සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved]

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

Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 371 MARKING

2025/12/16
ictfromabc
Ravindu Bandaranayake

1. The BIOS programme activates as soon as the computer is switched on. The BIOS programme is located in, පරිගණකය සක්‍රිය වූ වහාම BIOS වැඩසටහන ක්‍රියාත්මක වේ. BIOS වැඩසටහන පිහිටා ඇත්තේ,

   1) Cache Memory. වාරක මතකය.
   2) <u>ROM. පඨන මාත්‍ර මතකය.</u>
   3) CMOS RAM. CMOS මතකය.
   4) Northbridge Chipset. නෝර්ත්බ්‍රිජ් චිප්සෙට්.
   5) Southbridge Chipset. සවුත්බ්‍රිජ් චිප්සෙට්.

   Answer : 2)

   1. Cache Memory
   Cache memory is a small, high-speed volatile memory located close to the CPU, designed to speed up access to frequently used data and instructions. It is not used for storing the BIOS. Cache memory loses its contents when the computer is powered off, making it unsuitable for storing firmware like the BIOS, which needs to be persistent.
   වාරක මතකය යනු නිතර භාවිතා කරන දත්ත සහ උපදෙස් වෙත ප්‍රවේශය වේගවත් කිරීම සඳහා නිර්මාණය කර ඇති CPU වෙත සමීපව පිහිටා ඇති කුඩා, අධිවේගී නශ්‍ය මතකයකි. එය BIOS ගබඩා කිරීම සඳහා භාවිතා නොවේ. පරිගණකය ක්‍රියා විරහිත වූ විට වාරක මතකය එහි අන්තර්ගතය නැති කර ගනී, එය BIOS වැනි ස්ථිරාංග ගබඩා කිරීමට නුසුදුසු වන අතර එය නොනැසී පැවතිය යුතුය.

   2. ROM
   The BIOS (Basic Input/Output System) is a firmware program that is essential for initializing and testing the hardware components of a computer during the booting process. When a computer is powered on, the BIOS is the first program to run, and it is responsible for performing the POST (Power-On Self-Test), initializing hardware components, and loading the operating system from the boot device.
   BIOS (මූලික ආදාන/ප්‍රතිදාන පද්ධතිය) යනු පරිගණකයේ ආරම්භ කිරීමේ ක්‍රියාවලියේදී පරිගණකයේ දෘඪාංග සංරචක ආරම්භ කිරීම සහ පරීක්ෂා කිරීම සඳහා අත්‍යවශ්‍ය වන ස්ථිරාංග වැඩසටහනකි. පරිගණකයක් ක්‍රියාත්මක වන විට, BIOS යනු ක්‍රියාත්මක වන පළමු වැඩසටහන වන අතර, එය (Power-On Self-Test) දෘඪාංග සංරචක ආරම්භ කිරීම සහ ඇරඹුම් උපාංගයෙන් මෙහෙයුම් පද්ධතිය පූරණය සඳහා වගකීම දරයි.

   The BIOS is stored in Read-Only Memory (ROM). ROM is a type of non-volatile memory, meaning that its contents are retained even when the computer is powered off. This makes ROM an ideal location for storing firmware like the BIOS, as it ensures that the BIOS program is always available to execute when the computer starts up. Traditionally, ROM chips were used to store the BIOS, but in modern computers, EEPROM (Electrically Erasable Programmable Read-Only Memory) or flash memory, which are types of ROM, are commonly used. Regardless of the specific type, the BIOS is fundamentally stored in ROM.
   BIOS ගබඩා කර ඇත්තේ කියවීමට පමණක් වන මතකය ROM තුළය. ROM යනු නශ්‍ය නොවන මතක වර්ගයකි, එනම් පරිගණකය ක්‍රියා විරහිත වූ විට පවා එහි අන්තර්ගතය රඳවා තබා ගනී. මෙය BIOS වැනි ස්ථිරාංග ගබඩා කිරීම සඳහා ROM සුදුසු ස්ථානයක් බවට පත් කරයි, එය පරිගණකය ආරම්භ වන විට BIOS වැඩසටහන ක්‍රියාත්මක කිරීමට සැමවිටම පවතින බව සහතික කරයි. සම්ප්‍රදායිකව, BIOS ගබඩා කිරීම සඳහා ROM චිප් භාවිතා කරන ලද, නමුත් නවීන පරිගණකවල, ROM වර්ග වන EEPROM (විද්‍යුත් වශයෙන් මකා දැමිය හැකි වැඩසටහන්ගත කළ හැකි කියවීමට පමණක් මතකය) හෝ ෆ්ලෑෂ් මතකය බහුලව භාවිතා වේ. වර්ගය කුමක් වුවත්, BIOS මූලික වශයෙන් ROM හි ගබඩා කර ඇත.

