සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved]

Ravindu Bandaranayake - ictfromabc

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 384 MARKING

2025/12/29
ictfromabc
Ravindu Bandaranayake

1. Which port is commonly used for connecting high-definition displays and supports both audio and video signals?
   අධි-විභේදන තිර සම්බන්ධ කිරීමට සහ ශ්‍රව්‍ය සහ දෘශ්‍ය සංඥා යන දෙකටම සහය දැක්වීම සඳහා බහුලව භාවිතා වන කෙවෙනිය කුමක්ද?

   1) USB-C 2) DVI 3) DisplayPort 4) PS/2 5) Serial port

   Answer : 3)

   Let's go through the options provided as answers.
   පිළිතුරු ලෙස සපයා ඇති විකල්ප හරහා යමු.

   1) USB-C
   According to this option, it is false. USB-C is versatile and supports both data transfer and video signals. However, it is not specifically designed for high-definition displays. While it can transmit audio and video, it is primarily used for other purposes like charging and connecting devices.
   මෙම විකල්පය අනුව, එය අසත්‍ය වේ. USB-C බහුකාර්ය වන අතර දත්ත හුවමාරුව සහ වීඩියෝ සංඥා යන දෙකටම සහය දක්වයි. කෙසේ වෙතත්, එය අධි-විභේදන සංදර්ශක සඳහා විශේෂයෙන් නිර්මාණය කර නැත. එයට ශ්‍රව්‍ය සහ දෘශ්‍ය සම්ප්‍රේෂණය කළ හැකි අතර, එය මූලික වශයෙන් උපාංග ආරෝපණය කිරීම සහ සම්බන්ධ කිරීම වැනි වෙනත් අරමුණු සඳහා භාවිතා වේ.

   2) DVI
   According to this option, it is false. DVI (Digital Visual Interface) is used for transmitting video signals but does not support audio. It is primarily for video output and does not fully meet the needs for high-definition displays with both video and audio signals.
   මෙම විකල්පය අනුව, එය අසත්‍ය වේ. DVI ඩිජිටල් දෘශ්‍ය අතුරුමුහුණත) වීඩියෝ සංඥා සම්ප්‍රේෂණය කිරීම සඳහා භාවිතා කරන නමුත් ශ්‍රව්‍ය සඳහා සහය නොදක්වයි. එය මූලික වශයෙන් වීඩියෝ ප්‍රතිදානය සඳහා වන අතර වීඩියෝ සහ ශ්‍රව්‍ය සංඥා යන දෙකම සහිත අධි-විභේදන සංදර්ශක සඳහා අවශ්‍යතා සම්පූර්ණයෙන්ම සපුරාලන්නේ නැත.

   3) DisplayPort
   According to this option, it is true. DisplayPort is designed to connect high-definition displays and supports both audio and video signals. It is commonly used in modern computers and monitors to transmit high-definition content, making it the correct answer.
   මෙම විකල්පය අනුව, එය සත්‍ය වේ. DisplayPort නිර්මාණය කර ඇත්තේ අධි-විභේදන සංදර්ශක සම්බන්ධ කිරීමට සහ ශ්‍රව්‍ය සහ දෘශ්‍ය සංඥා යන දෙකටම සහය දක්වයි. එය අධි-විභේදන අන්තර්ගතය සම්ප්‍රේෂණය කිරීම සඳහා නවීන පරිගණක සහ මොනිටරවල බහුලව භාවිතා වන අතර එය නිවැරදි පිළිතුර බවට පත් කරයි.

   4) PS/2
   According to this option, it is false. PS/2 is a legacy connector used for keyboards and mice. It does not transmit video or audio signals and is not related to high-definition displays.
   මෙම විකල්පය අනුව, එය අසත්‍ය වේ. PS/2 යනු යතුරුපුවරු සහ මූසික සඳහා භාවිතා කරන ලද උරුම සම්බන්ධකයකි. එය වීඩියෝ හෝ ශ්‍රව්‍ය සංඥා සම්ප්‍රේෂණය නොකරන අතර අධි-විභේදන සංදර්ශකවලට සම්බන්ධ නොවේ.

