සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved]

Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 291 MARKING

2025/09/27
ictfromabc
Ravindu Bandaranayake

1. Which of the following statements about floppy disks is true?
නම්‍ය තැටි පිළිබඳ පහත සඳහන් කුමන ප්‍රකාශය සත්‍ය වන්නේ ද?

   1) Floppy disks have a larger storage capacity compared to USB flash drives.
   USB සැනෙලි ධාවක වලට සාපේක්ෂව නම්‍ය තැටිවල විශාල ගබඩා ධාරිතාවක් ඇත.

   2) Floppy disks use optical storage technology to store data.
   නම්‍ය තැටි දත්ත ගබඩා කිරීම සඳහා දෘශ්‍ය ගබඩා තාක්ෂණය භාවිතා කරයි.

   3) Floppy disks were commonly used for software distribution and data transfer.
   මෘදුකාංග බෙදා හැරීම සහ දත්ත හුවමාරුව සඳහා නම්‍ය තැටි බහුලව භාවිතා විය.

   4) Floppy disks are known for their high data transfer speeds and durability.
   නම්‍ය තැටි ඒවායේ ඉහළ දත්ත හුවමාරු වේගය සහ කල්පැවැත්ම සඳහා ප්‍රසිද්ධය.

   5) Floppy disks are still widely used in modern computers for data storage.
   නවීන පරිගණකවල දත්ත ගබඩා කිරීම සඳහා නම්‍ය තැටි තවමත් බහුලව භාවිතා වේ.

   Answer : 3)

   The correct answer is 3) Floppy disks were commonly used for software distribution and data transfer.
   නිවැරදි පිළිතුර වන්නේ 3) මෘදුකාංග බෙදා හැරීම සහ දත්ත හුවමාරුව සඳහා නම්‍ය තැටි බහුලව භාවිතා විය.

   Floppy disks were once a popular medium for sharing software, transferring files, and creating backups, especially during the 1980s and 1990s. They were small, portable, and easy to use, but had very limited storage capacity (usually 1.44 MB for the standard 3.5-inch disk) compared to today's devices. The other options are false: USB drives have far more storage than floppy disks, floppies did not use optical storage (they used magnetic storage), they were slow and fragile, and they are no longer widely used in modern computers.
   විශේෂයෙන් 1980 වේදී සහ 1990 වේදී නම්‍ය තැටි මෘදුකාංග බෙදා ගැනීම, ගොනු මාරු කිරීම සහ උපස්ථ නිර්මාණය කිරීම සඳහා ජනප්‍රිය මාධ්‍යයක් විය. ඒවා කුඩා, අතේ ගෙන යා හැකි සහ භාවිතයට පහසු වූ නමුත් අද උපාංග හා සසඳන විට ඉතා සීමිත ගබඩා ධාරිතාවක් (සාමාන්‍යයෙන් සම්මත 3.5-අඟල් තැටිය සඳහා (1.44 MB) තිබුණි. අනෙක් විකල්ප අසත්‍ය වේ: USB ධාවක නම්‍ය තැටිවලට වඩා බොහෝ ගබඩා ප්‍රමාණයක් ඇත, නම්‍ය තැටි දෘශ්‍ය ගබඩාව භාවිතා කළේ නැත (ඒවා චුම්භක ගබඩාව භාවිතා කළේය), ඒවා මන්දගාමී සහ බිඳෙන සුළු වූ අතර ඒවා තවදුරටත් නවීන පරිගණකවල බහුලව භාවිතා නොවේ.

2. Choose the answer that includes an example of volatile memory and a characteristic of it.
නශ්‍ය මතකයට උදාහරණයක් සහ එහි ලක්ෂණයක් ඇතුලත් වන පිළිතුර තෝරන්න.

   1) Static random access memory does not store electrical charge for long periods of time.
   ස්ථිතික සසම්භාවී මතකය දිගු කාලයක් විද්‍යුත් ආරෝපණය රඳවා තබා නොගනී.

   2) Dynamic random access memory can store more capacity in less space.
   ගතික සසම්භාවී ප්‍රවේශ මතකයෙහි වැඩි ධාරිතාවයක් කුඩා ඉඩ ප්‍රමාණයක රඳවා තැබිය හැකිය.

