සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved]

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 203 MARKING

2025/07/01
ictfromabc
Ravindu Bandaranayake

1. Which programming development became more widespread during the third generation of computers?
   තුන්වන පරම්පරාවේ පරිගණක අතරතුර වඩාත් පුළුල් ලෙස ව්‍යාප්ත වූ ක්‍රමලේඛන සංවර්ධනය කුමක්ද?

   1) Machine code only/ යන්ත්‍ර කේතය පමණි
   2) Assembly language exclusively/ ඇසෙම්බ්ලි භාෂාව පමණක්
   3) <u>High-level languages like FORTRAN and COBOL</u>
      FORTRAN සහ COBOL වැනි ඉහළ මට්ටමේ භාෂා
   4) Neural network programming/ ස්නායුක ජාල ක්‍රමලේඛනය
   5) Visual programming environments/ දෘශ්‍ය ක්‍රමලේඛන පරිසරයන්

   Answer : 3)

   Explanation / පැහැදිලි කිරීම:

Option / විකල්පය 1
Incorrect. Machine code was used earlier but was not the main approach by the third generation. It is hard to maintain and less user-friendly.
වැරදියි. යන්ත්‍ර කේතය කලින් භාවිතා කරන ලද නමුත් තුන්වන පරම්පරාවේ ප්‍රධාන ප්‍රවේශය නොවීය. එය නඩත්තු කිරීමට අපහසු අතර අඩු පරිශීලක-හිතකාමී ය

Option / විකල්පය 2
Incorrect. Assembly language was common in the second generation but was gradually supplemented by more abstract high-level languages. It remained in use, but not exclusively.
වැරදියි. දෙවන පරම්පරාවේ ඇසෙම්බ්ලි භාෂාව සුලභ වූ නමුත් වඩාත් වියුක්ත ඉහළ මට්ටමේ භාෂා මගින් එය ක්‍රමයෙන් අතිරේකයක් කරන ලදී. එය භාවිතයේ පැවතුනි, නමුත් සම්පූර්ණයෙන්ම නොවේ.

Option / විකල්පය 3
Correct. The third generation saw the growing use of high-level languages like FORTRAN and COBOL, improving programming efficiency. These languages made software development more accessible and scalable.
නිවැරදියි. තුන්වන පරම්පරාව FORTRAN සහ COBOL වැනි ඉහළ මට්ටමේ භාෂා භාවිතයේ වැඩි වීම දර්ශනය විය, එමඟින් ක්‍රමලේඛන කාර්යක්ෂමතාව වැඩි දියුණු විය. මෙම භාෂා මෘදුකාංග සංවර්ධනය වඩාත් ප්‍රවේශ විය හැකි සහ පරිමාණය කළ හැකි විය.

Option / විකල්පය 4
Incorrect. Neural networks were theoretical at the time and not used in mainstream computing. They are associated with much later generations.
වැරදියි.එකල ස්නායුක ජාල න්‍යායාත්මක වූ අතර ප්‍රධාන ධාරාවේ පරිගණකකරණයේ භාවිතා නොවීය. ඒවා බොහෝ පසු පරම්පරාවන් සමඟ සම්බන්ධ වේ.

Option / විකල්පය 5
Incorrect. Visual programming environments emerged much later with personal computing. They did not exist during the third generation.
වැරදියි. පුද්ගලික පරිගණකකරණය සමඟ දෘශ්‍ය ක්‍රමලේඛන පරිසරයන් බොහෝ කලකට පසුව මතු විය. තුන්වන පරම්පරාව තුළ ඒවා නොතිබුණි.

2. Consider the following statements. Which statement(s) is/are correct?
පහත ප්‍රකාශ සලකා බලන්න. කුමන ප්‍රකාශය(ය) නිවැරදිද?