   5) Serial port
   According to this option, it is false. A serial port is mainly used for communication between devices and does not handle video or audio signals, making it irrelevant to connecting high-definition displays.
   මෙම විකල්පය අනුව, එය අසත්‍ය වේ. අනුක්‍රමික කෙවෙනියක් ප්‍රධාන වශයෙන් උපාංග අතර සන්නිවේදනය සඳහා භාවිතා වන අතර වීඩියෝ හෝ ශ්‍රව්‍ය සංඥා හසුරුවන්නේ නැත, එය අධි-විභේදන සංදර්ශක සම්බන්ධ කිරීමට අදාළ නොවේ.

2. Which of the following statements is true about 2's complement?
2හි අනුපූරක සම්බන්ධයෙන් සත්‍ය වන්නේ පහත කුමන ප්‍රකාශයද?

1) Used in first generation computers to perform logical operations.
තාර්කික මෙහෙයුම් සිදු කිරීම සඳහා පළමු පරම්පරාවේ පරිගණකවල භාවිතා විය.

2) Addition and subtraction are two different operations.
එකතු කිරීම සහ අඩු කිරීම එකිනෙකට වෙනස් මෙහෙයුම් දෙකක් ලෙස සිදු කෙරේ.

3) An extra bit is used to represent the sign.
ලකුණ නිරූපනය කිරීමට අමතර බිටුවක් භාවිතා කෙරේ.

4) <u>It helps to develop faster, cheaper hardware to perform arithmetic operations.</u>
අංක ගණිත මෙහෙයුම් ඉටු කිරීම සඳහා වඩා වේගවත්, මිල අඩු දෘඩාංග නිපදවීමට ඉවහල් වේ.

5) Usually represented by the hexadecimal number system.
සාමාන්‍යයෙන් ෂඩ් දශම සංඛ්‍යා පද්ධතිය මගින් නිරූපණය කෙරේ.

Answer : 4)

True, 2's complement is widely used in computer systems to represent signed integers. One of its key advantages is that it allows the same hardware circuits to be used for both addition and subtraction, simplifying the design and making it faster and cheaper to implement.
ඇත්ත වශයෙන්ම, 2 හි අනුපූරකය පරිගණක පද්ධතිවල ලකුණුවත් කරන ලද පූර්ණ සංඛ්‍යා නියෝජනය කිරීම සඳහා බහුලව භාවිතා වේ. එහි එක් ප්‍රධාන වාසියක් නම්, එය එකතු කිරීම සහ අඩු කිරීම යන දෙකටම එකම දෘඩාංග පරිපථ භාවිතා කිරීමට ඉඩ සලසා දීම, සැලසුම සරල කිරීම සහ එය ක්‍රියාත්මක කිරීම වේගවත් හා ලාභදායී කිරීමයි.

Why the Other Options are Incorrect,
වෙනත් විකල්ප වැරදි ඇයි,

- **Option / විකල්පය 1)**
False, 2's complement was not used in first-generation computers, which often used different methods for representing signed numbers. It became more standard in later generations.
අසත්‍ය වේ, 2 හි අනුපූරකය පළමු පරම්පරාවේ පරිගණකවල භාවිතා නොකළ අතර, බොහෝ විට ලකුණුවත් කරන ලද අංක නියෝජනය කිරීම සඳහා විවිධ ක්‍රම භාවිතා කරන ලදී. පසු පරම්පරාවල එය වඩාත් සම්මත විය.

- **Option / විකල්පය 2)**
False, In 2's complement arithmetic, subtraction can be performed using addition by adding the 2's complement of the number, which simplifies the hardware design.
අසත්‍ය වේ, 2 හි අනුපූරක අංක ගණිතයේදී, දෘඩාංග නිර්මාණය සරල කරන සංඛ්‍යාවේ 2 හි අනුපූරකය එකතු කිරීමෙන් එකතු කිරීම භාවිතයෙන් අඩු කිරීම සිදු කළ හැක.