   3) Data in primary read only memory is entered by the motherboard manufacturer.
   ප්‍රාථමික පඨනමාත්‍ර මතකයෙහි දත්ත ඇතුලත් කරනු ලබන්නේ මවු පුවරුව නිෂ්පාදන සමාගම විසිනි.

   4) Compact discs, digital multiple discs belong to the category of memory units.
   සංයුක්ත තැටි, සංඛ්‍යාංක බහුවිධ තැටි අයත් වන්නේ මතක ඒකකයන් ගණයටය.

   5) Magnetic disks are used to secure data entered into computers.
   චුම්භකිත තැටි, පරිගණක තුල ඇතුලත් කරන දත්ත සුරක්ෂිත කිරීම සඳහා භාවිතා කරනු ලබයි.

   Answer : 2)

1) Static random access memory (SRAM) is indeed a type of memory, but the statement provided is about SRAM, not volatile memory in general. Additionally, the characteristic mentioned, "does not store electrical charge for long periods of time," is accurate for SRAM, but it doesn't represent volatile memory as a whole.
ස්ථිතික සසම්භාවී ප්‍රවේශ මතකය (SRAM) ඇත්තෙන්ම මතක වර්ගයකි, නමුත් සපයා ඇති ප්‍රකාශය SRAM ගැන මිස පොදුවේ නෂ්‍ය මතකය සම්බන්ධව නොවේ. මීට අමතරව, සඳහන් කර ඇති ලක්ෂණය, දිගු කාලයක් සඳහා විදුලි ආරෝපණ ගබඩා නොකරයි, SRAM සඳහා නිවැරදි වේ, නමුත් එය සමස්තයක් ලෙස නෂ්‍ය මතකය නියෝජනය නොකරයි.

2) Dynamic random access memory (DRAM) is a type of volatile memory. Volatile memory refers to memory that loses its stored information when the power is turned off. DRAM stores data in a cell that needs to be periodically refreshed, which contrasts with static random access memory (SRAM) that does not need to be refreshed as frequently. The characteristic mentioned in option 2, about storing more capacity in less space, is indeed a feature of DRAM.
ගතික සසම්භාවී ප්‍රවේශ මතකය (DRAM) යනු නෂ්‍ය මතක වර්ගයකි. නෂ්‍ය මතකය යනු බලය අක්‍රිය වූ විට එහි ගබඩා කර ඇති තොරතුරු නැති වන මතකයයි. DRAM වරින් වර නැවුම් කළ යුතු ලෙස දත්ත ගබඩා කරයි, එය නිතර නිතර නැවුම් කිරීමට අවශ්‍ය නොවන ස්ථිතික සසම්භාවී ප්‍රවේශ මතකය SRAM සමඟ වෙනස් වේ. විකල්ප 2 හි සඳහන් වන ලක්ෂණය, “අඩු ඉඩක් තුළ වැඩි ධාරිතාවක් ගබඩා කිරීම” ගැන, ඇත්තෙන්ම DRAM හි ලක්ෂණයකි.

3) The statement describes primary read-only memory (ROM), not volatile memory. ROM is non-volatile memory, meaning it retains its stored data even when the power is turned off. Additionally, the characteristic mentioned, "entered by the motherboard manufacturer," is specific to the manufacturing process of ROM chips and doesn't represent a characteristic of volatile memory.
මෙම ප්‍රකාශය පඨන මාත්‍රා මතකය (ROM) විස්තර කරයි, එය නෂ්‍ය මතකය නොවේ. ROM යනු නෂ්‍ය නොවන මතකයකි, එනම් බලය අක්‍රිය වූ විට පවා එහි ගබඩා කර ඇති දත්ත රඳවා තබා ගනී. මීට අමතරව, සඳහන් කර ඇති ලක්ෂණය, “මවුපුවරු නිෂ්පාදකයා විසින් ඇතුල් කරන ලදී”. ROM චිප්ස් නිෂ්පාදන ක්‍රියාවලියට විශේෂිත වන අතර නෂ්‍ය මතකයේ ලක්ෂණයක් නියෝජනය නොකරයි.

