සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved ]

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

**Information and Communication Technology**
**තොරතුරු හා සන්නිවේදන තාක්ෂණය**

# 2026 QUIZ NO 295 MARKING

2025/10/01
ictfromabc
Ravindu Bandaranayake

1. Which of the following statement(s) about cache memory in computers is/are correct?
පරිගණකවල ඇති නිහිත මතක ගැන ඇති පහත ප්‍රකාශ අතරින් නිවැරදි ප්‍රකාශ(ය) වන්නේ?

A. First-level cache memory is very fast, but it operates at a slower speed than the processor.
පළමු මට්ටමේ නිහිත මතකය ඉතා වේගවත් නමුත් එය සකසනයේ වේගයට වඩා අඩු වේගයකින් ක්‍රියාත්මක වේ.

B. Second-level and third-level cache memory can be located on the motherboard or in the processor.
දෙවන මට්ටමේ හා තෙවන මට්ටමේ නිහිත මතක මවු පුවරුව මත හෝ සකසනය තුළ පිහිටා තිබිය හැක.

C. Third-level cache memory, when located on the motherboard, depletes faster than main memory.
තෙවන මට්ටමේ නිහිත මතකය මවුපුවරුවේ පිහිටා ඇති විට ප්‍රධාන මතකයට වඩා වේගයෙන් අඩු වේ.

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

**Answer: 2)**

**Explanation / පැහැදිලි කිරීම:**

**A. First-level cache memory is very fast, but it operates at a slower speed than the processor**
**පළමු මට්ටමේ නිහිත මතකය ඉතා වේගවත් නමුත් එය සකසනයේ වේගයට වඩා අඩු වේගයකින් ක්‍රියාත්මක වේ.**
True, L1 (First-level) cache is the fastest type of cache memory, but it still operates slightly slower than the processor. It is designed to provide data to the CPU at very high speeds.
සත්‍ය වේ, L1 (පළමු මට්ටමේ) නිහිත යනු වේගවත්ම ආකාරයේ නිහිත මතකය, නමුත් එය තවමත් සකසනයට වඩා තරමක් මන්දගාමීව ක්‍රියා කරයි. එය ඉතා ඉහළ වේගයකින් CPU වෙත දත්ත සැපයීම සඳහා නිර්මාණය කර ඇත.

**B. Second-level and third-level cache memory can be located on motherboard or in the processor**
**දෙවන මට්ටමේ හා තෙවන මට්ටමේ නිහිත මතක මවු පුවරුවේ තුල හෝ සකසනය තුල පිහිටා තිබිය හැක.**
True, L2 and L3 cache can be located either on the processor chip (on-die) or on the motherboard in older systems. In modern systems, L2 and L3 caches are typically integrated into the processor.
සත්‍ය වේ, L2 සහ L3 නිහිත සකසන චිපයේ on-die හෝ පැරණි පද්ධතිවල මවු පුවරුවේ ස්ථානගත කළ හැක. නවීන පද්ධතිවල, L2 සහ L3 නිහිත සාමාන්‍යයෙන් සකසනයට අනුකලනය වේ.

**C. Third-level cache memory, when located on the motherboard, depletes faster than main memory**
**තෙවන මට්ටමේ නිහිත මතකය මවුපුවරුවේ පිහිටා ඇති විට ප්‍රධාන මතකයට වඩා වේගයෙන් අඩු වේ.**
False, Cache memory does not "deplete" in the same way memory is used up. L3 cache is faster than main memory (RAM) and is not depleted faster; it simply holds data temporarily for quicker access by the CPU.
අසත්‍ය වේ, මතකය භාවිතා කරන ආකාරයටම නිහිත මතකය ක්ෂය නොවේ. L3 නිහිතය ප්‍රධාන මතකයට RAM වඩා වේගවත් වන අතර වේගයෙන් ක්ෂය නොවේ. එය හුදෙක් CPU මගින් ඉක්මන් ප්‍රවේශය සඳහා දත්ත තාවකාලිකව රඳවා තබා ගනී.

Only statements A and B are correct, so the answer is 2) A and B only.
A සහ B ප්‍රකාශ පමණක් නිවැරදි බැවින් පිළිතුර 2) A හා B පමණි.