- **Option / විකල්පය 3)**
False, In 2's complement representation, the most significant bit (MSB) doubles as the sign bit and as part of the magnitude of the number. There's no separate "extra" bit for the sign.
අසත්‍ය වේ, 2 හි අනුපූරක නිරූපණයෙහි, වඩාත්ම සැලකිය යුතු බිටු (MSB) සංකේත බිටු ලෙස සහ සංඛ්‍යාවේ විශාලත්වයේ කොටසක් ලෙස දෙගුණ වේ. ලකුණ සඳහා වෙනම "අමතර" බිට් එකක් නොමැත.

- **Option / විකල්පය 5)**
False, 2's complement numbers are typically represented in binary. Hexadecimal can be used to represent any binary number compactly, but it is not specifically tied to 2's complement.
අසත්‍ය වේ, 2 හි අනුපූරක සංඛ්‍යා සාමාන්‍යයෙන් ද්විමය වශයෙන් නිරූපණය කෙරේ. ඕනෑම ද්විමය සංඛ්‍යාවක් සංයුක්තව නිරූපණය කිරීමට ෂඩ් දශම භාවිතා කළ හැකි නමුත් එය 2 හි අනුපූරකයට නිශ්චිතව බැඳී නැත.

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

3. Which logic gates are typically used in the implementation of a half adder?
අර්ධ ආකලකය ක්‍රියාත්මක කිරීමේදී සාමාන්‍යයෙන් භාවිතා වන තාර්කික ද්වාර මොනවාද?

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

Answer : 2)

A half adder is a basic digital circuit used to add two single-bit binary numbers (A and B). It has two outputs.
අර්ධ ආකලකය යනු තනි-බිටි ද්විමය අංක දෙකක් A සහ B එකතු කිරීම සඳහා භාවිතා කරන මූලික ඩිජිටල් පරිපථයකි. එහි නිමැවුම් දෙකක් ඇත.

- **Sum (S):** The result of the addition $(A \oplus B)$.
  එකතුව (S): එකතු කිරීමේ ප්‍රතිඵලය - $(A \oplus B)$
- **Carry (C):** The carry generated from the addition (A . B).
  රැගෙන යාම (C): එකතු කිරීමෙන් ජනනය වන රැගෙන යාම - (A . B)

So the answer is "2) AND and XOR gates".
එම නිසා පිළිතුර වන්නේ "2) AND and XOR gates" වේ.

4. Which of the following is the primary goal of database normalization?
දත්ත සමුදාය ප්‍රමතකරණය කිරීමේ මූලික ඉලක්කය පහත සඳහන් දේවලින් කුමක් ද?

1) Minimizing the number of tables in the database.
දත්ත ගබඩාවේ ඇති වගු ගණන අවම කිරීම.
2) Ensuring that all data is stored in a single table.
සියලුම දත්ත තනි වගුවක ගබඩා කර ඇති බව සහතික කිරීම.
3) Eliminating data redundancy and ensuring data integrity.
දත්ත අතිරික්තය ඉවත් කිරීම සහ දත්ත අඛණ්ඩතාව සහතික කිරීම.
4) Speed up query performance by denormalizing tables.
වගු විකෘති කිරීම මගින් විමසුම් කාර්ය සාධනය වේගවත් කිරීම.
5) Ensuring that each table has only one column.
සෑම වගුවකටම ඇත්තේ එක් තීරුවක් පමණක් බව සහතික කිරීම.

Answer : 3)

The primary goal of database normalization is 3) Eliminating data redundancy and ensuring data integrity.
දත්ත සමුදාය ප්‍රමතකරණයේ මූලික ඉලක්කය වන්නේ 3) දත්ත අතිරික්තය ඉවත් කිරීම සහ දත්ත අඛණ්ඩතාව සහතික කිරීමයි.