A. MS Word is an example of application software / MS Word යනු යෙදුම් මෘදුකාංග සඳහා උදාහරණයකි.
B. Antivirus is an example of system software / ප්‍රති-වයිරස යනු පද්ධති මෘදුකාංග සඳහා උදාහරණයකි.
C. Worms are a type of malware / Worms යනු අනිෂ්ට මෘදුකාංග වර්ගයකි.

1) A only. A පමණි.
2) B only. B පමණි.
3) C only. C පමණි.
4) <u>A and C only.</u> A හා C පමණි.
5) All of the above. ඉහත සියල්ලම.

Answer : 4)

Explanation / පැහැදිලි කිරීම:

Statement / ප්‍රකාශය 1)
MS Word is used to perform a specific user-oriented task (word processing), so it's clearly application software.
MS Word නිශ්චිත පරිශීලක-නැඹුරු කාර්යයක් (වචන සැකසීම) සිදු කිරීමට භාවිතා කරයි, එබැවින් එය පැහැදිලිවම යෙදුම් මෘදුකාංගයකි.

Statement / ප්‍රකාශය 2)
Although antivirus interacts with the system, it is utility software, not core system software like OS or drivers. Utility software is considered a separate category from system software.
ප්‍රති-වයිරස පද්ධතිය සමඟ අන්තර් ක්‍රියා කළත්, එය උපයෝගිතා මෘදුකාංගයකි, OS හෝ ධාවක වැනි මූලික පද්ධති මෘදුකාංග නොවේ. උපයෝගිතා මෘදුකාංග පද්ධති මෘදුකාංග වලින් වෙනම කාණ්ඩයක් ලෙස සැලකේ.

Statement / ප්‍රකාශය 3)
Worms are self-replicating malicious programs that spread through networks. They are indeed a type of malware.
Worms යනු ජාල හරහා පැතිරෙන ස්වයං-ප්‍රතිනිර්මාණය කරන අනිෂ්ට වැඩසටහන් වේ. ඒවා ඇත්ත වශයෙන්ම අනිෂ්ට මෘදුකාංග වර්ගයකි.

Therefore the answer is 4) / එබැවින් පිළිතුර 4) වේ

3. A text document contains 256 characters, including letters, digits, and special symbols. If each character is encoded using Unicode (assuming 16 bits per character), how many bits are needed to store the document?
පාඨ ලේඛනයක අකුරු, ඉලක්කම් සහ විශේෂ සංකේත ඇතුළුව අක්ෂර 256 ක් අඩංගු වේ. සෑම අක්ෂරයක්ම යුනිකෝඩ් භාවිතයෙන් කේතනය කර ඇත්නම් (අක්ෂරයකට බිටු 16 ක් උපකල්පනය කළහොත්), ලේඛනය ගබඩා කිරීමට කොපමණ බිටු ප්‍රමාණයක් අවශ්‍ය වේද?

1) 256×8
2) 256×32
3) $2^8 \times 8$
4) <u>$2^8 \times 16$</u>
5) $2^9 \times 16$

Answer : 4)

Explanation / පැහැදිලි කිරීම:

The document has 256 characters / ලේඛනයේ අක්ෂර 256ක් ඇත
Each character is encoded using Unicode / සෑම අක්ෂරයක්ම යුනිකෝඩ් භාවිතයෙන් කේතනය කර ඇත
Assume Unicode uses 16 bits per character.
යුනිකෝඩ් අක්ෂරයකට බිටු 16ක් භාවිතා කරන බව උපකල්පනය කරන්න

- Step-by-step Calculation / පියවරෙන් පියවර ගණනය කිරීම:
- Number of characters / අක්ෂර ගණන = 256
- Bits per character/ අක්ෂරයකට බිටු = 16
- So, total bits needed/ එබැවින්, අවශ්‍ය මුළු බිටු = $256 \times 16 = 4096$ bits

Now let's evaluate each option / දැන් අපි එක් එක් විකල්පය ඇගයීමට ලක් කරමු:

- Option / විකල්පය 1)

This would be correct only if each character used 8 bits (like in ASCII).
මෙය නිවැරදි වන්නේ (ASCII හි මෙන්) සෑම අක්ෂරයක්ම බිටු 8 ක් භාවිතා කළේ නම් පමණි.

