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 412 MARKING

2026/01/26
ictfromabc
Ravindu Bandaranayake

1. Which of the following is an example of structured data?
ව්‍යුහගත දත්ත සඳහා උදාහරණයක් වන්නේ පහත සඳහන් කුමක්ද?

1) A handwritten note / අතින් ලියන ලද සටහනක්.
<u>2) A database table / දත්ත සමුදා වගුවක්.</u>
3) A social media post / සමාජ මාධ්‍ය සටහනක්.
4) An email message / ඊමේල් පණිවිඩයක්.
5) A digital photograph / ඩිජිටල් ඡායාරූපයක්.

Answer : 2)

Explanation / පැහැදිලි කිරීම

Structured data refers to data that is organized in a predefined format, typically within a table or database, making it easy to search, analyze, and retrieve. It is highly organized and often stored in rows and columns, such as in relational databases. A common example of structured data is a database table, where each row contains a specific record, and each column holds a particular type of data (like name, age, or email address). This structure makes it simple to perform operations like sorting, filtering, and querying.
ව්‍යුහගත දත්ත යනු සාමාන්‍යයෙන් වගුවක් හෝ දත්ත සමුදායක් තුළ, සෙවීමට, විශ්ලේෂණය කිරීමට සහ නැවත ලබා ගැනීමට පහසු කරවන පූර්ව නිශ්චිත ආකෘතියකින් සංවිධානය කර ඇති දත්ත වේ. එය ඉතා සංවිධානාත්මක වන අතර බොහෝ විට සම්බන්ධතා දත්ත සමුදායන් වැනි පේළි සහ තීරු වල ගබඩා කර ඇත. ව්‍යුහගත දත්ත සඳහා පොදු උදාහරණයක් වන්නේ දත්ත සමුදා වගුවකි, එහිදී එක් එක් පේළියේ නිශ්චිත වාර්තාවක් අඩංගු වන අතර, සෑම තීරුවකටම විශේෂිත දත්ත වර්ගයක් (නම, වයස, හෝ ඊමේල් ලිපිනය වැනි) ඇත. මෙම ව්‍යුහය වර්ග කිරීම, පෙරීම සහ විමසීම් වැනි මෙහෙයුම් සිදු කිරීම සරල කරයි.

On the other hand, unstructured data lacks such a predefined format, making it more difficult to store, search, and analyze. Examples of unstructured data include handwritten notes, social media posts, email messages, and digital photographs, which do not follow a consistent organizational structure. Structured data, like a database table, allows for faster processing and retrieval of information due to its predictable and consistent format. Thus, the correct answer is a database table, as it exemplifies the organization and format that defines structured data.
අනෙක් අතට, ව්‍යුහගත නොකළ දත්ත එවැනි පූර්ව නිශ්චිත ආකෘතියක් නොමැති අතර, ගබඩා කිරීම, සෙවීම සහ විශ්ලේෂණය කිරීම වඩාත් අපහසු වේ. ව්‍යුහගත නොවන දත්ත සඳහා උදාහරණ ලෙස ස්ථාවර ආයතනික ව්‍යුහයක් අනුගමනය නොකරන අතින් ලියන ලද සටහන්, සමාජ මාධ්‍ය පළ කිරීම්, විද්‍යුත් තැපැල් පණිවිඩ සහ ඩිජිටල් ඡායාරූප ඇතුළත් වේ. දත්ත සමුදා වගුවක් වැනි ව්‍යුහගත දත්ත, එහි පුරෝකථනය කළ හැකි සහ ස්ථාවර ආකෘතිය හේතුවෙන් තොරතුරු වේගයෙන් සැකසීමට සහ ලබා ගැනීමට ඉඩ සලසයි. මේ අනුව, නිවැරදි පිළිතුර වන්නේ දත්ත සමුදා වගුවක් වේ, එය ව්‍යුහගත දත්ත නිර්වචනය කරන සංවිධානය සහ ආකෘතිය නිදසුන් කරයි.

Therefore, the correct answer is option 2).
එබැවින්, නිවැරදි පිළිතුර විකල්ප 2) වේ.