Database normalization is a process used to organize a database into tables and columns to minimize redundancy and dependency. The purpose is to reduce data redundancy and ensure data integrity by organizing data logically.
දත්ත සමුදාය ප්‍රමතකරණය යනු අතිරික්තය සහ යැපීම අවම කිරීම සඳහා දත්ත සමුදායක් වගු සහ තීරු ලෙස සංවිධානය කිරීමට භාවිතා කරන ක්‍රියාවලියකි. පරමාර්ථය වන්නේ දත්ත අතිරික්තය අඩු කිරීම සහ දත්ත තාර්කිකව සංවිධානය කිරීමෙන් දත්ත අඛණ්ඩතාව සහතික කිරීමයි.

A brief explanation of why the other options are not the primary goal of database normalization:
දත්ත සමුදාය ප්‍රමතකරණයේ මූලික ඉලක්කය අනෙකුත් විකල්ප නොවන්නේ මන්දැයි කෙටි පැහැදිලි කිරීමක්:

Statement 1 (ප්‍රකාශය 1)
Normalization often leads to the creation of more tables, not fewer, as data is separated into distinct, related tables to reduce redundancy.
ප්‍රමතකරණය බොහෝ විට වැඩි වගු සෑදීමට මඟ පාදයි, අඩු නොවේ, මන්ද අතිරික්තය අඩු කිරීම සඳහා දත්ත එකිනෙකට වෙනස්, අදාළ වගු වලට වෙන් කරනු ලැබේ.

Statement 2 (ප්‍රකාශය 2)
This goes against the principles of normalization. Storing all data in a single table can lead to redundancy and difficulties in maintaining data integrity.

මෙය ප්‍රමතකරණයේ මූලධර්මවලට පටහැනි ය. සියලුම දත්ත තනි වගුවක ගබඩා කිරීම අතිරික්තතාවයට සහ දත්ත අඛණ්ඩතාව පවත්වා ගැනීමේ දුෂ්කරතාවයන්ට හේතු විය හැක.

Statement 4 (ප්‍රකාශය 4)
Denormalization is sometimes used to improve query performance, but it's not part of normalization. Normalization can sometimes make querying less efficient, and denormalization is a separate process used to optimize performance in some cases.
ප්‍රමතකරණය සමහර විට විමසුම් කාර්ය සාධනය වැඩි දියුණු කිරීමට භාවිතා කරයි, නමුත් එය ප්‍රමතකරණයේ කොටසක් නොවේ. ප්‍රමතකරණය සමහර විට විමසීම් කාර්යක්ෂමතාවයෙන් අඩු කරයි, සහ ප්‍රති-ප්‍රමතකරණය යනු සමහර අවස්ථාවල කාර්ය සාධනය ප්‍රශස්ත කිරීම සඳහා භාවිතා කරන වෙනම ක්‍රියාවලියකි.

Statement 5 (ප්‍රකාශය 5)
This is not a goal of normalization. Normalized tables typically have multiple columns, with each column representing a specific attribute of the data.
මෙය ප්‍රමතකරණයේ ඉලක්කයක් නොවේ. ප්‍රමතකරණය කරන ලද වගු වල සාමාන්‍යයෙන් බහුවිධ තීරු ඇත, සෑම තීරුවක්ම දත්තවල නිශ්චිත ගුණාංගයක් නියෝජනය කරයි.

5. Which normalization form eliminates transitive dependencies?
සංක්‍රාන්ති පරායත්තතා ඉවත් කරන ප්‍රමතකරණ ආකාරය කුමක්ද?

1) 1NF 2) 2NF 3) 3NF 4) BCNF 5) 4NF

Answer: 3)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

1) **1NF**
Ensures that the table has no repeating groups or arrays. Each column must have atomic values. It does not address transitive dependencies.
වගුවේ පුනරාවර්තන කණ්ඩායම් හෝ අරා නොමැති බව සහතික කරයි. සෑම තීරුවකම පරමාණුක අගයන් තිබිය යුතුය. එය සංක්‍රාන්ති පරායක්තතා ආමන්ත්‍රණය නොකරයි.

2) **2NF**
A table is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependent on the entire primary key. It does not eliminate transitive dependencies.
වගුවක් 1NF හි තිබේ නම් එය 2NF හි ඇති අතර සියලුම ප්‍රාථමික නොවන ගුණාංග සම්පූර්ණ ප්‍රාථමික යතුර මත සම්පූර්ණයෙන්ම ක්‍රියාකාරීව රඳා පවතී. එය සංක්‍රාන්ති පරායක්තතා ඉවත් නොකරයි.