Incorrect, because Unicode uses 16 bits, not 8.
වැරදියි, මන්ද යුනිකෝඩ් 8ක් නොව බිටු 16 ක් භාවිතා කරයි.

- Option / විකල්පය 2)

256 × 32 = 8192 bits
But we only need / නමුත් අපට අවශ්‍ය වන්නේ බිටු 256 × 16 = 4096 bits / පමණි, because / මන්ද:

Each character in Unicode typically uses 16 bits.
යුනිකෝඩ් හි සෑම අක්ෂරයක්ම සාමාන්‍යයෙන් බිටු 16ක් භාවිතා කරයි

32 bits would mean double the storage per character.
බිටු 32ක් යනු අක්ෂරයකට ගබඩා කිරීමේ හැකියාව දෙගුණ කරයි

So, choosing 256 × 32 means you're wrongly assuming that Unicode uses 32 bits per character.
එබැවින්, 256 × 32 තෝරා ගැනීමෙන් අදහස් වන්නේ යුනිකෝඩ් අක්ෂරයකට බිටු 32ක් භාවිතා කරන බව ඔබ වැරදි ලෙස උපකල්පනය කරන බවයි.

- Option / විකල්පය 3)

$2^8 = 256$, so this is 256 × 8 = 2048 bits / $2^8 = 256$ එබැවින් මෙය 256 × 8 = 2048 බිටු වේ
Incorrect, because again, it uses 8 bits per character (ASCII-style)
වැරදියි, මන්ද නැවතත්, එය අක්ෂරයකට බිටු 8ක් භාවිතා කරයි (ASCII-style)

- Option / විකල්පය 4)

$2^8 = 256$, and this uses 16 bits per character / $2^8 = 256$, සහ මෙය අක්ෂරයකට බිටු 16 භාවිතා කරයි
So this is 256 × 16 = 4096 bits / එබැවින් මෙය 256 × 16 = බිටු 4096

Correct and shows understanding of binary powers.
නිවැරදි සහ ද්විමය බල පිළිබඳ අවබෝධය පෙන්වයි

Best answer according to syllabus style
විෂය නිර්දේශට අනුව හොඳම පිළිතුර වේ

- Option / විකල්පය 5)

$2^9 = 512$, so this is 512 × 16 = 8192 bits / $2^9 = 512$, එබැවින් මෙය 512 × 16 = 8192 බිටු

Incorrect, because we only have 256 characters, not 512.
වැරදියි, මන්ද අපට ඇත්තේ අක්ෂර 256 ක් පමණි, 512 නොවේ

Therefore the answer is 4)
එබැවින් පිළිතුර 4 වේ

4. Which of the following statements about complementary number systems is true?
අනුපූරක සංඛ්‍යා පද්ධති පිළිබඳ පහත සඳහන් ප්‍රකාශ අතුරෙන් සත්‍ය වන්නේ කුමක්ද?

   1) 1's complement of a binary number is found by adding 1 to it.
   ද්විමය සංඛ්‍යාවක 1 හි අනුපූරකය එයට 1 එකතු කිරීමෙන් සොයාගත හැකිය.

   2) 2's complement is obtained by flipping all bits of a binary number.
   ද්විමය සංඛ්‍යාවක සියලුම බිටු පෙරලීමෙන් 2 හි අනුපූරකය ලබා ගනී.

   3) 1's complement and 2's complement of a number are always the same.
   සංඛ්‍යාවක 1 හි අනුපූරකය සහ 2 හි අනුපූරකය සැමවිටම සමාන වේ.

   4) 2's complement is obtained by taking the 1's complement and adding 1 to it.
   1 හි අනුපූරකය ගෙන එයට 1 එකතු කිරීමෙන් 2 හි අනුපූරකය ලබා ගනී.