2. What is the incorrect BCD value?
BCD අගයක් නොවන්නේ කුමක්ද?

1) 10011000 2) 01011001 3) 11001000 4) 10001000 5) 00110111

Answer : 3)

Explanation / පැහැදිලි කිරීම

BCD (Binary-Coded Decimal) is a representation of decimal digits (0–9) using 4-bit binary numbers. Each 4-bit group in a BCD value must range from 0000 (decimal 0) to 1001 (decimal 9) to be valid. If any 4-bit group exceeds this range, the entire value becomes invalid. To determine if a BCD value is correct, each 4-bit segment is examined individually to check whether it falls within the valid range.
BCD (Binary-Coded Decimal) යනු 4-බිට් ද්විමය සංඛ්‍යා භාවිතා කරමින් දශම සංඛ්‍යා 0–9 නිරූපණයකි. BCD අගයක ඇති සෑම 4-බිට් කාණ්ඩයක්ම වලංගු වීමට 0000 (දශම 0) සිට 1001 (දශම 9) දක්වා පරාසයක තිබිය යුතුය. ඕනෑම 4-බිට් කාණ්ඩයක් මෙම පරාසය ඉක්මවා ගියහොත්, සම්පූර්ණ අගය අවලංගු වේ. BCD අගයක් නිවැරදි දැයි තීරණය කිරීම සඳහා, එය වලංගු පරාසය තුළට වැටෙන්නේ දැයි පරීක්ෂා කිරීම සඳහා සෑම 4-බිට් කොටසක්ම තනි තනිව පරීක්ෂා කරනු ලැබේ.

Option 3, 11001000, is invalid because the first 4-bit group, 1100, represents the decimal number 12, which is outside the allowable range for BCD. The other options—10011000, 01011001, 10001000, and 00110111—all have 4-bit groups within the valid range of 0–9, making them correct. Thus, the incorrect BCD value is 11001000, as it does not adhere to the rules of BCD representation.
විකල්පය 3, 11001000, වලංගු නොවන බැවින් පළමු 4-බිට් කාණ්ඩය වන 1100, BCD සඳහා අවසර ලත් පරාසයෙන් පිටත දශම අංක 12 නියෝජනය කරයි. අනෙකුත් විකල්ප - 10011000, 01011001, 10001000 සහ 00110111 - සියල්ලටම (0–9) වලංගු පරාසය තුළ 4-බිට් කාණ්ඩ ඇති අතර, ඒවා නිවැරදි කරයි. මේ අනුව, වැරදි BCD අගය 11001000 වේ, මන්ද එය BCD නිරූපණයේ නීතිවලට අනුකූල නොවන බැවිනි.

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

3. Which of the following combinations of gates can replace a single XOR gate?
තනි XOR ද්වාරයක් ප්‍රතිස්ථාපනය කළ හැක්කේ පහත සඳහන් ද්වාරවල කුමන සංයෝජනවලටද?

1) AND and OR 2) NOT, AND, and OR 3) NAND and OR
4) AND and OR 5) NOR and AND

Answer: 2)

Explanation / පැහැදිලි කිරීම:

The XOR gate output is true (1) only when its two inputs are different. So, to replace a XOR gate using basic logic gates we have to implement that logic to get the correct output.
XOR ද්වාර ප්‍රතිදානය සත්‍ය (1) එහි ආදාන දෙක වෙනස් වූ විට පමණි. ඉතින්, මූලික තාර්කික ද්වාර භාවිතා කරමින් XOR ද්වාරයක් ප්‍රතිස්ථාපනය කිරීම සඳහා නිවැරදි ප්‍රතිදානය ලබා ගැනීම සඳහා අපට එම තර්කය ක්‍රියාත්මක කළ යුතුය.

XOR gate's logical function can be replaced by basic logic gates as follows,
XOR ද්වාරයෙහි තාර්කික ශ්‍රිතය පහත පරිදි මූලික තාර්කික ද්වාර මගින් ප්‍රතිස්ථාපනය කළ හැක.

$$A \oplus B = A\overline{B} + \overline{A}B$$

Therefore, it is clear that XOR gate can be replaced by a combination of AND, OR and NOT gates. The correct answer is option 2).
එබැවින්, XOR ද්වාරය AND, OR සහ NOT ද්වාරවල සංයෝජනයකින් ප්‍රතිස්ථාපනය කළ හැකි බව පැහැදිලිය. නිවැරදි පිළිතුර විකල්ප 2) වේ.