3. CMOS RAM
CMOS RAM (Complementary Metal-Oxide-Semiconductor RAM) is a small amount of memory used to store configuration settings for the BIOS, such as system time and hardware settings. CMOS RAM is powered by a small battery to retain its data when the computer is off, but it does not store the BIOS program itself. Instead, it stores the settings that the BIOS reads during initialization.
CMOS RAM (Complementary Metal-Oxide-Semiconductor RAM) යනු පද්ධති කාලය සහ දෘඩාංග සැකසුම් වැනි BIOS සඳහා වින්‍යාස සැකසුම් ගබඩා කිරීමට භාවිතා කරන කුඩා මතක ප්‍රමාණයකි. CMOS RAM පරිගණකය ක්‍රියා විරහිත වූ විට එහි දත්ත රඳවා ගැනීමට කුඩා බැටරියකින් බල ගැන්වෙන නමුත් එය BIOS වැඩසටහනම ගබඩා නොකරයි. ඒ වෙනුවට, එය ආරම්භයේදී BIOS කියවන සැකසුම් ගබඩා කරයි.

4. Northbridge Chipset
The Northbridge chipset is part of the motherboard's chipset and is responsible for handling communications between the CPU, RAM, and high-speed peripherals such as the graphics card. It does not store the BIOS program. The Northbridge plays a critical role in performance but is not involved in storing firmware.
Northbridge චිප්සෙට් එක මවුපුවරුවේ චිප්සෙට් එකේ කොටසක් වන අතර CPU,RAM සහ ග්‍රැෆික් කාඩ් වැනි අධිවේගී උපාංග අතර සන්නිවේදනය හැසිරවීමේ වගකීම දරයි. එය BIOS වැඩසටහන ගබඩා නොකරයි. Northbridge කාර්ය සාධනය සඳහා තීරණාත්මක කාර්යභාරයක් ඉටු කරන නමුත් ස්ථිරාංග ගබඩා කිරීමට සම්බන්ධ නොවේ.

5. Southbridge Chipset
The Southbridge chipset handles lower-speed peripheral communications, such as USB, audio, and storage interfaces. Like the Northbridge, it does not store the BIOS. The Southbridge works in conjunction with the Northbridge to manage the flow of data between the CPU and peripheral devices but is not related to firmware storage.
සවුත්බ්‍රිජ් චිප්සෙට් එක USB, ශ්‍රව්‍ය සහ ගබඩා අතුරුමුහුණත් වැනි අඩු-වේග පර්යන්ත සන්නිවේදනයන් හසුරුවයි. Northbridge මෙන්ම එය BIOS ගබඩා නොකරයි. CPU සහ පර්යන්ත උපාංග අතර දත්ත ප්‍රවාහය කළමනාකරණය කිරීම සඳහා Southbridge, Northbridge සමඟ එක්ව ක්‍රියා කරන නමුත් ස්ථිරාංග ගබඩාවට සම්බන්ධ නොවේ.

So, the BIOS is stored in ROM because ROM provides the necessary non-volatile storage that ensures the BIOS is always available upon powering on the computer. The other options (cache memory, CMOS RAM, Northbridge, and Southbridge chipsets) serve different roles in the computer's architecture and are not suitable for storing the BIOS firmware.
එබැවින්, BIOS, ROM හි ගබඩා කර ඇත්තේ ROM මඟින් අවශ්‍ය නශ්‍ය නොවන ආචයනය සපයන නිසා පරිගණකයේ බල ගැන්වීමේදී BIOS සැමවිටම පවතින බව සහතික කරයි. අනෙකුත් විකල්ප (cache memory, CMOS RAM, Northbridge, and Southbridge chipsets) පරිගණක ආකෘතියේ විවිධ භූමිකාවන් ඉටු කරන අතර BIOS ස්ථිරාංග ගබඩා කිරීම සඳහා සුදුසු නොවේ.