2. An advantage of using free and open source software is,
නිදහස් සහ විවෘත මූලාශ්‍ර මෘදුකාංග භාවිතයේ වාසියක් වන්නේ,

1) Virus attacks are minimal for this type of softwares.
මෙම වර්ගයේ මෘදුකාංග සඳහා වෛරස් ප්‍රහාරයන් අවම වේ.
2) More and more people around the world are using these softwares.
ලොව පුරා වැඩි පිරිසක් මෙම මෘදුකාංග භාවිතා කිරීම.
3) These softwares can be sold at a high price.
මෙම මෘදුකාංග අධික මිලකට විකිණිය හැකිවීම.
4) Existence of standards on these softwares.
මෙම මෘදුකාංග පිළිබඳව සම්මතයන් පැවතීම.
5) These softwares are well designed for every task.
සෑම කාර්යයක් සඳහාම මෙම මෘදුකාංග ප්‍රමාණවත් පරිදි නිර්මාණය වී තිබීම.

**Answer: 1)**

Let's consider each of the answers here.
මෙහි ඇති එක් එක් පිළිතුරු පිළිබඳව සලකා බලමු.

**Option / විකල්පය 1)**
Open-source software has its source code available to the public, allowing security vulnerabilities to be identified and fixed quickly by the community. As a result, the chances of virus attacks can be lower compared to proprietary software, where security issues may take longer to address.
විවෘත-මූලාශ්‍ර මෘදුකාංගය මහජනතාවට එහි ප්‍රභව කේතය ඇති අතර, ප්‍රජාව විසින් ආරක්ෂක දුර්වලතා හඳුනා ගැනීමට සහ ඉක්මනින් නිවැරදි කිරීමට ඉඩ සලසයි. එහි ප්‍රතිඵලයක් වශයෙන්, ආරක්ෂක ගැටලු විසඳීමට වැඩි කාලයක් ගතවන හිමිකාර මෘදුකාංග හා සසඳන විට වෛරස් ප්‍රහාරවල අවස්ථා අඩු විය හැක.

**Option / විකල්පය 2)**
While it's true that open-source software is widely used globally, this statement is not an advantage by itself. The number of users doesn't necessarily make software inherently better or more secure.
විවෘත මූලාශ්‍ර මෘදුකාංග ගෝලීය වශයෙන් බහුලව භාවිතා වන බව සත්‍යයක් වුවද, මෙම ප්‍රකාශය තනිකරම වාසියක් නොවේ. පරිශීලකයින් සංඛ්‍යාව අවශ්‍යයෙන්ම මෘදුකාංග නෛසර්ගිකව වඩා හොඳ හෝ ආරක්ෂිත බවට පත් නොකරයි.

**Option / විකල්පය 3)**
This is not true for free and open-source software. FOSS is generally available for free or at a low cost. Although service and support around FOSS can be monetized, the software itself is not typically sold at a high price.
මෙය නිදහස් සහ විවෘත මෘදුකාංග සඳහා සත්‍ය නොවේ. FOSS සාමාන්‍යයෙන් නොමිලේ හෝ අඩු මුදලකට ලබා ගත හැක. FOSS වටා ඇති සේවාව සහ සහාය මුදල් ඉපැයීමට හැකි වුවද, මෘදුකාංගය සාමාන්‍යයෙන් ඉහළ මිලකට අලෙවි නොවේ.

**Option / විකල්පය 4)**
Many open-source projects follow specific standards, but this is not exclusive to FOSS. Proprietary software can also follow standards. Thus, while important, it's not a unique advantage of FOSS.
බොහෝ විවෘත මූලාශ්‍ර ව්‍යාපෘති නිශ්චිත ප්‍රමිතීන් අනුගමනය කරයි, නමුත් මෙය FOSS සඳහා පමණක් සීමා නොවේ. හිමිකාර මෘදුකාංග ද ප්‍රමිතීන් අනුගමනය කළ හැකිය. මේ අනුව, වැදගත් වුවද, එය FOSS හි අනන්‍ය වාසියක් නොවේ.