   5) 1's complement is used more often than 2's complement for representing signed numbers.
   ලකුණු කළ සංඛ්‍යා නිරූපණය කිරීම සඳහා 1 හි අනුපූරකය 2 හි අනුපූරකයට වඩා බොහෝ විට භාවිතා වේ.

   Answer : 4)

   Explanation / **පැහැදිලි කිරීම**:

   The 1's complement and 2's complement are two methods for representing negative numbers in binary systems.
   ද්විමය පද්ධතිවල සෘණ සංඛ්‍යා නිරූපණය කිරීම සඳහා 1 හි අනුපූරකය සහ 2 හි අනුපූරකය ක්‍රම දෙකකි.

   1's Complement / 1 හි අනුපූරකය:
   Obtained by flipping all bits (changing 0s to 1s and 1s to 0s).
   සියලුම බිටු පෙරලීමෙන් ලබා ගන්නා ලදී.

   2's Complement / 2 හි අනුපූරකය:
   Obtained by taking the 1's complement and then adding 1.
   1 හි අනුපූරකය ගෙන 1 එකතු කිරීමෙන් ලබා ගනී.

   Therefore the correct answer is **option 4)**.
   එබැවින් නිවැරදි පිළිතුර විකල්පය 4) වේ.

5. Which of the following Karnaugh maps is correct for the given expression?
දී ඇති ප්‍රකාශනය සඳහා පහත සඳහන් කානෝ සිතියම් අතුරෙන් නිවැරදි වන්නේ කුමක්ද?

$$\overline{(\overline{A}.B.C) + (\overline{C}.B)} + (A.B)$$

A.

C \ AB	00	01	11	10
0	1	0	1	1
1	1	0	1	1

B.

B \ AC	01	00	10	11
0	0	1	1	1
1	1	0	1	1

C.

C \ AB	10	00	01	11
0	1	1	0	0
1	1	1	1	1

1) A and B only. A හා B පමණි.
2) A and C only. A හා C පමණි.
3) B and C only. B හා C පමණි.
4) All of the above. ඉහත සියල්ලම.
5) None of the above ඉහත කිසිවක් නොවේ

Answer : 1)

Explanation / **පැහැදිලි කිරීම**:

This is the Truth table for the given Boolean expression.
මෙය දී ඇති බූලියානු ප්‍රකාශනය සඳහා සත්‍ය වගුවයි.

ICN	A	B	C	F
1	0	0	0	1
2	0	0	1	1
3	0	1	0	0
4	0	1	1	0
5	1	0	0	1
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

In this truth table, ICN 3, 4 give 0 as the output. First, check whether all the 0s are in the correct places.
මෙම සත්‍යතා වගුවේ, ICN 3, 4 ප්‍රතිදානය ලෙස 0 ලබා දෙයි. පළමුව, සියලුම 0 නිවැරදි ස්ථානවල තිබේදැයි පරීක්ෂා කරන්න.

Then you will find that Karnaugh Map A and B are correct, but C is not. The difference between A and C is that when you shift the AB combination in Karnaugh Map A one position to the right, it becomes Karnaugh Map C. So the answers should also shift one position to the right. The correct Karnaugh Map for C is
එවිට ඔබට කානෝ සිතියම A සහ B නිවැරදි බව සොයා ගත හැකි අතර, C එසේ නොවේ. A සහ C අතර වෙනස නම්, ඔබ A කානෝ සිතියමහි AB සංයෝජනය දකුණට එක් ස්ථානයකට මාරු කළ විට, එය C කානෝ සිතියම බවට පත්වීමයි. එබැවින් පිළිතුරු ද එක් ස්ථානයක් දකුණට මාරු කළ යුතුය. C සඳහා නිවැරදි කානෝ සිතියම:

AB	10	00	01	11
0	1	1	0	1
1	1	1	0	1

6. What is the output of the full adder when, A = 0, B = 1 and Cin = 1?
A = 0, B = 1 **සහ** Cin = 1 **වූ විට පූර්ණාකලකයේ ප්‍රතිදානය කුමක්ද?**