2. Which of the following is considered the first commercially available personal computer?
පහත සඳහන් ඒවායින් වාණිජමය වශයෙන් මිලදී ගතහැකි පළමු පුද්ගලික පරිගණකය ලෙස සැලකෙන්නේ කුමක්ද?

1) IBM 360 2) ALTAIR 8800 3) ENIAC 4) Commodore PET 5) Apple III

Answer : 2)

The answer is **ALTAIR 8800**, which is considered the first commercially available personal computer. It was introduced in 1975 and became popular among hobbyists and early computer enthusiasts. The **ALTAIR 8800** was a microcomputer that could be purchased as a kit and assembled at home. It used the Intel 8080 microprocessor and is credited with sparking the personal computing revolution. Although it lacked the user-friendly interfaces of later computers, its affordability and potential for customization made it accessible to the public, marking the beginning of the personal computer era.
පිළිතුර වාණිජමය වශයෙන් ලබා ගත හැකි පළමු පුද්ගලික පරිගණකය ලෙස සැලකෙන **ALTAIR 8800** වේ. එය 1975 දී හඳුන්වා දුන් අතර විනෝදාංග කරන්නන් සහ මුල් පරිගණක ලෝලීන් අතර ජනප්‍රිය විය. **ALTAIR 8800** යනු කට්ටලයක් ලෙස මිල දී ගෙන නිවසේදී එකලස් කළ හැකි ක්ෂුද්‍ර පරිගණකයකි. එය **ALTAIR 8800** ක්ෂුද්‍ර සකසනය භාවිතා කරන ලද අතර පුද්ගලික පරිගණක විප්ලවය ඇති කළ බවට ගෞරවය හිමි වේ. පසුකාලීන පරිගණකවල පරිශීලක-හිතකාමී අතුරුමුහුණත් එහි නොතිබුණද, එහි දැරිය හැකි මිල සහ අභිරුචිකරණය සඳහා ඇති හැකියාව නිසා එය මහජනතාවට ප්‍රවේශ විය හැකි අතර එය පුද්ගලික පරිගණක යුගයේ ආරම්භය සනිටුහන් කළේය.

So, the correct answer is 2.
එබැවින්, නිවැරදි පිළිතුර 2 වේ.

3. What is the one's complement of the answer obtained by calculating 11101001_2 - 10011101_2 ?
11101001_2 - 10011101_2 යන ගණනය කිරීමෙන් ලැබෙන පිළිතුරෙහි එකෙහි අනුපූරකය කුමක්වේද?

1) 10110011_2 2) $\underline{01001100_2}$ 3) 10110100_2 4) 01001101_2 5) 11001100_2

Answer : 2)

Convert into Decimal															
11101001_2								10011101_2							
1	1	1	0	1	0	0	1	1	0	0	1	1	1	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1
233_{10}								157_{10}							

- $233_{10} - 157_{10} = 76_{10}$

Decimal to Binary							
76_{10}							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	1	0	0	1	1	0	0
01001100_2							

The one's complement of a positive number is its reciprocal representation.
ධන සංඛ්‍යාවක එකෙහි අනුපූරකය එහි ද්විමය නිරූපණයම වේ.

So, the correct answer is 2.
එබැවින් නිවැරදි පිළිතුර 2 වේ.

4. What is the final answer obtained when the following Boolean expression is simplified?
පහත බූලිය ප්‍රකාශනය සුළු කළ විට ලැබෙන අවසාන පිළිතුර කුමක්ද?

$1(\overline{CA}+BB)+BC$

1) 1 2) $\overline{C}+B$ 3) BC 4) $\underline{\overline{C}+\overline{A}+B}$ 5) BC+A

Answer : 4)

Let's get the answer as below.
පහත පරිදි පිළිතුර ලබාගනිමු.

$1(\overline{CA}+BB)+BC$
$= 1(\overline{C}+\overline{A}+BB)+BC$ De Morgan's law ද මුවර් ප්‍රමේයය
$= \overline{C}+\overline{A}+BB+BC$ Identity law සර්වසාම්‍ය න්‍යාය
$= \overline{C}+\overline{A}+B+BC$ Idempotent law තදේවභාවී න්‍යාය
$= \overline{C}+\overline{A}+B$ Redundancy law සමතිරික්ත න්‍යාය

So, the answer is 4).
එබැවින්, පිළිතුර 4) වේ.