4) Compact discs (CDs) and digital versatile discs (DVDs) are examples of non-volatile storage media, not volatile memory. These storage media retain data even when the power is turned off, unlike volatile memory. The statement is not about volatile memory.
සංයුක්ත තැටි (CD) සහ ඩිජිටල් බහුකාර්ය තැටි (DVD) නෂ්‍ය මතකය නොව නෂ්‍ය නොවන ගබඩා මාධ්‍ය සඳහා උදාහරණ වේ. නෂ්‍ය මතකය මෙන් නොව බලය අක්‍රිය වූ විට පවා මෙම ගබඩා මාධ්‍ය දත්ත රඳවා ගනී. මෙම ප්‍රකාශය නෂ්‍ය මතකය ගැන නොවේ.

5) Magnetic disks, such as hard disk drives (HDDs), are examples of non-volatile storage devices, not volatile memory. While they are used to store data in computers, they retain data even when the power is turned off, which is a characteristic of non-volatile memory. The statement does not describe volatile memory.
දෘඪ තැටි ධාවකයන් (HDD) වැනි චුම්බක තැටි නෂ්‍ය මතකය නොව නෂ්‍ය නොවන ගබඩා උපාංග සඳහා උදාහරණ වේ. ඒවා පරිගණකවල දත්ත ගබඩා කිරීමට භාවිතා කරන අතර, බලය අක්‍රිය වූ විට පවා ඒවා දත්ත රඳවා තබා ගනී, එය නෂ්‍ය නොවන මතකයේ ලක්ෂණයකි. මෙම ප්‍රකාශය නශ්‍ය මතකය විස්තර නොකරයි.

Accordingly, the answer is 2). / ඒ අනුව පිළිතුර වන්නේ 2) වේ.

3. What is the binary equivalent to decimal 68.125?
දශමය 68.125 ට තුල්‍ය ද්විමය අගය කුමක්ද?

1) 00100010.01_2 2) <u>01000100.001_2</u> 3) 01000100.1_2 4) 01111101.0_2 5) 10111011.110_2

Answer : 2)

First, let's Convert 68 to binary / මුලින්ම 68 ද්විමය බවට පරිවර්තනය කරමු							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	0	0	1	0	0
$68 = 01000100_2$							

Now let's convert 0.125 into binary / 0.125 ද්විමය බවට පරිවර්තනය කරමු
. 125 × 2 0 . 25 × 2 0 . 5 × 2 1 . 00
$0.125 = 001_2$

$68.125 = 01000100.001_2$

Therefore, the correct answer is 2) 01000100.001.
එබැවින්, නිවැරදි පිළිතුර 2) 01000100.001 වේ.

4. Which of the following is TRUE about a full adder?
පූර්ණ ආකලක සම්බන්ධයෙන් පහත සඳහන් දේවලින් සත්‍ය වන්නේ කුමක්ද?

1) It can be built using two half adders and an OR gate
එය අර්ධ ආකලක දෙකක් සහ OR ද්වාරයක් භාවිතයෙන් ගොඩනගා ගත හැකිය.
2) It requires only one XOR gate
එයට අවශ්‍ය වන්නේ එක් XOR ද්වාරයක් පමණි.
3) It cannot handle carry-in
එයට රැගෙන යාම හැසිරවිය නොහැක.
4) It is slower than a half adder in all cases
එය සෑම අවස්ථාවකම අර්ධ ආකලකයකට වඩා මන්දගාමී වේ.
5) It can subtract numbers directly
එයට සංඛ්‍යා කෙලින්ම අඩු කළ හැකිය.

Answer: 1)

A full adder adds three inputs: A, B, and carry-in (Cin), producing a sum and a carry-out (Cout).
පූර්ණ ආකලක ආදාන තුනක් එකතු කරයි: A, B සහ රැගෙන යාම (Cin), එකතුවක් සහ රැගෙන යාමක් (Cout) නිපදවයි.

It can be constructed by combining two half adders (each adds two inputs) and an OR gate to combine the carry outputs.
එය අර්ද ආකලක දෙකක් (එක් එක් යෙදවුම් දෙකක් එකතු කරයි) සහ රැගෙන යාමේ ප්‍රතිදාන ඒකාබද්ධ කිරීම සඳහා OR ද්වාරයක් ඒකාබද්ධ කිරීමෙන් ගොඩනගා ගත හැකිය.

