
දත්ත ගබඩාවක ප්‍රති-ප්‍රමතකරණයේ ප්‍රධාන වාසිය කුමක්ද?

1) Minimizing storage space usage / ගබඩා ඉඩ භාවිතය අවම කිරීම.
2) Reducing data redundancy / දත්ත අතිරික්තය අඩු කිරීම.
3) Improving query performance for read-heavy operations.
කියවීමේ බර මෙහෙයුම් සඳහා විමසුම් කාර්ය සාධනය වැඩි දියුණු කිරීම.
4) Eliminating complex joins between tables / වගු අතර සංකීර්ණ සන්ධි ඉවත් කිරීම.
5) Ensuring strict ACID compliance / දැඩි ACID අනුකූලතාව සහතික කිරීම.

Answer: 3)

Denormalization is considered the opposite of normalization, but they serve different purposes in database design.
ප්‍රමතකරණයේ ප්‍රතිවිරුද්ධ දෙය ලෙස ප්‍රති-ප්‍රමතකරණය සැලකේ, නමුත් ඒවා දත්ත සමුදාය නිර්මාණයේදී විවිධ අරමුණු ඉටු කරයි.

While normalization focuses on reducing redundancy and improving data integrity, denormalization reintroduces redundancy to improve query performance and simplify database design for specific use cases.
ප්‍රමතකරණය අතිරික්තය අඩු කිරීම සහ දත්ත අඛණ්ඩතාව වැඩි දියුණු කිරීම කෙරෙහි අවධානය යොමු කරන අතර, විමසුම් කාර්ය සාධනය වැඩි දියුණු කිරීමට සහ විශේෂිත භාවිත අවස්ථා සඳහා දත්ත සමුදා නිර්මාණය සරල කිරීමට ප්‍රති-ප්‍රමතකරණය අතිරික්තතාව නැවත හඳුන්වා දෙයි.

While it can have some advantages, the main result of denormalization is an increase in data redundancy
එය යම් වාසි තිබිය හැකි වුවද, ප්‍රති-ප්‍රමතකරණයෙහි ප්‍රධාන ප්‍රතිඵලය වන්නේ දත්ත අතිරික්තය වැඩි වීමයි

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
Incorrect. Denormalization increases storage due to data redundancy.
වැරදියි. දත්ත අතිරික්තය හේතුවෙන් ප්‍රති-ප්‍රමතකරණය ගබඩාව වැඩි කරයි.

Option / විකල්ප 2)
Incorrect. Denormalization increases redundancy.
වැරදියි. ප්‍රති-ප්‍රමතකරණය අතිරික්තය වැඩි කරයි.

Option / විකල්ප 3)
Correct. Denormalization speeds up read-heavy queries by reducing joins.
නිවැරදියි. ප්‍රති-ප්‍රමතකරණය සම්බන්ධවීම් අඩු කිරීමෙන් කියවීමට බර විමසුම් වේගවත් කරයි.

Option / විකල්ප 4)
Incorrect. Denormalization reduces joins, but its main benefit is query performance.
වැරදියි. ප්‍රති-ප්‍රමතකරණය සම්බන්ධවීම් අඩු කරයි, නමුත් එහි ප්‍රධාන වාසිය විමසුම් කාර්ය සාධනයයි.

Option / විකල්ප 5)
Incorrect. Denormalization does not affect ACID compliance, which is for transactional systems.
වැරදියි. ප්‍රති-ප්‍රමතකරණය ගනුදෙනු පද්ධති සඳහා වන ACID අනුකූලතාවයට බලපාන්නේ නැත.

6. Which SQL command would remove all rows from a table called orders without deleting the table itself?
කුමන SQL විධානය මඟින් වගුව මකා නොදමමින් orders නම් වගුවකින් සියලුම පේළි ඉවත් කරයිද?

1) DROP TABLE orders; 2) DELETE TABLE orders; 3) DELETE FROM orders;
4) TRUNCATE TABLE orders; 5) REMOVE FROM orders;

Answer: 3)

To remove all rows from a table called orders without deleting the table itself, the SQL command used should be carefully selected.
වගුව මකා දැම්මකින් තොරව ඇණවුම් නම් වගුවකින් සියලුම පේළි ඉවත් කිරීමට, භාවිතා කරන SQL විධානය ප්‍රවේශමෙන් තෝරා ගත යුතුය.