5. The answer to the booleon expression $F=\overline{(\bar{X}.\bar{Y})}.(\overline{X+Y})$ when simplified using De Morgan's law,
$F=\overline{(\bar{X}.\bar{Y})}.(\overline{X+Y})$ ප්‍රකාශය ද මුවාර් න්‍යාය භාවිතයෙන් සුළු කළ විට පිළිතුර වනුයේ,

1) <u>0</u> 2) 1 3) X 4) Y 5) X.Y

Answer : 1)

Let's get the answer as below.
පහත පරිදි පිළිතුර ලබාගනිමු.

$F=\overline{(\bar{X}.\bar{Y})}.(\overline{X+Y})$	
$(\bar{\bar{X}}+\bar{\bar{Y}}).(\overline{X+Y})$	De Morgan's law ද මුවාර් ප්‍රමේයය
$(X+\bar{\bar{Y}}).(\overline{X+Y})$	Double Inverse law ද්විත්ව ප්‍රතිලෝම න්‍යාය
$(X+Y).(\overline{X+Y})$	Double Inverse law ද්විත්ව ප්‍රතිලෝම න්‍යාය
$(X+Y).\bar{X}\,\bar{Y}$	De Morgan's law ද මුවාර් ප්‍රමේයය
$(\bar{X}\,\bar{Y}\,X)+\bar{X}\,\bar{Y}\,Y$	Distribution law විඝටන න්‍යාය
$0+\bar{X}\,\bar{Y}\,Y$	Complement law ප්‍රතිලෝම න්‍යාය
$\bar{X}\,\bar{Y}\,Y$	Identity law සර්වසාම්‍ය න්‍යාය
0	Complement law ප්‍රතිලෝම න්‍යාය

Therefore, the answer is 1).
එමනිසා, පිළිතුර 1) වේ.

6. Which of the following is not a characteristic of flat file architecture, which is a database architecture?
දත්ත සමුදා ආකෘතියක් වන පැතලි ගොනු ආකෘතියේ ලක්ෂණයක් නොවන්නේ මින් කුමක්ද?

1) All data is stored in a single table.
සියලුම දත්ත තනි එක් වගුවක් තුළ ගබඩා කරනු ලබයි.
2) Finding data is difficult.
දත්ත සොයාගැනීම අපහසුය.
3) Difficulty changing the type of data entered.
ඇතුළත් කර ඇති දත්ත වල වර්ගය වෙනස් කිරීමට අපහසු වීම.
4) <u>Updating is easy.</u>
යාවත්කාලීන කිරීම පහසු වේ.
5) The possibility of writing the same data over and over again is high.
එකම දත්තය නැවත නැවත ලියවීමට ඇති හැකියාව වැඩිය.

Answer: 4)

The flat file architecture stores all data in a single table, making it simple in structure but prone to several limitations and inefficiencies. Let's evaluate the options.
පැතලි ගොනු ආකෘති ශිල්පය සියලුම දත්ත තනි වගුවක ගබඩා කරයි, එය ව්‍යුහයෙන් සරල වන නමුත් සීමාවන් සහ අකාර්යක්ෂමතා කිහිපයකට ගොදුරු වේ. විකල්පයන් ඇගයීමට ලක් කරමු.

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
This is true for flat file systems, where data lacks a relational structure.
දත්ත සම්බන්ධතා ව්‍යුහයක් නොමැති පැතලි ගොනු පද්ධති සඳහා මෙය සත්‍ය වේ.

Option / විකල්ප 2)
Without indexing or relational organization, searching for specific data in a flat file is inefficient.
සුචිගත කිරීම හෝ සම්බන්ධතා සංවිධානය කිරීමකින් තොරව, පැතලි ගොනුවක නිශ්චිත දත්ත සෙවීම අකාර්යක්ෂම වේ.

Option / විකල්ප 3)
Modifying the schema or data type is cumbersome because flat files lack flexibility.
පැතලි ගොනුවල නම්‍යශීලී බවක් නොමැති නිසා යෝජනා ක්‍රමය හෝ දත්ත වර්ගය වෙනස් කිරීම අපහසු වේ.