1) Sum = 0, Cout = 0
2) Sum = 1, Cout = 0
3) <u>Sum = 0, Cout = 1</u>
4) Sum = 1, Cout = 1
5) None of the above / **ඉහත කිසිවක් නොවේ**

Answer : 3)

Explanation / **පැහැදිලි කිරීම**:

The full adder calculates the sum and carry out using following logic:
පූර්ණාකලකය එකතුව ගණනය කර පහත තර්කනය භාවිතා කර සිදු කරයි:

Sum: $A \oplus B \oplus Cin$ (XOR operation)
Cout: $(A \cdot B) + (B \cdot Cin) + (A \cdot Cin)$ (OR of the AND conditions)

Step by step calculation / **පියවරෙන් පියවර ගණනය කිරීම**:
1.A = 0, B = 1, Cin = 1

2.Sum Calculation:
Sum = $A \oplus B \oplus Cin = 0 \oplus 1 \oplus 1 = 0$
(since $1 \oplus 1 = 0$, and $0 \oplus 0 = 0$)

3.Carry out Calculation:
Cout = $(A \cdot B) + (B \cdot Cin) + (A \cdot Cin)$

Substituting the values / අගයන් ආදේශ කිරීම:

Cout = (0.1) + (1.1) + (0.1) = 0 + 1 + 0 = 1
Thus, the output is / මේ අනුව, ප්‍රතිදානය වන්නේ: Sum = 0, Cout = 1

Therefore the answer is 3)
එබැවින් පිළිතුර 3) වේ

7. Which of the following statements accurately describes the significance of "universal gates" in digital logic?
ඩිජිටල් තර්කනයේ "universal gates" වල වැදගත්කම නිවැරදිව විස්තර කරන පහත සඳහන් ප්‍රකාශ අතුරින් කවරක්ද?

   1) They are the only gates capable of performing arithmetic operations.
   අංක ගණිත මෙහෙයුම් සිදු කළ හැකි එකම ද්වාර ඒවා වේ.

   2) They can be used to implement any Boolean function without requiring other types of gates.
   වෙනත් ආකාරයේ ද්වාර අවශ්‍ය නොවී ඕනෑම බූලියන් ශ්‍රිතයක් ක්‍රියාත්මක කිරීමට ඒවා භාවිතා කළ හැකිය.

   3) They are designed to operate at any voltage level, making them universally compatible.
   ඒවා ඕනෑම වෝල්ටියතා මට්ටමක ක්‍රියාත්මක වන පරිදි නිර්මාණය කර ඇති අතර එමඟින් ඒවා විශ්වීය වශයෙන් අනුකූල වේ.

   4) They are the fastest logic gates available, leading to minimal propagation delay.
   ඒවා ලබා ගත හැකි වේගවත්ම තාර්කික ද්වාර වන අතර, එමඟින් අවම ප්‍රචාරණ ප්‍රමාදයකට මග පාදයි.

   5) They are used to create the most power-efficient logic circuits.
   ඒවා වඩාත් බල-කාර්යක්ෂම තාර්කික පරිපථ නිර්මාණය කිරීමට භාවිතා කරයි.

   Answer : 2)

   Explanation / පැහැදිලි කිරීම:

   The term "universal gate" refers to a logic gate that can be used to construct any other basic logic gate (AND, OR, NOT, XOR, etc.) or any Boolean function.
   "universal gate" යන පදය වෙනත් ඕනෑම මූලික තාර්කික ද්වාරයක් (AND, OR, NOT, XOR, ආදිය) හෝ ඕනෑම බූලියන් ශ්‍රිතයක් ගොඩනැගීමට භාවිතා කළ හැකි තාර්කික ද්වාරයකට යොමු වේ.