**Option / විකල්පය 5)**
This is a generalization and not always true. While some open-source software is highly capable, the quality and design can vary greatly depending on the project. Not all open-source software is designed to handle "every task."
මෙය සාමාන්‍යකරණයක් වන අතර සෑම විටම සත්‍ය නොවේ. සමහර විවෘත-මූලාශ්‍ර මෘදුකාංග ඉහළ හැකියාවක් ඇති අතර, ව්‍යාපෘතිය අනුව ගුණාත්මකභාවය සහ සැලසුම බොහෝ සෙයින් වෙනස් විය හැක. සියලුම විවෘත මූලාශ්‍ර මෘදුකාංග "සෑම කාර්යයක්ම" හැසිරවීමට නිර්මාණය කර නැත.

So, the correct answer number is 1.
එමනිසා, නිවැරදි පිළිතුරු අංකය 1 වේ.

3. Which of the following represent the bitwise XNOR operation of the two binary numbers $01010101_2$ and $10101010_2$?
පහත කවරක් $01010101_2$ සහ $10101010_2$ යන ද්විමය සංඛ්‍යා දෙකෙහි බිටු ලෙස බිටු අනුසාරිත XNOR මෙහෙයුම නිරූපණය කරයිද?

<u>1) $00000000_2$</u> 2) $11110000_2$ 3) $11001100_2$ 4) $11111111_2$ 5) $10010011_2$

**Answer: 1)**

**Logic of XNOR gate / XNOR ද්වාරයේ තර්කය**

| A | B | A XNOR B |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $00000000_2$ | | | | | | | |

`

Therefore, the correct answer is 1). / එබැවින්, නිවැරදි පිළිතුර 1) වේ.

4. Which of the following equation(s) is/are equivalent to the Boolean expression $A(\overline{DC} + D\overline{C}) + 1D$?
පහත කුමන ප්‍රකාශ(ය) $A(\overline{DC} + D\overline{C}) + 1D$ යන බූලියන් ප්‍රකාශනයට සමාන වේද?

A. $A\overline{D} + A\overline{C} + D$
B. A+D
C. 1

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

**Answer: 4)**

**Explanation / පැහැදිලි කිරීම:**

| | |
|---|---|
| $A(\overline{DC} + D\overline{C}) + 1D$ | |
| $= A(\overline{D} + \overline{C} + D\overline{C}) + 1D$ | De Morgan's law ද මුවාර් ප්‍රමේයය |
| $= A(\overline{D} + \overline{C}) + 1D$ | Redundancy law සමතිරික්ත න්‍යාය |
| $= A\overline{D} + A\overline{C} + 1D$ | Distributive law විඝටන න්‍යාය |
| $= A\overline{D} + A\overline{C} + D$ | Identity law සර්වසාම්‍ය න්‍යාය |
| $= A\overline{D} + D + A\overline{C}$ | Commutative law න්‍යාදේශ න්‍යාය |
| $= A + D + A\overline{C}$ | Redundancy law සමතිරික්ත න්‍යාය |
| $= A + A\overline{C} + D$ | Commutative law න්‍යාදේශ න්‍යාය |
| $= A + D$ | Redundancy law සමතිරික්ත න්‍යාය |

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

5. Which logic gate is equivalent to the operation "if either one but not both"?
"එකක් මිස දෙකම නොවේ" යන මෙහෙයුමට සමාන තාර්කික ද්වාරය කුමක්ද?

<u>1) XOR</u> 2) OR 3) XNOR 4) NAND 5) NOR

**Answer: 1)**

**Explanation / පැහැදිලි කිරීම:**

"If either one but not both" means if both inputs are one then the output become zero while if one input become one and other become zero then the output will be one.
"එකක් මිස දෙකම නොවේ නම්" යනු ආදාන දෙකම එකක් නම් ප්‍රතිදානය ශුන්‍ය වන අතර එක් ආදානයක් එකක් වී අනෙක ශුන්‍ය වුවහොත් ප්‍රතිදානය එකක් වනු ඇත යන්නයි.

The relevant logic gate to that definition is XOR gate.
එම නිර්වචනයට අදාළ තාර්කික ද්වාරය XOR ද්වාරය වේ.