Option / විකල්ප 4)
Incorrect. Updating flat files is challenging due to the absence of indexing, relationships, and data consistency checks.
වැරදියි. සුචිගත කිරීම, සම්බන්ධතා සහ දත්ත අනුකූලතා පරීක්ෂාවන් නොමැති වීම හේතුවෙන් පැතලි ගොනු යාවත්කාලීන කිරීම අභියෝගාත්මක වේ.

Option / විකල්ප 5)
True, because flat files often result in data redundancy and duplication.
සත්‍ය වේ, පැතලි ගොනු බොහෝ විට දත්ත අතිරික්තය සහ අනුපිටපත් ඇති කරයි.

So, the characteristic updating is easy does not apply to flat file architecture.
මේ අනුව, යාවත්කාලීන කිරීම පහසුය යන ලක්ෂණය පැතලි ගොනු ආකෘති ශිල්පයට අදාළ නොවේ.

The correct answer is option 4).
නිවැරදි පිළිතුර විකල්පය 4) වේ.

7. Which of the following is/are true about weak entities?
දුර්වල භූතාර්ථ සම්බන්ධයෙන් පහත සඳහන් කවරක් සත්‍ය ද?

A. Weak entities have their own primary keys.
දුර්වල භූතාර්ථවලට ඔවුන්ගේම ප්‍රාථමික යතුරු ඇත.

B. They participate in a total participation relationship with the strong entity.
ඔවුන් ශක්තිමත් භූතාර්ථ සමඟ පූර්ණ සහභාගිත්ව සම්බන්ධතාවයකට සහභාගී වේ.

C. Weak and strong entities are always part of a one-to-many relationship.
දුර්වල සහ ශක්තිමත් භූතාර්ථ සැම විටම ඒක-බහු සම්බන්ධතාවයක කොටසකි.

D. They are identified using a combination of a partial key and a foreign key.
අර්ධ යතුරක සහ ආගන්තුක යතුරක සංයෝජනයක් භාවිතයෙන් ඒවා හඳුනා ගැනේ.

E. Strong entities must always have a foreign key referencing weak entities.
ශක්තිමත් භූතාර්ථවලට සැම විටම දුර්වල භූතාර්ථවලට යොමු කරන ආගන්තුක යතුරක් තිබිය යුතුය.

1) B only. B පමණි.
2) D only. D පමණි.
3) E only. E පමණි.
4) <u>B and D only.</u> B හා D පමණි.
5) A and C only. A හා C පමණි.

Answer: 4)

Explanation of each statement / එක් එක් ප්‍රකාශය පැහැදිලි කිරීම:

Statement / ප්‍රකාශය A
False. Weak entities do not have independent primary keys. They rely on a combination of a partial key and a foreign key referencing the strong entity.
අසත්‍ය වේ. දුර්වල භූතාර්ථවලට ස්වාධීන ප්‍රාථමික යතුරු නොමැත. ඔවුන් රඳා පවතින්නේ ප්‍රබල භූතාර්ථයක් යොමු කරන අර්ධ යතුරක සහ ආගන්තුක යතුරක එකතුවක් මත ය.

Statement / ප්‍රකාශය B
True. Every weak entity must be associated with a strong entity, indicating total participation.
සත්‍ය වේ. සෑම දුර්වල භූතාර්ථයක්ම පූර්ණ සහභාගීත්වය පෙන්නුම් කරමින් ශක්තිමත් භූතාර්ථයක් සමඟ සම්බන්ධ විය යුතුය.

Statement / ප්‍රකාශය C
False. The relationship cardinality depends on the specific database design.
අසත්‍ය වේ. සම්බන්ධතා ගණනීයතාවය විශේෂිත දත්ත සමුදා නිර්මාණය මත රඳා පවතී.

Statement / ප්‍රකාශය D
True. Weak entities are uniquely identified by combining their partial key with the foreign key from the strong entity.
සත්‍ය වේ. දුර්වල භූතාර්ථ අනන්‍ය ලෙස හඳුනාගනු ලබන්නේ ඒවායේ අර්ධ යතුර ප්‍රබල භූතාර්ථයේ ආගන්තුක යතුර සමඟ ඒකාබද්ධ කිරීමෙනි.