Explanation of each command / එක් එක් විධානය පිළිබඳ පැහැදිලි කිරීම:

1) **`DROP TABLE orders;`**
This command deletes the entire table along with its structure and all the rows inside it. The table would no longer exist in the database. Hence, it is incorrect.
මෙම විධානය මඟින් සම්පූර්ණ වගුව එහි ව්‍යුහය සහ එහි ඇති සියලුම පේළි මකා දමයි. වගුව තවදුරටත් දත්ත ගබඩාවේ නොපවතියි. එබැවින් එය වැරදිය.

2) **`DELETE TABLE orders;`**
This is not a valid SQL syntax. The `DELETE` command is used to remove rows from a table, but it must be followed by `FROM` instead of `TABLE`. Thus, this option is invalid.
මෙය වලංගු SQL වාක්‍ය ඛණ්ඩයක් නොවේ. වගුවකින් පේළි ඉවත් කිරීමට `DELETE` විධානය භාවිතා කරයි, නමුත් එය `TABLE` වෙනුවට `FROM` අනුගමනය කළ යුතුය. මේ අනුව, මෙම විකල්පය වලංගු නොවේ.

3) **`DELETE FROM orders;`**
This command removes all rows from the orders table while preserving the table structure. Since the question explicitly states that the table itself should not be deleted, this is the correct answer.
මෙම විධානය මඟින් වගු ව්‍යුහය ආරක්ෂා කරන අතරම ඇණවුම් වගුවෙන් සියලුම පේළි ඉවත් කරයි. වගුව මකා නොදැමිය යුතු බව ප්‍රශ්නයේ පැහැදිලිව සඳහන් වන බැවින්, මෙය නිවැරදි පිළිතුරයි.

Additionally, this command allows for fine-grained control if a `WHERE` clause is added to delete specific rows instead of all rows.
මීට අමතරව, සියලුම පේළි වෙනුවට විශේෂිත පේළි මකා දැමීමට `WHERE` වගන්තියක් එකතු කළහොත්, මෙම විධානය සියුම් පාලනයක් සඳහා ඉඩ සලසයි.

4) **`TRUNCATE TABLE orders;`**
This command also removes all rows from the table, but it operates differently. `TRUNCATE` resets the table by deallocating all data pages, which is faster than `DELETE` for large datasets.
මෙම විධානය වගුවෙන් සියලුම පේළි ඉවත් කරයි, නමුත් එය වෙනස් ආකාරයකින් ක්‍රියා කරයි. `TRUNCATE` විශාල දත්ත කට්ටල සඳහා `DELETE` ට වඩා වේගවත් වන, සියලු දත්ත පිටු බෙදා හැරීමෙන් වගුව යළි සකසයි.

However, the question emphasizes preserving the table, and while `TRUNCATE` does preserve the table structure, it bypasses certain database features like row-by-row triggers and transaction logs. Therefore, it is not the correct choice for this scenario.
කෙසේ වෙතත්, ප්‍රශ්නය වගුව සංරක්ෂණය කිරීම අවධාරණය කරන අතර, `TRUNCATE` විසින් වගු ව්‍යුහය ආරක්ෂා කරන අතර, එය පේළියෙන් පේළි ප්‍රේරක සහ ගනුදෙනු ලොග වැනි ඇතැම් දත්ත සමුදා විශේෂාංග මඟ හරියි. එමනිසා, මෙම අවස්ථාව සඳහා එය නිවැරදි තේරීම නොවේ.

5) **`REMOVE FROM orders;`**
This is not a valid SQL command. The term `REMOVE` is not part of standard SQL syntax. Hence, this option is incorrect.
මෙය වලංගු SQL විධානයක් නොවේ. `REMOVE` යන යෙදුම සම්මත SQL වාක්‍ය ඛණ්ඩයේ කොටසක් නොවේ. එබැවින් මෙම විකල්පය වැරදියි.