4. Time-sharing systems primarily aim to,
කාල විභජන පද්ධති මූලික වශයෙන් අරමුණු කරන්නේ,

    1) Provide long CPU response / දිගු CPU ප්‍රතිචාරයක් ලබා දීම.
    2) Minimize user tasks / පරිශීලක කාර්යයන් අවම වීම.
    3) Reduce CPU idle time / CPU නිෂ්ක්‍රිය කාලය අඩු වීම.
    4) Support single-user programs / තනි පරිශීලක වැඩසටහන් සඳහා සහාය වීම.
    5) Handle one process / එක් ක්‍රියාවලියක් හැසිරවීම.

Answers : 3)

Let's consider each option one by one.
එක් එක් විකල්පය එකින් එක සලකා බලමු.

1) Provide long CPU response / දිගු CPU ප්‍රතිචාරයක් ලබා දීම.
This is incorrect. Time-sharing systems are designed to provide quick and efficient CPU response times, not long ones. The goal is to ensure that the CPU time is shared among multiple users or processes, allowing for prompt responses.
මෙය වැරදියි. කාල විභජන පද්ධති සැලසුම් කර ඇත්තේ ඉක්මන් සහ කාර්යක්ෂම CPU ප්‍රතිචාර කාලයන් සැපයීමට මිස දීර්ඝ කාලයක් නොවේ. ඉලක්කය වන්නේ CPU කාලය බහු පරිශීලකයින් හෝ ක්‍රියාවලින් අතර බෙදාගෙන ඇති බව සහතික කිරීම, ඉක්මන් ප්‍රතිචාර සඳහා ඉඩ ලබා දීමයි.

2.) Minimize user tasks / පරිශීලක කාර්යයන් අවම වීම.
This is not the primary goal of time-sharing systems. Time-sharing systems are more focused on efficiently managing CPU time rather than minimizing the number of tasks a user must handle.
කාල විභජන පද්ධතිවල මූලික අරමුණ මෙය නොවේ. කාල විභජන පද්ධති වැඩි අවධානයක් යොමු කරන්නේ පරිශීලකයෙකු විසින් හැසිරවිය යුතු කාර්යයන් ගණන අවම කිරීමට වඩා CPU කාලය කාර්යක්ෂමව කළමනාකරණය කිරීම කෙරෙහි ය.

3) Reduce CPU idle time / CPU නිෂ්ක්‍රිය කාලය අඩු වීම.
This is correct. Time-sharing systems aim to minimize CPU idle time by rapidly switching between tasks and users. This ensures that the CPU is utilized efficiently, keeping it busy and reducing the periods when it is not performing any useful work.
මෙය නිවැරදියි. කාර්ය සහ පරිශීලකයන් අතර වේගයෙන් මාරු වීමෙන් CPU නිෂ්ක්‍රිය කාලය අවම කිරීම කාල විභජන පද්ධති අරමුණු කරයි. මෙමගින් CPU කාර්යක්ෂමව භාවිතා කිරීම සහතික කරයි, එය කාර්යබහුලව තබා ගැනීම සහ එය කිසිදු ප්‍රයෝජනවත් කාර්යයක් සිදු නොකරන කාල සීමාවන් අඩු කරයි.

4) Support single-user programs /තනි පරිශීලක වැඩසටහන් සඳහා සහය වීම.
This is incorrect. Time-sharing systems are designed to support multiple users and processes simultaneously. They are intended to manage and share resources among several users rather than focusing on single-user programs.
මෙය වැරදියි. කාල විභජන පද්ධති සැලසුම් කර ඇත්තේ බහු පරිශීලකයින්ට සහ ක්‍රියාවලින්ට එකවර සහය දැක්වීමටය. ඒවා තනි-පරිශීලක වැඩසටහන් කෙරෙහි අවධානය යොමු කරනවාට වඩා පරිශීලකයින් කිහිප දෙනෙකු අතර සම්පත් කළමනාකරණය කිරීමට සහ බෙදා ගැනීමට අදහස් කෙරේ.

5) Handle one process / එක් ක්‍රියාවලියක් හැසිරවීම.
This is incorrect. Time-sharing systems are specifically designed to handle multiple processes and users at the same time. They manage the execution of several tasks concurrently by dividing CPU time among them.
මෙය වැරදියි. කාල විභජන පද්ධති විශේෂයෙන් නිර්මාණය කර ඇත්තේ බහු ක්‍රියාවලි සහ පරිශීලකයන් එකවර හැසිරවීමටය. ඔවුන් අතර CPU කාලය බෙදීම මගින් කාර්යයන් කිහිපයක් සමගාමීව ක්‍රියාත්මක කිරීම කළමනාකරණය කරයි.