Truth table relevant to XOR gate (Consider A and B are two inputs while F is an output):
XOR ද්වාරයට අදාළ සත්‍යතා වගුව (A සහ B යනු ආදාන දෙකක් වන අතර F ප්‍රතිදානයක් ලෙස සලකන්න):

| ICN | A | B | F |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 1 |
| 3 | 1 | 0 | 1 |
| 4 | 1 | 1 | 0 |

Therefore, the correct answer is XOR. / එබැවින් නිවැරදි පිළිතුර XOR වේ.

6. Which of the following statements about file handling modes in Python is correct?
පයිතන්හි ගොනු හැසිරවීමේ ආකාර පිළිබඳ පහත ප්‍රකාශ අතුරින් නිවැරදි වන්නේ කුමක්ද?

1) The mode `'r'` is used to open a file for both reading and writing.
කියවීම සහ ලිවීම යන දෙකටම ගොනුවක් විවෘත කිරීමට `'r'` ආකාරය භාවිතා කරයි.
2) The mode `'a'` is used to read from a file, and it raises an error if the file does not exist.
ගොනුවක් කියවීමට `'a'` ආකාරය භාවිතා කරන අතර, ගොනුව නොමැති නම් එය දෝෂයක් මතු කරයි.
3) <u>The mode `'w'` will create a new file if it doesn't exist, and if the file exists, it will overwrite its content.</u>
`'w'` ආකාරය ගොනුව නොමැති නම් නව ගොනුවක් සාදනු ඇත, සහ ගොනුව පවතී නම්, එය එහි අන්තර්ගතය උඩින් නැවත ලියයි.
4) The mode `'r+'` is used to open a file for writing only, and it creates a new file if it doesn't exist.
ලිවීම සඳහා පමණක් ගොනුවක් විවෘත කිරීමට `'r+'` ආකාරය භාවිතා කරන අතර එය නොමැති නම් එය නව ගොනුවක් සාදයි.
5) The mode `'x'` will append data to the end of the file if it already exists.
`'x'` ආකාරය ගොනුව දැනටමත් පවතී නම් ගොනුවේ අවසානයට දත්ත එකතු කරයි.

**Answer: 3)**

Let's consider each file handling modes with each of their function.
එක් එක් ගොනු හැසිරවීමේ ප්‍රකාරයන් ඒවායේ එක් එක් කාර්යය සමග සලකා බලමු.

- `'r'`
  Opens a file for reading. Raises an error if the file doesn't exist.
  කියවීම සඳහා ගොනුවක් විවෘත කරයි. ගොනුව නොමැති නම් දෝෂයක් මතු කරයි.

- `'w'`
  Opens a file for writing. Creates a new file if it doesn't exist or overwrites if it does.
  ලිවීම සඳහා ගොනුවක් විවෘත කරයි. එය නොපවතියි නම් නව ගොනුවක් සාදන්න හෝ එය තිබේ නම් උඩින් (overwrite) ලියයි.

- 'a'
  Opens a file for appending. Creates a new file if it doesn't exist.
  එකතු කිරීම සඳහා ගොනුවක් විවෘත කරයි. එය නොමැති නම් නව ගොනුවක් සාදයි.

- `'r+'`
  Opens a file for both reading and writing. Raises an error if the file doesn't exist.
  කියවීම සහ ලිවීම යන දෙකටම ගොනුවක් විවෘත කරයි. ගොනුව නොමැති නම් දෝෂයක් මතු කරයි.

- `'x'`
  Creates a new file. Raises an error if the file already exists.
  නව ගොනුවක් සාදයි. ගොනුව දැනටමත් තිබේ නම් දෝෂයක් මතු කරයි.

Explanation of each option:
එක් එක් විකල්පය පැහැදිලි කිරීම:

**Option / විකල්ප 1)**
Incorrect statement. The mode **'r'** is used only for reading, not both reading and writing.
වැරදි ප්‍රකාශයකි. **'r'** ප්‍රකාරය කියවීම සඳහා පමණක් භාවිතා වේ, කියවීම සහ ලිවීම යන දෙකම නොවේ.

**Option / විකල්ප 2)**
Incorrect statement. The mode **'a'** is for appending to a file, not reading. It doesn't raise an error if the file doesn't exist but creates a new file.
වැරදි ප්‍රකාශයකි. **'a'** ප්‍රකාරය ගොනුවකට එකතු කිරීම සඳහා මිස කියවීම නොවේ. ගොනුව නොමැති නමුත් නව ගොනුවක් නිර්මාණය කරන්නේ නම් එය දෝෂයක් මතු නොකරයි.