3) **3NF**
A table is in 3NF if it is in 2NF and every non-prime attribute is only dependent on the primary key. Eliminates transitive dependencies.
වගුවක් 2NF හි තිබේ නම් 3NF හි ඇති අතර සෑම ප්‍රාථමික නොවන ගුණාංගයක්ම ප්‍රාථමික යතුර මත පමණක් රඳා පවතී. සංක්‍රාන්ති පරායක්තතා ඉවත් කරයි.

4) **BCNF**
Addresses certain anomalies not covered by 3NF (especially when there are overlapping candidate keys). It does not specifically focus on transitive dependencies, as 3NF already eliminates them.
3NF මගින් ආවරණය නොවන ඇතැම් විෂමතා ආමන්ත්‍රණය කරයි (විශේෂයෙන් අතිච්ඡාදනය වන අපේක්ෂක යතුරු ඇති විට). 3NF දැනටමත් ඒවා ඉවත් කර ඇති බැවින් එය සංක්‍රාන්ති පරායක්තතා කෙරෙහි විශේෂයෙන් අවධානය යොමු නොකරයි.

5) **4NF**
A table is in 4NF if it is in BCNF and does not have any multivalued dependencies. It does not address transitive dependencies, as those are already removed in 3NF.
වගුවක් BCNF හි තිබේ නම් සහ බහු අගය කළ පරායක්තතා නොමැති නම් 4NF හි ඇත. 3NF හි දැනටමත් ඉවත් කර ඇති බැවින් එය සංක්‍රාන්ති පරායක්තතා ආමන්ත්‍රණය නොකරයි.

The correct answer is 3) 3NF (Third Normal Form).
නිවැරදි පිළිතුර 3) 3NF (තුන්වන ප්‍රමත අවස්ථාව) වේ.

6. Which SQL operator is used to match a pattern in a string?
string එකක රටාවක් ගැළපීමට භාවිතා කරන්නේ කුමන SQL මෙහෙයවනයද?

1) LIKE 2) MATCH 3) FIND 4) SEARCH 5) SIMILAR

Answer: 1)

Explanation / පැහැදිලි කිරීම:

The LIKE operator in SQL is used to match patterns in strings. It allows you to search for specific patterns using wildcard characters:
SQL හි ඇති LIKE මෙහෙයවනය strings වල රටා ගැලපීමට භාවිතා කරයි. එය Wildcard අක්ෂර භාවිතයෙන් නිශ්චිත රටා සෙවීමට ඉඩ දෙයි:

% - Matches zero or more characters / අනුලකුණු ශුන්‍ය හෝ වැඩි ගණනකට ගැලපේ.
_ - Matches exactly one character / හරියටම එක අනුලකුණකට ගැලපේ.

Ex: `SELECT * FROM table_name WHERE column_name LIKE 'A%';`

This query retrieves all rows where the value in column_name starts with the letter 'A'.
මෙම විමසුම column_name අගය 'A' අකුරෙන් ආරම්භ වන සියලුම පේළි ලබා ගනී.

Therefore, the correct answer is 1) LIKE.
එබැවින්, නිවැරදි පිළිතුර 1) LIKE යන්නයි.

Other options like MATCH, FIND, SEARCH and SIMILAR are not standard SQL operators for pattern matching.
MATCH, FIND, SEARCH සහ SIMILAR වැනි වෙනත් විකල්ප, රටා ගැලපීම සඳහා සම්මත SQL මෙහෙයවනයන් නොවේ.