   This means that with just one type of universal gate, you can build any complex digital circuit.
   මෙයින් අදහස් කරන්නේ එක් ආකාරයක සංරචක තාර්කික ද්වාරයකින් ඔබට ඕනෑම සංකීර්ණ ඩිජිටල් පරිපථයක් ගොඩනගා ගත හැකි බවයි.

   For example, you can create an AND gate using only NAND gates, an OR gate using only NAND gates, and a NOT gate using only NAND gates. Therefore, you can build any Boolean expression using just NAND gates. The same is true for NOR gates.
   උදාහරණයක් ලෙස, ඔබට NAND ද්වාර පමණක් භාවිතා AND ද්වාරයක්, NAND ද්වාර පමණක් භාවිතා කර OR ද්වාරයක් සහ NAND ද්වාර පමණක් භාවිතා කර NOT ද්වාරයක් සෑදිය හැකිය. එබැවින්, ඔබට NAND ද්වාර පමණක් භාවිතා කර ඕනෑම බූලියන් ප්‍රකාශනයක් ගොඩනගා ගත හැකිය. NOR ද්වාර සඳහාද මෙයම සත්‍ය වේ.

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

8. What will be the output of the following code?
පහත කේතයේ ප්‍රතිදානය කුමක් වේද?

```
a = "Python"
print(a[1:4])
```

1) yth 2) Pyt 3) tho 4) ytho 5) ython

Answer : 1)

Explanation / **පැහැදිලි කිරීම**:

String a = "Python" has indices
String a = "Python" **දර්ශක ඇත**

P(0), y(1), t(2), h(3), o(4), n(5).

Slice a[1:4] includes index 1 to 3 (excluding 4)
Slice a[1:4] **හි දර්ශකය 1 සිට 3 දක්වා ඇතුළත් වේ (4 හැර)**

Characters at indices 1, 2, 3: 'y', 't', 'h'
දර්ශක 1, 2, 3 හි අක්ෂර: 'y', 't', 'h'

Output/**ප්‍රතිදානය**: 'yth'

9. What is the result of this code?
පහත කේතයේ ප්‍රතිදානය කුමක් වේද?

```
x = [1, 2, 3]
y = x.copy()
print(x is y)
```

1) True 2) False 3) None 4) Error 5) 0

Answer: 2)

Explanation / **පැහැදිලි කිරීම**:

Step-by-Step Explanation / **පියවරෙන් පියවර පැහැදිලි කිරීම**:

`x = [1, 2, 3]`
Creates a new list in memory / **මතකයේ නව ලැයිස්තුවක් නිර්මාණය කරයි.**
Variable `x` now refers to that list / **විචල්‍ය `x` දැන් එම ලැයිස්තුවට යොමු කරයි.**

`y = x.copy()`
This creates a shallow copy of the list `[1, 2, 3]`.
මෙය ලැයිස්තුවේ පිටපතක් නිර්මාණය කරයි `[1, 2, 3]`.

`y` now points to a new list object in memory.
`y` දැන් මතකයේ ඇති නව ලැයිස්තු වස්තුවකට යොමු කරයි.

This new list contains the same values but is stored separately from `x`.
මෙම නව ලැයිස්තුවේ එකම අගයන් අඩංගු නමුත් `x` වලින් වෙන වෙනම ගබඩා කර ඇත.

Now in memory / **දැන් මතකයේ**:
`x` ⟶ `[1, 2, 3]` # original list / **මුල් ලැයිස්තුව**
`y` ⟶ `[1, 2, 3]` # copy: different object with same contents
පිටපත: එකම අන්තර්ගතයන් සහිත විවිධ වස්තුව

`print(x is y)`
The is operator checks whether `x` and `y` refer to the exact same object (same memory address).
`x` ක්‍රියාකරු `y` සහ හරියටම එකම වස්තුවට (එකම මතක ලිපිනයට) යොමු කරන්නේද යන්න පරීක්ෂා කරයි.