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

5. In, ………… allocation files are stored in non-contiguous blocks, with each block containing a pointer to the next block in the file.
…………… ගොනු විභජනයේදී, ගොනු යාබද නොවන කෝෂ වල ගබඩා කර ඇති අතර, සෑම කෝෂයකම ගොනුවේ මීළඟ කොටස වෙත දර්ශකයක් අඩංගු වේ.

1) Contiguous යාබද
2) Indexed සුචක
3) Sequential අනුක්‍රමික
4) Linked සබැඳි
5) Direct සෘජු

Answer : 4)

File allocation methods are techniques used by operating systems to store files on disk storage.
ගොනු විභජන ක්‍රම යනු තැටි ගබඩාවේ ගොනු ගබඩා කිරීම සඳහා මෙහෙයුම් පද්ධති භාවිතා කරන තාක්ෂණික ක්‍රම වේ.

1. Contiguous Allocation / යාබද ගොනු විභාජනය
In contiguous allocation, a file is stored in blocks that are next to each other on the disk. This method can lead to fragmentation over time, making it hard to find large blocks of free space, which limits the file size to the available contiguous space.
යාබද ගොනු විභජනයේ දී, ගොනුවක් තැටියේ එකිනෙකට යාබදව ඇති මතක කාණ්ඩ වල ගබඩා කර ඇත. මෙම ක්‍රමය කාලයත් සමඟ ඛණ්ඩනය වීමට හේතු විය හැකි අතර, විශාල නිදහස් ඉඩ කොටස් සොයා ගැනීමට අපහසු වන අතර එමඟින් ගොනුවක ප්‍රමාණය, පවතින යාබද ඉඩ ප්‍රමාණයට සීමා කරයි.

2. Indexed Allocation / සුචක ගොනු විභාජනය
In indexed allocation, a file has an index block that lists the locations of all its data blocks. The blocks can be scattered across the disk, preventing external fragmentation and allowing for more flexible storage.
සුචක ගොනු විභජනයේ දී, ගොනුවක එහි සියලුම දත්ත කොටස්වල ස්ථාන ලැයිස්තුගත කරන දර්ශක මතක කාණ්ඩ එකක් ඇත. බ්ලොක් තැටිය පුරා විසිරී ඇති අතර, බාහිර ඛණ්ඩනය වැළැක්වීම සහ වඩාත් නම්‍යශීලීව ගබඩා කිරීමට ඉඩ සලසයි.

3. Linked Allocation / සබැඳි ගොනු විභාජනය
In linked allocation, files are stored as a chain of blocks that can be scattered across the disk. Each block points to the next, allowing files to grow easily and avoiding the need for contiguous space.
සබැඳි ගොනු විභජනයේ දී, ගොනු තැටිය පුරා විසිරී යා හැකි මතක කාණ්ඩ දාමයක් ලෙස ගබඩා කර ඇත. සෑම මතක කාණ්ඩයක්ම ඊළඟ මතක කාණ්ඩයට යොමු කරයි, ලිපිගොනු පහසුවෙන් වර්ධනය වීමට ඉඩ සලසයි සහ අඛණ්ඩ ඉඩ අවශ්‍යතාවයක් නැත.

We can fill the blank as follows.
අපට පහත පරිදි හිස්තැන පිරවිය හැකිය.

In, linked allocation files are stored in non-contiguous blocks, with each block containing a pointer to the next block in the file.
සබැඳි ගොනු විභාජනයේදී, ගොනු යාබද නොවන කෝෂ වල ගබඩා කර ඇති අතර, සෑම බ්ලොක් එකකම ගොනුවේ මීළඟ කොටස වෙත දර්ශකයක් අඩංගු වේ.

Therefore the correct answer is, **4) Linked**
එබැවින් නිවැරදි පිළිතුර වන්නේ, **4) සබැඳි**

6. What is the name for building relationships with other entities considering a relationship between several entities as one entity?
භූතාර්ථ කිහිපයක් අතර වූ සම්බන්ධයක් එක් භූතාර්ථයක් ලෙස සලකා වෙනත් භූතාර්ථයක් සමග සම්බන්ධතා ගොඩ නැගීම හඳුන්වන නම කුමක්ද?