**Option / විකල්ප 3)**
Correct statement. The mode **'w'** will create a new file if it doesn't exist. If the file exists, it will overwrite its content.
නිවැරදි ප්‍රකාශයකි. **'w'** ප්‍රකාරය එය නොමැති නම් නව ගොනුවක් සාදනු ඇත. ගොනුව පවතී නම්, එය එහි අන්තර්ගතය උඩින් ලියයි.

**Option / විකල්ප 4)**
Incorrect statement. The mode **'r+'** is for reading and writing, not writing only. It raises an error if the file doesn't exist.
වැරදි ප්‍රකාශයකි. **'r+'** ප්‍රකාරය ලිවීමට පමණක් නොව කියවීමට සහ ලිවීම සඳහා වේ. ගොනුව නොමැති නම් එය දෝෂයක් මතු කරයි.

**Option / විකල්ප 5)**
Incorrect statement. The mode **'x'** creates a new file and raises an error if the file exists, but it doesn't append data to an existing file.
වැරදි ප්‍රකාශයකි. **'x'** ප්‍රකාරය නව ගොනුවක් සාදන අතර ගොනුව තිබේ නම් දෝෂයක් ඇති කරයි, නමුත් එය පවතින ගොනුවකට දත්ත එකතු නොකරයි.

The correct answer is 3) The mode 'w' will create a new file if it does not exist, and if the file exists, it will overwrite its content.
නිවැරදි පිළිතුර 3) `'w'` ප්‍රකාරය ගොනුව නොමැති නම් නව ගොනුවක් සාදනු ඇත, සහ ගොනුව පවතී නම්, එය එහි අන්තර්ගතය උඩින් නැවත ලියයි.

7. What is the output given by the Python statement `A == 'A'`?
   `A == 'A'` යන පයිතන් ප්‍රකාශය මගින් ලබා දෙනු ලබන ප්‍රතිදානය කුමක්ද?

   1) `True`
   2) `0`
   3) `1`
   4) <u>`NameError: name 'A' is not defined`</u>
   5) `False`

   **Answer: 4)**

   **Explanation / පැහැදිලි කිරීම:**

   The Python statement A = = 'A' would indeed result in a NameError if the variable A is not defined before the comparison.
   පයිතන් ප්‍රකාශය A = = 'A' සැසඳීමට පෙර A විචල්‍යය අර්ථ දක්වා නොමැති නම් ඇත්ත වශයෙන්ම NameError ඇති කරයි.

**A is not defined:**
The statement assumes that A is a variable that has been assigned a value. However, if A has not been declared or assigned a value, it will be considered undefined.
ප්‍රකාශය උපකල්පනය කරන්නේ A යනු අගයක් පවරන ලද විචල්‍යයක් බවයි. කෙසේ වෙතත්, A ප්‍රකාශ කර නොමැති නම් හෝ අගයක් පවරා නොමැති නම්, එය නිර්වචනය නොකළ ලෙස සලකනු ලැබේ.

**Comparison operation:**
The = = operator is used to compare the values of two operands. In this case, it's trying to compare A with the string "A".
= = ක්‍රියාකරු ඔපෙරාන්ඩ් දෙකක අගයන් සංසන්දනය කිරීමට භාවිතා කරයි. මෙම අවස්ථාවේදී, එය A තන්තුව සමඟ A සංසන්දනය කිරීමට උත්සාහ කරයි.

**NameError exception:**
Since A is undefined, Python raises a NameError exception indicating that the name A is not recognized.
A නිර්වචනය නොකළ බැවින්, පයිතන් A නම හඳුනාගෙන නැති බව අඟවමින් NameError ව්‍යතිරේකයක් මතු කරයි.