Therefore, the correct answer is 3)
එබැවින්, නිවැරදි පිළිතුර 3) වේ.

7. What will be the output of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් වේවිද?

```
a = [1,2,3,4,5]
print(a[2:4])
```

1) `[1, 2]` 2) `[2, 3]` 3) `[3, 4]` 4) `[4, 5]` 5) `[3, 4, 5]`

Answer : 3)

Explanation / පැහැදිලි කිරීම

1 **List Creation / List නිර්මාණය**
`a = [1, 2, 3, 4, 5]`
This line creates a list `a` containing the elements "`[1, 2, 3, 4, 5]`".
මෙම පේළිය "`[1, 2, 3, 4, 5]`" elements list එකක් නිර්මාණය කරයි.

2 **Slicing the List / list එකක් slicing කිරීම**
`print(a[2:4])`
This line slices the list '`a`' from index 2 to index 4. In Python, list slicing is performed using the syntax '`a[start:stop]`', where '`start`' is the index to start from (inclusive), and '`stop`' is the index to end at (exclusive).
මෙම පේළිය '`a`' list එකේ 2සිට දර්ශක 4දක්වා slicing කරයි. Python හි, '`a[start:stop]`' වාක්‍ය ඛණ්ඩය භාවිතයෙන් list slicing කිරීම සිදු කරයි, එහිදී '`start`' යනු (inclusive),ආරම්භ කළ යුතු දර්ශකය වේ. '`stop`' යනු (exclusive) අවසන් විය යුතු දර්ශකයයි.

Here, '`a[2:4]`' means / මෙහි, '`a[2:4]`' යන්නෙන් අදහස් වන්නේ:

- Start at index 2 (the third element, which is '`3`').
  2වන index එකෙන් ආරම්භ කරන්න (තුන්වන element එක, එනම් '`3`').

- Stop at index 4 (the fifth element, which is '`5`', but not including it).
  index 4හි නවත්වන්න (පස්වන element එක, එනම් '`5`' ඇතුළත් නොවේ).

So, the slice '`a[2:4]`' includes elements at index 2 and 3, which are '`3`' and '`4`'.
එබැවින්, '`a[2:4]`' slice එකෙහි '`3`' සහ '`4`' වන index 2 සහ 3 හි elements වල ඇතුළත් වේ.

Therefore, the correct answer is **`[3, 4]`.**
එබැවින් නිවැරදි පිළිතුර **`[3, 4]`** වේ.