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

```
def func(x, y):
    return x + y

result = func("4", 2)
```

1) 6 2) "42" 3) "4"2 4) 42 5) TypeError

Answer: 5)

Explanation / පැහැදිලි කිරීම

- The function func(x, y) takes two parameters x and y, and returns their sum x + y.
  func(x, y) ශ්‍රිතය x සහ y පරාමිති දෙකක් ගන්නා අතර ඒවායේ එකතුව x + y ලබා දෙයි.
- In the function call func("4", 2), "4" is a string, and 2 is an integer.
  func("4", 2) ශ්‍රිතයේ "4" යනු string එකක් වන අතර 2 යනු පූර්ණ සංඛ්‍යාවකි.
- In Python, attempting to add a string and an integer (i.e., "4" + 2) will result in a TypeError, because Python does not automatically convert the integer to a string in this context.
  පයිතන්හි, string එකක් සහ පූර්ණ සංඛ්‍යාවක් (එනම්, "4" + 2) එකතු කිරීමට උත්සාහ කිරීම TypeError වලට හේතු වනු ඇත, මන්ද පයිතන්හි මෙම සන්දර්භය තුළ ස්වයංක්‍රීයව නිඛිල string එකකට පරිවර්තනය නොකරන බැවිනි.

Correct answer is **5)**.
නිවැරදි පිළිතුර **5)** වේ.

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

```
dict1={"a":10,"b":2,"c":3}
str1=""
for i in dict1:
   str1=str1+str(dict1[i])+" "
   str2=str1[ :- 1]
print(str2[ ::- 1])
```

1) `3 2` 2) `3 2 10` 3) `3 2 01` 4) `10 2 3` 5) `Error`

Answer : 3)

Explanation / පැහැදිලි කිරීම

This Python code initializes a dictionary `dict1` and iterates through its keys, appending the string representation of each value followed by a space to `str1`.
මෙම පයිතන් කේතය ශබ්දකෝෂයක් `dict1` ආරම්භ කර එහි යතුරු හරහා පුනරාවර්තනය කරයි, එක් එක් අගයේ තන්තු නිරූපණය `str1` ට ඉඩක් එක් කරයි.

After the loop, `str2` is assigned the value of `str1` with the last character (a trailing space) removed.
ලූපයෙන් පසුව, `str2` ට `str1` හි අගය පවරනු ලබන්නේ අවසාන අක්ෂරය (පසුගාමී ඉඩක්) ඉවත් කරමිනි.

The final print statement reverses `str2` using slicing ([::-1]) and outputs the reversed string.
අවසාන මුද්‍රණ ප්‍රකාශය ස්ලයිසිං ([::-1]) භාවිතයෙන් `str2` ආපසු හරවා ප්‍රතිලෝම තන්තුව ප්‍රතිදානය කරයි.

The values in the dictionary are processed in the order of keys (a, b, c), which maps to 10, 2 and 3 respectively.
ශබ්දකෝෂයේ ඇති අගයන් පිළිවෙලින් 10, 2 සහ 3 දක්වා සිතියම් ගත කරන යතුරු (a, b, c) අනුපිළිවෙලින් සකසනු ලැබේ.

Thus, `str1` becomes 10 2 3 , and `str2` becomes 10 2 3. Reversing `str2` gives `3 2 01`. Therefore, the output is `3 2 01`
මේ අනුව, `str1` 10 2 3 බවටත්, `str2` 10 2 3 බවටත් පත් වේ. `str2` ප්‍රතිවර්තනය කිරීමෙන් `3 2 01` ලැබේ. එබැවින් ප්‍රතිදානය `3 2 01` වේ.

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

9. What is the result of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිඵලය කුමක්ද?

```
result = 0
for i in range(1, 5):
    for j in range(i):
        if i % j == 0:
            result += 1
print(result)
```

1) `4`
2) `5`
3) `7`
4) `ZeroDivisionError: integer modulo by zero`
5) None of the above.