Therefore, the correct output of the statement would be,
එබැවින්, ප්‍රකාශයේ නිවැරදි ප්‍රතිදානය වනුයේ,

4) `NameError: name 'A' is not defined`

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

```
x = 10
y = x << 2
print(y)
```

1) `20`    2) `30`    3) `40`    4) `50`    5) `45`

**Answer: 3)**

**Explanation / පැහැදිලි කිරීම:**

The above Python code uses the << operator to perform a left shift, moving the bits of a number to the left by a specified number of positions and then displays the resulting decimal value.
ඉහත පයිතන් කේතය left shift සිදු කිරීමට << මෙහෙයවනය භාවිතා කරයි, එය නිශ්චිත ස්ථාන ගණනකින් සංඛ්‍යාවක බිටු වමට ගෙන ගොස් ප්‍රතිදානය ලෙස එහි දශමය අගය පෙන්වයි.

Let's substitute the value of x / x හි අගය ආදේශ කරමු:
`y = 10 << 2`

| Conversion of 10 into binary<br>10 ද්වීමය බවට පරිවර්තනය කිරීම | | | | | | | |
|---|---|---|---|---|---|---|---|
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| $1010_2$ | | | | | | | |

| `10 << 2`<br>Performing left shift (shifts the bits of number to left by 2)<br>left shift සිදු කිරීම (සංඛ්‍යාව බිටු දෙකක් වමට මාරු කරයි) | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| | | | | | | | <<2 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| $101000_2$ | | | | | | | |

| Conversion of $101000_2$ into decimal<br>$101000_2$ දශමය බවට පරිවර්තනය කිරීම | | | | | | | |
|---|---|---|---|---|---|---|---|
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 32 | 0 | 8 | 0 | 0 | 0 |
| 32+8 | | | | | | | |
| 40 | | | | | | | |

The value of the variable y becomes 40.
y විචල්‍යයේ අගය 40 වෙයි.

Therefore, the output of the above Python code is 3) 40.
එබැවින්, ඉහත Python කේතයේ ප්‍රතිදානය වන්නේ 3) 40.

9. Which of the following statements is true about Python lists?
පයිතන් ලැයිස්තු සම්බන්ධයෙන් පහත සඳහන් කුමන ප්‍රකාශය සත්‍ය ද?

1) Lists in Python are immutable, meaning they cannot be changed after they are created.
පයිතන් හි ලැයිස්තු වෙනස් කළ නොහැකි ය, එනම් ඒවා නිර්මාණය කිරීමෙන් පසුව වෙනස් කළ නොහැක.

2) You can add elements to a list using the `append()` method.
ඔබට `append()` ශ්‍රිතය භාවිතයෙන් ලැයිස්තුවකට මූලාංග එක් කළ හැක.

3) Lists in Python can only contain elements of the same type.
පයිතන් හි ලැයිස්තු වල අඩංගු විය හැක්කේ එකම වර්ගයේ මූලාංග පමණි.

4) Lists in Python have a fixed size and cannot grow or shrink.
පයිතන් හි ලැයිස්තු වලට ස්ථාවර විශාලත්වයක් ඇති අතර වර්ධනය වීමට හෝ හැකිලීමට නොහැක.

5) None of the above.
ඉහත කිසිවක් නොවේ.

**Answer: 2)**

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

**Option / විකල්ප 1)**
This statement is false. In Python, lists are mutable, which means that you can modify them after they are created, such as by adding, removing, or changing elements.
මෙම ප්‍රකාශය අසත්‍ය වේ. Python හි, ලැයිස්තු විකෘති වේ, එයින් අදහස් කරන්නේ ඔබට ඒවා නිර්මාණය කිරීමෙන් පසුව, මූලාංග එකතු කිරීම, ඉවත් කිරීම හෝ වෙනස් කිරීම වැනි ඒවා වෙනස් කළ හැකි බවයි.

**Option / විකල්ප 2)**
This statement is true. The `append()` method allows you to add elements to the end of a list. For example, if you have a list `my_list = [1, 2]` and you call `my_list.append(3)`, the list becomes `[1, 2, 3]`.
මෙම ප්‍රකාශය සත්‍ය වේ. `append()` ක්‍රමය මඟින් ඔබට ලැයිස්තුවක අවසානයට මූලද්‍රව්‍ය එක් කිරීමට ඉඩ සලසයි. උදාහරණයක් ලෙස, ඔබට `my_list = [1, 2]` ලැයිස්තුවක් තිබේ නම් සහ ඔබ `my_list.append(3)` ලෙස අමතන්නේ නම්, ලැයිස්තුව `[1, 2, 3]` බවට පත් වේ.