1) Generalization / සාමාන්‍යකරණය
2) Specialization / විශේෂිකරණය
3) Union / සංධානගත වීම
4) Aggregation / සංයුක්තකරණය
5) Mapping / අනුරූපකරණය

Answer: 4)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

1) Generalization / සාමාන්‍යකරණය
Generalization is the process of combining multiple similar entities into a higher-level entity (ex: "Car" and "Truck" become "Vehicle").
සාමාන්‍යකරණය යනු බහු සමාන භූතාර්ථ ඉහළ මට්ටමේ භූතාර්ථයකට ඒකාබද්ධ කිරීමේ ක්‍රියාවලියයි (උදා: මෝටර් රථය සහ ට්‍රක් රථය වාහනය බවට පත්වේ).

2) Specialization / විශේෂිකරණය
Specialization is the process of creating sub-entities from a general entity (ex: "Vehicle" is specialized into "Car" and "Truck").
විශේෂිකරණය යනු සාමාන්‍ය භූතාර්ථයකින් උප භූතාර්ථ නිර්මාණය කිරීමේ ක්‍රියාවලියයි (උදා: වාහනය මෝටර් රථය සහ ට්‍රක් රථය ලෙස විශේෂිත කර ඇත).

3) Union / සංධානගත වීම
Union is not an EER concept. however, in databases, it refers to combining results from multiple tables or queries.
සංධානගත වීම EER සංකල්පයක් නොවේ කෙසේ වෙතත්, දත්ත සමුදායන්හි, එය බහු වගු හෝ විමසුම් වලින් ප්‍රතිඵල ඒකාබද්ධ කිරීම අදහස් කරයි.

4) Aggregation / සංයුක්තකරණය
Aggregation is a concept in the Enhanced Entity-Relationship (EER) model, where a relationship between multiple entities is treated as a single higher-level entity.
සංයුක්තකරණය යනු වැඩිදියුණු කළ භූතාර්ථ-සම්බන්ධතා (EER) ආකෘතියේ සංකල්පයක් වන අතර, එහිදී බහු භූතාර්ථ අතර සම්බන්ධතාවයක් තනි ඉහළ මට්ටමේ භූතාර්ථයක් ලෙස සලකනු ලැබේ.

This allows for building relationships with other entities by considering a relationship itself as an entity in another relationship. It is commonly used when one entity needs to participate in multiple relationships.
සම්බන්ධතාවයක් තවත් සම්බන්ධතාවයක භූතාර්ථයක් ලෙස සලකා බැලීමෙන් අනෙකුත් භූතාර්ථ සමඟ සබඳතා ගොඩනඟා ගැනීමට මෙය ඉඩ සලසයි. එක් භූතාර්ථයකට බහු සම්බන්ධතාවලට සහභාගී වීමට අවශ්‍ය වූ විට එය බහුලව භාවිතා වේ.

5) Mapping / අනුරූපකරණය
Mapping is used in conceptual database design to define relationships between entities and attributes, but it does not treat relationships as entities.
සංකල්පීය දත්ත සමුදා නිර්මාණයේදී භූතාර්ථ සහ ගුණාංග අතර සම්බන්ධතා නිර්වචනය කිරීමට අනුරූපකරණය භාවිතා කරයි, නමුත් එය සම්බන්ධතා භූතාර්ථ ලෙස සලකන්නේ නැත.

Therefore, the correct answer is 4).
එබැවින්, නිවැරදි පිළිතුර වන්නේ 4) වේ.

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

```
def test(value):
    if value % 2 == 0:
        return [x**2 for x in range(value)]
    else:
        return {x: x**2 for x in range(value)}

print(test(3))
```

1) [1,4,9]
2) {0:0,1:1}
3) {0:0, 1:1, 2:4, 3:9}
4) {0: 0, 1: 1, 2: 4}
5) None

Answer: 4)

Explanation / පැහැදිලි කිරීම:

The Python code defines a function test(value) that behaves differently based on whether the input value is even or odd.
පයිතන් කේතය ආදාන value ඉරට්ටේ හෝ ඔත්තේ ද යන්න මත පදනම්ව වෙනස් ලෙස හැසිරෙන ශ්‍රිත test(value) නිර්වචනය කරයි.

If value is even, the function returns a list, generating a list of squares of integers from 0 to value - 1.
value ඉරට්ටේ නම්, ශ්‍රිතය ලැයිස්තුවක් ලබා දෙයි, 0 සිට value – 1 දක්වා පූර්ණ සංඛ්‍යා වර්ග ලැයිස්තුවක් ජනනය කරයි.