8. What is the output of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
a = 5
b = 10
c = 15
result = a | b & c ^ a
print(result)
```

1) `10` 2) `5` 3) `15` 4) `0` 5) `8`

Answer : 3)

Below is a description of each of the guidelines and their order of priority.
පහත දැක්වෙන්නේ මෙහි දක්වා ඇති එක් එක් මෙහෙයවනයන් හා ඒවාගේ ප්‍රමුඛතා අනුපිළිවෙල වේ.

`|` - Bitwise OR
`&` - Bitwise AND
`^` - Bitwise XOR

In Python, bitwise operators have different precedence levels, which determine the order of operations:
පයිතන් හි, bitwise operators හට විවිධ ප්‍රමුඛතා මට්ටම් ඇත, එය මෙහෙයුම් අනුපිළිවෙල තීරණය කරයි:

- & (AND) has the highest precedence. / & (AND) ඉහළම ප්‍රමුඛතාවය ඇත.
- ^ (XOR) comes next. / ^ (XOR) ඊළඟට පැමිණේ.
- | (OR) has the lowest precedence. / | (OR) අඩුම ප්‍රමුඛතාවය ඇත.

Below is the order of solving this / පහතින් මෙය විසඳන පිළිවෙල දක්වා ඇත.

`a = 5` (binary: `0101`)
`b = 10` (binary: `1010`)
`c = 15` (binary: `1111`)

b & c				
	1	0	1	0
AND	1	1	1	1
	1	0	1	0

1010 ^ a				
	1	0	1	0
XOR	0	1	0	1
	1	1	1	1

a | 1111				
	0	1	0	1
OR	1	1	1	1
	1	1	1	1

Accordingly, the output is 15. So, the answer number is 3).
ඒ අනුව, ප්‍රතිදානය 15 වේ. එමනිසා, පිළිතුරු අංකය 3) වේ.

9. Which list method is used to sort the elements of a list in place?
ලැයිස්තුවක මූලාංග, වර්ග කිරීමට භාවිතා කරන ලැයිස්තු ක්‍රමවේදය (method) කුමක්ද?

1) `sort()` 2) `sorted()` 3) `order()` 4) `arrange()` 5) `Sort()`

Answer : 1)

Option / විකල්පය 1)
The `sort()` method is a built-in Python list method used to sort the elements of the list in place. This means that the method rearranges the elements of the original list itself and does not create a new list. This method modifies the original list and does not return any value (returns None). It sorts the list in ascending order by default, but you can specify the `reverse = True` parameter to sort in descending order.
`sort()` ක්‍රමය යනු ලැයිස්තුවේ ඇති මූලද්‍රව්‍ය ස්ථානගත කිරීම සඳහා භාවිතා කරන built-in Python ලැයිස්තු ක්‍රමයකි. මෙයින් අදහස් කරන්නේ ක්‍රමය මුල් ලැයිස්තුවේ මූලද්‍රව්‍ය නැවත සකස් කරන අතර නව ලැයිස්තුවක් නිර්මාණය නොකරන බවයි. මෙම ක්‍රමය මුල් ලැයිස්තුව වෙනස් කරන අතර කිසිදු අගයක් ලබා නොදේ (None නැවත ලබා දෙයි). එය පෙරනිමියෙන් ලැයිස්තුව ආරෝහණ අනුපිළිවෙලට සකසයි, නමුත් ඔබට අවරෝහණ අනුපිළිවෙලින් වර්ග කිරීමට `reverse = True` පරාමිතිය නියම කළ හැක.

Option / විකල්පය 2)
The `sorted()` function, unlike `sort()` does not modify the original list. Instead, it returns a new list that is sorted. This allows you to keep the original list unchanged while getting a sorted version of it. This function can be used with any iterable, not just lists, and also supports an optional reverse parameter to sort in descending order.
`sorted()` ශ්‍රිතය `sort()` මෙන් නොව, මුල් ලැයිස්තුව වෙනස් නොකරයි. ඒ වෙනුවට, එය වර්ග කළ නව ලැයිස්තුවක් ලබා දෙයි. මෙමගින් ඔබට මුල් ලැයිස්තුවේ අනුපිළිවෙළට සකස් කරන ලද අනුවාදයක් ලබා ගන්නා අතරතුර නොවෙනස්ව තබා ගැනීමට ඉඩ සලසයි. මෙම ශ්‍රිතය ලැයිස්තු පමණක් නොව, ඕනෑම පුනරාවර්තනයක් සමඟ භාවිතා කළ හැකි අතර, අවරෝහණ අනුපිළිවෙලට වර්ග කිරීමට විකල්ප ප්‍රතිලෝම පරාමිතියකට ද සහය දක්වයි.

Option / විකල්පය 3)