Statement / ප්‍රකාශය E
False. It is weak entities that reference strong entities using a foreign key, not the other way around.
අසත්‍ය වේ. ආගන්තුක යතුරක් භාවිතා කරමින් ශක්තිමත් භූතාර්ථ යොමු කරන්නේ දුර්වල භූතාර්ථ මිස අනෙක් අතට නොවේ.

Therefore, the correct answer is 4) B and D only.
එබැවින්, නිවැරදි පිළිතුර 4) B සහ D පමණි වේ.

8. What will be the output of the following Python program?
පහත පයිතන් වැඩසටහනේ ප්‍රතිදානය කුමක්ද?

```
a = [10, 20, 30, 40]
b = a.copy()
b[1] = 99
print(a)
print(b)
```

1) `[10, 99, 30, 40]` `[10, 99, 30, 40]`
2) `[10, 20, 30, 40]` `[99, 20, 30, 40]`
3) `[10, 99, 30, 40]` `[99, 20, 30, 40]`
4) `[10, 20, 30, 40]` `[10, 99, 30, 40]`
5) `Error: 'list' object has no attribute 'copy'`

Answer: 4)

Explanation / පැහැදිලි කිරීම:

The Python program demonstrates the use of the `copy()` method to create a copy of a list. Initially, `a` is defined as `[10, 20, 30, 40]`. When `b = a.copy()` is executed, a new list `b` is created as a copy of `a`, meaning it contains the same elements but is a separate object in memory. Modifying `b[1] = 99` changes only the second element of the copied list `b` to `99`, leaving the original list `a` unchanged. Consequently, when `a` is printed, it outputs `[10, 20, 30, 40]`, reflecting its original state, while `b` outputs `[10, 99, 30, 40]`, showing the updated value. This behavior highlights that, changes to the copied list do not affect the original list.
ලැයිස්තුවක පිටපතක් නිර්මාණය කිරීම සඳහා `copy()` ක්‍රමය භාවිතා කිරීම පයිතන් වැඩසටහන මගින් පෙන්නුම් කරයි. මුලදී, `a` යනු `[10, 20, 30, 40]` ලෙස අර්ථ දැක්වේ. `b = a.copy()` ක්‍රියාත්මක කළ විට, `a` හි පිටපතක් ලෙස නව ලැයිස්තුවක් `b` නිර්මාණය වේ, එනම් එහි එකම මූලාංග අඩංගු නමුත් එය මතකයේ වෙනම වස්තුවකි. `b[1] = 99` වෙනස් කිරීමෙන් පිටපත් කළ ලැයිස්තුවේ `b` දෙවන මූලාංගය පමණක් `99` දක්වා වෙනස් කරයි, මුල් ලැයිස්තුව නොවෙනස්ව පවතී. එහි ප්‍රතිඵලයක් ලෙස, `a` මුද්‍රණය කළ විට, එහි මුල් තත්ත්වය පිළිබිඹු කරමින් `[10, 20, 30, 40]` ප්‍රතිදානය කරන අතර `b` ප්‍රතිදානය `[10, 99, 30, 40]` යාවත්කාලීන කළ අගය පෙන්වයි. පිටපත් කළ ලැයිස්තුවේ වෙනස්කම් මුල් ලැයිස්තුවට බලපාන්නේ නැති බව මෙම හැසිරීම උද්දීපනය කරයි.

Final Output / අවසාන ප්‍රතිදානය:

```
[10, 20, 30, 40]
[10, 99, 30, 40]
```

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

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

```
s = "Python"
print(s[::-1])
```

1) `Python` 2) `nohtyP` 3) `P y t h o n` 4) `n o h t y P` 5) Error

Answer: 2)

Explanation / පැහැදිලි කිරීම:

The given Python code assigns the string "Python" to the variable s. The slicing operation `s[::-1]` is used to reverse the string. In Python, slicing allows selecting specific portions of a sequence, such as a string, list, or tuple. Here, the `[::-1]` syntax works as follows: the first colon (`:`) selects all characters in the string, and the `-1` step specifies that the characters should be taken in reverse order. As a result, the string "Python" is reversed to "nohtyP". The `print()` function then outputs this reversed string. Therefore, the final output of the code is "nohtyP".
ලබා දී ඇති Python කේතය "Python" string එකක් s විචල්‍යයට පවරයි. String එකක් ප්‍රතිලෝම කිරීමට `s[::-1]` slice කැපීම භාවිතා කරයි. Python හි, string, list, tuple වැනි අනුපිළිවෙලක නිශ්චිත කොටස් තෝරාගැනීමට slice කැපීම ඉඩ දෙයි. මෙහිදී, `[::-1]` වාක්‍ය ඛණ්ඩය පහත පරිදි ක්‍රියා කරයි: පළමු තිරුව (`:`) string හි සියලුම අක්ෂර තෝරන අතර -1 පියවර මඟින් අක්ෂර ප්‍රතිලෝම අනුපිළිවෙලට ගත යුතු බව සඳහන් කරයි. එහි ප්‍රතිඵලයක් වශයෙන්, Python string එක "nohtyP" වෙත ආපසු හැරේ. එවිට `print()` ශ්‍රිතය මෙම ප්‍රතිලෝම string එක ප්‍රතිදානය කරයි. එබැවින්, කේතයේ අවසාන ප්‍රතිදානය "nohtyP" වේ.

Therefore, the correct answer is 2)
එබැවින්, නිවැරදි පිළිතුර 2)

10. What will be the output of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
def modify_list(lst, value):
    lst[0] *= 2
    lst.append(value)
    lst = lst + [100]
    return lst
x = [5, 10, 15]
y = modify_list(x, 20)
print("x:", x)
print("y:", y)
```

1) `x: [5, 10, 15]`
   `y: [10, 10, 15, 20, 100]`

2) `x: [10, 10, 15, 20]`
   `y: [10, 10, 15, 20, 100]`

3) `x: [10, 10, 15, 20]`
   `y: [5, 10, 15]`

4) `x: [10, 10, 15, 20]`
   `y: [5, 10, 15, 100]`

5) The code will throw an error.

Answer : 2)

- **`x = [5, 10, 15]`**
  The list x is initialized.
  ලැයිස්තුව x ආරම්භ කර ඇත.

- **`modify_list(lst, value)`**

1. **`lst[0] *= 2:`**
   Modifies the first element of `lst` by multiplying it by 2. Since `lst` is passed by reference, this change affects the original list x.
   `lst` හි පළමු මූලාංගය 2 න් ගුණ කිරීමෙන් වෙනස් කරයි. `lst` යොමු කිරීම මගින් සම්මත කර ඇති බැවින්, මෙම වෙනස මුල් ලැයිස්තුව x ට බලපායි.

2. **`lst.append(value):`**
   Appends value (20) to the list, modifying x further.
   ලැයිස්තුවට අගය (20) එකතු කරයි, x තවදුරටත් වෙනස් කරයි.

3. **`lst = lst + [100]:`**
   This creates a new list by concatenating `lst` with `[100]`. However, this new list is not reflected in `x` because `lst` is now pointing to a new object, breaking the reference to the original list.
   මෙය `[100]` සමඟ `lst` සම්බන්ධ කිරීමෙන් නව ලැයිස්තුවක් නිර්මාණය කරයි. කෙසේ වෙතත්, මෙම නව ලැයිස්තුව `x` හි පිළිබිඹු නොවේ, මන්ද, `lst` දැන් නව වස්තුවක් වෙත යොමු කරමින්, මුල් ලැයිස්තුවට යොමු කිරීම බිඳ දමයි.

4. **`return lst`**
   Returns the modified list, including `[100]`.
   `[100]`ඇතුළුව වෙනස් කළ ලැයිස්තුව ආපසු ලබා දෙයි.

- **`y = modify_list(x, 20):`**
  `x` is modified to `[10, 10, 15, 20]`, but `y` gets the value `[10, 10, 15, 20, 100]`.
  `x`, `[10, 10, 15, 20]` ලෙස වෙනස් කර ඇත, නමුත් `y` අගය `[10, 10, 15, 20, 100]` ලබා ගනී.

Output / ප්‍රතිදානය

- `x: [10, 10, 15, 20]` (modified in-place before `lst` was reassigned / `lst` නැවත පැවරීමට පෙර ස්ථානයේ වෙනස් කරන ලදි).
- `y: [10, 10, 15, 20, 100]` (returned from the function / ශ්‍රිතයෙන් ආපසු ලබා ගනී).

The correct answer is 2).
නිවැරදි පිළිතුර 2) වේ.