They don't — `x.copy()` created a new object, even though it looks the same.
ඔවුන් එසේ නොකරයි - `x.copy` එය සමාන ලෙස පෙනුනද, නව වස්තුවක් නිර්මාණය කළේය.

So `x is y → False`.

Summary / සාරාංශය:
`copy()` duplicates the data, not the memory reference.
`copy()` මඟින් දත්ත අනුපිටපත් කරයි, මතක යොමුව නොවේ.

Therefore, `x is y → False`

Therefore the answer is 2)
එබැවින් පිළිතුර 2 වේ

10. Which function can be used to evaluate a string as a Python expression?
string එකක් පයිතන් ප්‍රකාශනයක් ලෙස ඇගයීමට භාවිතා කළ හැක්කේ කුමන ශ්‍රිතයද?

1) `str()` 2) `exec()` 3) `eval()` 4) `parse()` 5) `compile()`

Answer : 3)

The function that can be used to evaluate a string as a Python expression is `eval()`.
Python ප්‍රකාශනයක් ලෙස string ඇගයීමට භාවිතා කළ හැකි ශ්‍රිතය `eval()` වේ.

1. **`str()`**
This function is used to convert an object to its string representation. It does not evaluate a string as a Python expression.
වස්තුවක් එහි strings නිරූපණයට පරිවර්තනය කිරීමට මෙම ශ්‍රිතය භාවිතා කරයි. එය string එකක් Python ප්‍රකාශනයක් ලෙස ඇගයීමට ලක් නොකරයි.

2. **`exec()`**
This function is used to execute a string of Python code, which can include statements, functions, and classes. While it can run Python code, it does not return the result of an expression.
මෙම ශ්‍රිතය Python කේත මාලාවක් ක්‍රියාත්මක කිරීමට භාවිතා කරයි, එයට statements, functions, සහ classes ඇතුළත් විය හැක. එය පයිතන් Python ක්‍රියාත්මක කළ හැකි අතර, එය ප්‍රකාශනයක ප්‍රතිඵලය ලබා නොදේ.

3. **`eval()`**
This function evaluates a string as a Python expression and returns the result. It is specifically designed to evaluate expressions, making it the correct answer for evaluating a string as a Python expression.
මෙම string ශ්‍රිතය Python ප්‍රකාශනයක් ලෙස ඇගයීමට ලක් කර ප්‍රතිඵලය ලබා දෙයි. එය ප්‍රකාශන ඇගයීම සඳහා විශේෂයෙන් නිර්මාණය කර ඇති අතර, එය Python ප්‍රකාශනයක් ලෙස strings ඇගයීම සඳහා නිවැරදි පිළිතුර බවට පත් කරයි.

4. **`parse()`**
This function is not a standard Python function for evaluating strings as expressions. It might refer to parsing libraries or methods, but it does not evaluate a string as a Python expression directly.
මෙම ශ්‍රිතය strings ප්‍රකාශන ලෙස ඇගයීම සඳහා සම්මත Python ශ්‍රිතයක් නොවේ. එය විග්‍රහ කිරීමේ libraries හෝ methods වෙත යොමු විය හැකි නමුත් එය Python string ප්‍රකාශනයක් ලෙස සෘජුව ඇගයීමට ලක් නොකරයි.

5. **`compile()`**
This function compiles a string of Python code into a code object, which can then be executed using `exec()` or evaluated using `eval()`. While it is part of the process, `compile()` itself does not evaluate the string directly.
මෙම ශ්‍රිතය Python කේත මාලාවක් කේත වස්තුවක් බවට සම්පාදනය කරයි, එය පසුව `exec()` භාවිතයෙන් හෝ `eval()`භාවිතයෙන් ඇගයීමට ලක් කළ හැක. එය ක්‍රියාවලියේ කොටසක් වන අතර, `compile()` විසින්ම strings කෙලින්ම ඇගයීමට ලක් නොකරයි.

So, the correct answer is 3).
එබැවින්, නිවැරදි පිළිතුර 3).