Answer : 4)

The code results in a `ZeroDivisionError` because of the line `if i % j == 0`. When `j` is `0`, dividing by zero is attempted, which is not allowed in Python. The outer loop iterates `i` from 1 to `4`, and the inner loop iterates `j` from `0` up to `i-1`. When `i` is `1`, the inner loop sets `j` to `0`, and the condition `i % j == 0`

tries to perform a modulo operation with 0, leading to the error. Thus, the program fails before it can complete and print any result.
if i % j == 0 රේඛාව නිසා කේතය ZeroDivisionError දෝශය ඇති කරයි. j 0 වන විට, ශුන්‍යයෙන් බෙදීමට උත්සාහ කරනු ලැබේ, එයට පයිතන් හි ඉඩ නොදේ. පිටත ලූපය 1 සිට 4 දක්වා i පුනරාවර්තනය කරයි, සහ අභ්‍යන්තර ලූපය 0 සිට i-1 දක්වා j පුනරාවර්තනය කරයි. i 1 වන විට, අභ්‍යන්තර ලූපය j 0 ලෙස සකසයි, සහ i % j == 0 කොන්දේසිය 0 සමඟ මොඩියුල මෙහෙයුමක් සිදු කිරීමට උත්සාහ කරයි, එය දෝෂයට තුඩු දෙයි. මේ අනුව, ඕනෑම ප්‍රතිඵලයක් සම්පූර්ණ කර මුද්‍රණය කිරීමට පෙර වැඩසටහන අසාර්ථක වේ.

10. Which of the following is true about the <font> tag in HTML?
HTML හි <font> උසුලනය සම්බන්ධයෙන් පහත සඳහන් දේවලින් සත්‍ය කුමක්ද?

1) It is a valid tag in HTML 5 for changing font style and color.
එය අකුරු විලාසය සහ වර්ණය වෙනස් කිරීම සඳහා HTML 5 හි වලංගු උසුලනයකි.
2) It is used in modern HTML for accessibility improvements.
එය ප්‍රවේශ්‍යතා වැඩිදියුණු කිරීම් සඳහා නවීන HTML හි භාවිතා වේ.
3) It is deprecated and should be replaced with CSS.
එය අත්හරින ලද අතර CSS සමඟ ප්‍රතිස්ථාපනය කළ යුතුය.
4) It is used to include custom fonts using the @font-face rule.
එය @font-face රීතිය භාවිතයෙන් අභිරුචි අකුරු ඇතුළත් කිරීමට භාවිතා කරයි.
5) It allows inline JavaScript for font manipulation.
එය අකුරු හැසිරවීම සඳහා JavaScript වලට ඉඩ සලසයි.

Answer: 3)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
False. The <font> tag is not part of HTML5. It was used in older HTML versions but is now deprecated.
වැරදියි. <font> උසුලනය HTML5 හි කොටසක් නොවේ. එය පැරණි HTML අනුවාද වල භාවිතා කරන ලද නමුත් දැන් අවලංගු කර ඇත.

Option / විකල්ප 2)
False. The <font> tag has no role in accessibility improvements, CSS and semantic HTML are used instead.
වැරදියි. ප්‍රවේශ්‍යතා වැඩිදියුණු කිරීම් සඳහා <font> උසුලනයට භූමිකාවක් නැත, ඒ වෙනුවට CSS සහ අර්ථකථන HTML භාවිතා වේ.

Option / විකල්ප 3)
True. The <font> tag has been deprecated in favor of CSS for styling text, including font style, size, and color.
නිවැරදියි. අකුරු විලාසය, ප්‍රමාණය සහ වර්ණය ඇතුළුව පෙළ හැඩගැන්වීම සඳහා CSS සඳහා <font> උසුලනය අවලංගු කර ඇත.

Option / විකල්ප 4)
False. Custom fonts are included using CSS with the @font-face rule, not the <font> tag.
වැරදියි. අභිරුචි අකුරු ඇතුළත් කර ඇත්තේ CSS භාවිතයෙන් @font-face රීතිය සමඟ මිස <font> උසුලනය නොවේ.

Option / විකල්ප 5)
False. The <font> tag does not support inline JavaScript, and such manipulation should be done using CSS or JavaScript applied to elements dynamically.
වැරදියි. උසුලනය පේළිගත JavaScript සඳහා සහය නොදක්වන අතර, එවැනි උපාමාරු සිදු කළ යුත්තේ CSS හෝ JavaScript භාවිතයෙන් මූලද්‍රව්‍යවලට ගතිකව යෙදිය යුතුය.

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