**Option / විකල්ප 3)**
This statement is false. Python lists can contain elements of different types. For example, a list can contain integers, strings, and other data types all together like `[1, "apple", 3.14]`.
මෙම ප්‍රකාශය අසත්‍ය වේ. පයිතන් ලැයිස්තු වල විවිධ වර්ගවල මූලද්‍රව්‍ය අඩංගු විය හැක. උදාහරණයක් ලෙස, ලැයිස්තුවක පූර්ණ සංඛ්‍යා, string සහ `[1, "apple", 3.14]` වැනි අනෙකුත් දත්ත වර්ග අඩංගු විය හැක.

**Option / විකල්ප 4)**
This statement is false. Python lists do not have a fixed size. You can add elements using methods like `append()` or `extend()` and remove elements using methods like `remove()` or `pop()`, allowing lists to grow or shrink dynamically.
මෙම ප්‍රකාශය අසත්‍ය වේ. පයිතන් ලැයිස්තු වලට ස්ථාවර ප්‍රමාණයක් නොමැත. ඔබට `append()` හෝ `extend()` වැනි ක්‍රම භාවිතයෙන් මූලද්‍රව්‍ය එක් කළ හැකි අතර, ලැයිස්තු ගතිකව වර්ධනය වීමට හෝ හැකිලීමට ඉඩ දෙමින්, `remove()` හෝ `pop()` වැනි ක්‍රම භාවිතයෙන් මූලාංග ඉවත් කරන්න.

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

10. What is the output of the following python program?
පහත පයිතන් වැඩසටහනේ ප්‍රතිදානය කුමක්ද?

```
n = 4
fact = 1
i = 2
while i <= n:
      fact = fact*i
      i = i+1
print (fact)
```

1) 6 2) 120 3) 24 4) 2 5) 1

**Answer: 3)**

**Explanation / පැහැදිලි කිරීම:**

This Python program calculates the factorial of the number n, which is set to 4. It initializes a variable fact to 1, which will hold the product of the numbers.
මෙම පයිතන් ක්‍රමලේඛය 4 ලෙස සකසා ඇති n සංඛ්‍යාවේ සාධක අගය ගණනය කරයි. එය සංඛ්‍යාවල ගුණිතය රඳවා තබන විචල්‍ය සත්‍යයක් 1 ට ආරම්භ කරයි.

The variable i is initialized to 2 and serves as the loop counter. The while loop runs as long as i is less than or equal to n.
i විචල්‍යය 2 ට ආරම්භ කර ලූප ගණකය ලෙස ක්‍රියා කරයි. i, n ට වඩා අඩු හෝ සමාන වන තාක් ලූපය ධාවනය වේ.

Inside the loop, fact is multiplied by i, and then i is incremented by 1. The loop iterates for i values of 2, 3, and 4.
ලූපය තුළ, fact i මගින් ගුණ කරනු ලැබේ, පසුව i, 1 න් වැඩි වේ. ලූපය 2, 3 සහ 4 යන i අගයන් සඳහා පුනරාවර්තනය වේ.

On each iteration, fact is updated as follows when i is 2, fact becomes (1 × 2 = 2), when i is 3, fact updates to (2 × 3 = 6),and finally, when i is 4, fact becomes (6 × 4 = 24).
එක් එක් පුනරාවර්තනයකදී, i, 2 වන විට සත්‍යය පහත පරිදි යාවත්කාලීන වේ, fact (1 × 2 = 2), i, 3 වන විට fact යාවත්කාලීන වේ (2 × 3 = 6), සහ අවසාන වශයෙන්, i, 4 වන විට, fact (6 × 4 = 24) බවට පත් වේ.

Once i exceeds 4, the loop terminates, and the final value of fact, which is 24, is printed. Thus, the output of the program is 24.
i, 4 ඉක්මවූ පසු, ලූපය අවසන් වන අතර, සත්‍යයේ අවසාන අගය, එනම් 24, මුද්‍රණය වේ. මේ අනුව, වැඩසටහනේ ප්‍රතිදානය 24 කි.

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