If value is odd, the function returns a dictionary, creating a dictionary where each key is an integer from 0 to value - 1, and its value is the square of that integer.
value ඔත්තේ නම්, ශ්‍රිතය dictionary එකක් ලබා දෙයි, එක් එක් යතුර 0 සිට value - 1 දක්වා නිඛිලයක් වන dictionary එකක් නිර්මාණය කරයි, සහ එහි අගය එම නිඛිලයේ වර්ග වේ.

When test(3) is called, the input 3 is odd, so the function executes the else branch and returns the dictionary {x: x**2 for x in range(3)}.
test(3) හැඳින්වූ විට, ආදානය 3 ඔත්තේ වේ, එබැවින් ශ්‍රිතය වෙනත් ශාඛාව ක්‍රියාත්මක කර dictionary එක {x: x**2 for x in range(3)} ලබා දෙයි.

The range range(3) generates integers 0, 1, 2 resulting in the dictionary {0:0, 1:1, 2:4}. This dictionary is printed as the output.
range(3) පරාසය නිඛිල 0, 1, 2 ජනනය කරන අතර එහි ප්‍රතිඵලයක් ලෙස {0:0, 1:1, 2:4} dictionary එක ලබා දෙයි. මෙම dictionary එක ප්‍රතිදානය ලෙස මුද්‍රණය කර ඇත.

Therefore, the correct answer is 4).
එබැවින්, නිවැරදි පිළිතුර 4) වේ.

8. What is the main difference between a list and a tuple in Python?
පයිතන්හි ලැයිස්තුවක් සහ tuple එකක් අතර ඇති ප්‍රධාන වෙනස කුමක්ද?

1) Lists are immutable, while tuples are mutable.
ලැයිස්තු වෙනස් කළ නොහැකි අතර tuple එකක් වෙනස් කළ හැක.
2) Lists are mutable, while tuples are immutable.
ලැයිස්තු වෙනස් කළ හැකි අතර tuple එකක් වෙනස් කළ නොහැක.
3) Lists can only store numbers, while tuples can store any type of data.
ලැයිස්තු වලට සංඛ්‍යා පමණක් ගබඩා කළ හැකි අතර, tuple වලට ඕනෑම ආකාරයක දත්ත ගබඩා කළ හැක.
4) There is no difference between lists and tuples
list සහ tuple එකක් අතර වෙනසක් නැත.
5) None of the above
ඉහත කිසිවක් නොවේ

Answer: 2)

Explanation / පැහැදිලි කිරීම:

In Python, the main difference between lists and tuples lies in their mutability:
Python හි, ලැයිස්තු සහ tuples අතර ප්‍රධාන වෙනස පවතින්නේ ඒවායේ වෙනස් වීමේ/නොවීමේ හැකියාව තුලය

Lists / ලැයිස්තු

- Mutable. Their contents can be changed after creation.
වෙනස් කළ හැකි ය. නිර්මාණය කිරීමෙන් පසු ඒවායේ අන්තර්ගතය වෙනස් කළ හැකිය.

- You can add, remove, or modify elements in a list.
ඔබට ලැයිස්තුවක ඇති මූලද්‍රව්‍ය එක් කිරීමට, ඉවත් කිරීමට හෝ වෙනස් කිරීමට හැකිය.

Tuples

- Immutable. Once created, their contents cannot be changed.
වෙනස් කළ නොහැකි ය. නිර්මාණය කළ පසු, ඒවායේ අන්තර්ගතය වෙනස් කළ නොහැක.

- You cannot add, remove, or modify elements in a tuple.
ඔබට tuple එකක මූලද්‍රව්‍ය එකතු කිරීමට, ඉවත් කිරීමට හෝ වෙනස් කිරීමට නොහැක.

The correct answer is 2) Lists are mutable, while tuples are immutable.
නිවැරදි පිළිතුර 2) ලැයිස්තු වෙනස් කළ හැකි අතර tuples වෙනස් කළ නොහැකි ය.

Other options / වෙනත් විකල්ප:

Option / විකල්ප 1)
This is incorrect because it's the opposite of the truth.
මෙය සත්‍යයට ප්‍රතිවිරුද්ධ බැවින් මෙය වැරදිය.