The `order()` method does not exist in Python for sorting lists. It is not a standard function or method in Python's list operations. If you come across `order()` in other contexts, it might be a custom method or a misinterpretation of the sorting operations provided by Python.
ලැයිස්තු වර්ග කිරීම සඳහා `order()` ක්‍රමය Python හි නොමැත. එය Python හි ලැයිස්තු මෙහෙයුම් වල සම්මත ශ්‍රිතයක් හෝ ක්‍රමයක් නොවේ. ඔබට වෙනත් සන්දර්භයන්හිදී `order()` හමු වුවහොත්, එය අභිරුචි ක්‍රමයක් හෝ Python විසින් සපයනු ලබන වර්ග කිරීමේ මෙහෙයුම් වැරදි ලෙස අර්ථ දැක්වීමක් විය හැකිය.

Option / විකල්පය 4)

Similar to `order()`, `arrange()` is not a standard Python method or function for sorting lists. It does not exist in the Python language or its standard libraries. It might be confused with other methods or could be a term used in other programming contexts or custom implementations.
`order()`, `arrange()` සමාන ලැයිස්තු වර්ග කිරීම සඳහා සම්මත Python ක්‍රමයක් හෝ ශ්‍රිතයක් නොවේ. එය පයිතන් භාෂාවේ හෝ එහි සම්මත පුස්තකාලවල නොපවතී. එය වෙනත් ක්‍රම සමඟ ව්‍යාකූල විය හැකිය, නැතහොත් වෙනත් ක්‍රමලේඛන සන්දර්භයන් හෝ අභිරුචි ක්‍රියාත්මක කිරීම් වලදී භාවිතා කරන යෙදුමක් විය හැකිය.

Option / විකල්පය 5)

This method also does not exist in Python for list sorting. It is not a built-in function or method in Python for sorting lists. The term might be misleading or used in other programming languages or libraries, but it is not part of Python's standard list operations.
මෙම ක්‍රමය ලැයිස්තු වර්ග කිරීම සඳහා Python හි ද නොපවතී. එය ලැයිස්තු වර්ග කිරීම සඳහා පයිතන් හි ගොඩනඟන ලද ශ්‍රිතයක් හෝ ක්‍රමයක් නොවේ. මෙම යෙදුම නොමඟ යවන හෝ වෙනත් ක්‍රමලේඛන භාෂා හෝ පුස්තකාලවල භාවිතා කළ හැක, නමුත් එය Python හි සම්මත ලැයිස්තු මෙහෙයුම් වල කොටසක් නොවේ.

In summary, when dealing with list sorting in Python, you would use `sort()` to modify the original list in place and `sorted()` to obtain a new sorted list without altering the original. The other terms are not part of Python's standard list operations for sorting.
සාරාංශයක් ලෙස, Python හි ලැයිස්තු වර්ග කිරීම සමඟ කටයුතු කරන විට, ඔබ මුල් ලැයිස්තුව වෙනස් කිරීමට `sort()` භාවිතා කරන අතර මුල් ලැයිස්තුව වෙනස් නොකර නව වර්ග කළ ලැයිස්තුවක් ලබා ගැනීමට `sorted()` භාවිතා කරයි. අනෙකුත් නියමයන් වර්ග කිරීම සඳහා Python හි සම්මත ලැයිස්තු මෙහෙයුම්වල කොටසක් නොවේ.

Therefore, the correct method to sort the elements of a list in place is 1) `sort()`
එබැවින්, ලැයිස්තුවක මූලද්‍රව්‍ය ස්ථානගත කිරීමේ නිවැරදි ක්‍රමය වන්නේ 1) `sort()` වේ.

10. Which of the following is incorrect about descriptive lists?
විස්තරාත්මක ලැයිස්තු සම්බන්ධයෙන් නිවැරදි නොවන්නේ කුමක්ද?

1) `<dl>` is a descriptive list / `<dl>` යනු විස්තරාත්මක ලැයිස්තු වේ.
2) `<dt>` is a descriptive term/topic / `<dt>` යනු විස්තරාත්මක මාතෘකා වේ.
3) `<dd>` is a descriptive data / `<dd>` යනු විස්තරාත්මක දත්ත වේ.
4) <u>`<dl>` has no closing tag / `<dl>` ට අවසන් කිරීමේ උසුලනයක් නොපවතී.</u>
5) `<dl><dt><dd>` is the correct order / `<dl><dt><dd>` යනු නිවැරදි පිළිවෙලයි.

Answer : 4)

All these tags have closing tags. And these tags are used as follows.
මෙම උසුලන සියල්ලටම අවසන් කිරීමේ උසුලන ද ඇත. තවද, මෙම උසුලන පහත පරිදි භාවිතා වේ.

```
<dl>
    <dt>Science</dt>
    <dd>Maths</dd>
    <dd>Bio</dd>
    <dd>Physics</dd>
    <dd>Chemistry</dd>
    <dd>ICT</dd>
</dl>
```

→

Science
    Maths
    Bio
    Physics
    Chemistry
    ICT