Option / විකල්ප 3)
Both lists and tuples can store any type of data (ex: integers, strings, objects, etc.).
ලැයිස්තු සහ tuple යන දෙකටම ඕනෑම ආකාරයක දත්ත ගබඩා කළ හැකිය (උදා: පූර්ණ සංඛ්‍යා, strings, වස්තු, ආදිය).

Option / විකල්ප 4)
This is incorrect, because their mutability is a key difference.
මෙය වැරදියි, මන්ද ඒවායේ වෙනස් වීමේ/නොවීමේ ප්‍රධාන වෙනසක් වේ.

Option / විකල්ප 5)
Not applicable since option 2 is correct.
විකල්ප 2 නිවැරදි බැවින් අදාළ නොවේ.

9. What happens when you execute the following Python code?
ඔබ පහත පයිතන් කේතය ක්‍රියාත්මක කළ විට කුමක් සිදුවේද?

```
x = "3.14"
y = int(x)
```

1) y is assigned the value 3 / y ට 3 හි අගය පවරනු ලැබේ.
2) y is assigned the value 314 / y ට 314 හි අගය පවරනු ලැබේ.
3) A ValueError is raised / ValueError එකක් මතු වේ.
4) y is assigned the value 3.14 as an integer / y ට 3.14 හි අගය පූර්ණ සංඛ්‍යාවක් ලෙස පවරා ඇත.
5) The code executes, but y remains undefined.
කේතය ක්‍රියාත්මක වේ, නමුත් y හි අගය නිර්වචනය කර නොමැත.

Answer: 3)

Explanation / පැහැදිලි කිරීම:

The variable x is a string containing the value "3.14".
x විචල්‍යය යනු "3.14" අගය අඩංගු string එකකි.

The int() function is used to convert strings to integers. However, it only works if the string represents a valid integer (ex: "3").
strings පූර්ණ සංඛ්‍යා බවට පරිවර්තනය කිරීමට int() ශ්‍රිතය භාවිතා කරයි. කෙසේ වෙතත්, එය ක්‍රියාත්මක වන්නේ string එක වලංගු පූර්ණ සංඛ්‍යාවක් නියෝජනය කරන්නේ නම් පමණි (උදා: "3").

Since "3.14" represents a floating-point number and not an integer, attempting to convert it to an integer using int() raises a ValueError.
"3.14" නියෝජනය කරන්නේ දශම සංඛ්‍යාවක් මිස පූර්ණ සංඛ්‍යාවක් නොවන බැවින්, එය int() භාවිතයෙන් පූර්ණ සංඛ්‍යාවක් බවට පරිවර්තනය කිරීමට උත්සාහ කිරීම ValueError එකක් මතු කරයි.

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

10. How do you specify a default background color for a webpage using HTML?
HTML භාවිතයෙන් වෙබ් පිටුවක් සඳහා පෙරනිමි පසුබිම් වර්ණයක් නියම කරන්නේ කෙසේද?

1) By adding a `background` tag in the `<body>` section.
`<body>` කොටසේ `background` උසුලනය එක් කිරීමෙන්.

2) By linking a stylesheet with the `color` attribute.
`color` ගුණාංගය සමඟ මෝස්තර පත්‍රිකාවක් සම්බන්ධ කිරීමෙන්.

3) By using the `bgcolor` attribute in the `<body>` tag.
`<body>` උසුලනයේ `bgcolor` ගුණාංගය භාවිතා කිරීමෙන්.

4) By adding `background-color` to the `<head>` tag.
`<head>` උසුලනයට `background-color` එකතු කිරීමෙන්.

5) By using `<background>` tags around the content.
අන්තර්ගතය වටා `<background>` උසුලන භාවිතා කිරීමෙන්.

Answer: 3)

Explanation / පැහැදිලි කිරීම:

In HTML, you can specify the default background color of a webpage using the `bgcolor` attribute inside the `<body>` tag.
HTML හි, ඔබට `<body>` උසුලනය තුළ ඇති `bgcolor` ගුණාංගය භාවිතයෙන් වෙබ් පිටුවක පෙරනිමි පසුබිම් වර්ණය නියම කළ හැක.

Example / උදාහරණය:

```
<html>
<body bgcolor="lightblue">
  <h1>Welcome!</h1>
</body>
</html>
```

The correct answer is 3) By using the `bgcolor` attribute in the `<body>` tag.
නිවැරදි පිළිතුර 3) `<body>` උසුලනයේ `bgcolor` ගුණාංගය භාවිතා කිරීමෙන් ය.