Option 2)
A full adder requires more than one XOR gate for the sum.
පූර්ණ ආකලක එකතුව සඳහා XOR ද්වාර එකකට වඩා අවශ්‍ය වේ.

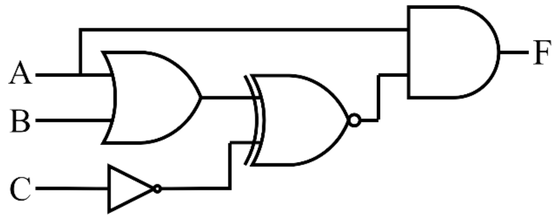
Option 3)
A full adder is specifically designed to handle carry-in, unlike a half adder.
පූර්ණ ආකලක විශේෂයෙන් නිර්මාණය කර ඇත්තේ අර්ද ආකලක මෙන් නොව රැගෙන යාම හැසිරවීමටයි.

Option 4)
While slightly more complex, it is not necessarily "slower in all cases"; speed depends on implementation.
තරමක් සංකීර්ණ වුවද, එය අනිවාර්යයෙන්ම “සියලු අවස්ථාවන්හිදී මන්දගාමීම නොවේ” වේගය ක්‍රියාත්මක කිරීම මත රඳා පවතී.

Option 5)
Full adders perform addition, not subtraction directly.
පූර්ණ ආකලක එකතු කිරීම සිදු කරයි, අඩු කිරීම සෘජුවම සිදු නොකරයි.

Therefore the correct answer is option 1) / එබැවින් නිවැරදි පිළිතුර විකල්පය 1) වේ

5. Consider the following logic circuit.
පහත දැක්වෙන තාර්කික පරිපථය සලකන්න.

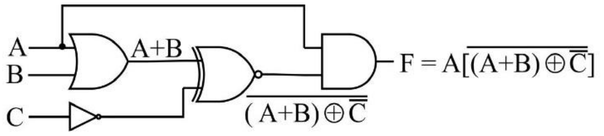


When the truth table related to the above logic circuit is constructed, the answer related to column F is,
ඉහත තාර්කික පරිපථයට අදාළ සත්‍යතා වගුව ගොඩනැගූ විට එහි F තීරුවට අදාළ පිළිතුර වන්නේ,

1)	2)	3)	4)	5)
F	F	F	F	F
1	0	1	0	1
1	0	0	0	1
1	0	1	0	1
0	1	0	0	0
0	1	1	1	1
0	1	0	0	0
1	0	0	1	1
1	1	1	0	1

Answer : 4)

The logic circuit corresponding to the above question, along with its associated truth table, is shown below.
ඉහත ප්‍රශ්ණයට අදාළ තාර්කික පරිපථය සහා එම පරිපථයට අදාළ සත්‍යතා වගුව පහත දැක්වේ.



A	B	C	$\bar{C}$	(A + B)	$\overline{(A+B)\oplus\bar{C}}$	F = A[$\overline{(A+B)\oplus\bar{C}}$]
0	0	0	1	0	0	0
0	0	1	0	0	1	0
0	1	0	1	1	1	0
0	1	1	0	1	0	0
1	0	0	1	1	1	1
1	0	1	0	1	0	0
1	1	0	1	1	1	1
1	1	1	0	1	0	0

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

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

```
x = [1, 2, 3]
x.append([4, 5])
print(len(x)+24//7)
```

1) 8   2) 7   3) 3   4) 4   5) 5

Answer : 4)

Answer : 2)

Let's consider the process of this python code step by step.
මෙම පයිතන් කේතයේ ක්‍රියාවලිය පියවරෙන් පියවර සලකා බලමු.

`x = [1, 2, 3]`
A list `x` is created with three elements, `[1, 2, 3]`.
මූලාංග තුනකින් සමන්විත `x` ලැයිස්තුවක් `[1, 2, 3]` ලෙස සාදනු ලැබේ.

`x.append([4, 5])`
The `.append()` method is used to add a new element to the list `x`. The element added is a list `[4, 5]`, so after this operation, `x` becomes `[1, 2, 3, [4, 5]]`. This means that the list `[4, 5]` is nested within `x` as a single element.
`x` ලැයිස්තුවට නව මූලාංගයක් එක් කිරීමට `.append()` ශ්‍රිතය භාවිතා කරයි. එකතු කරන ලද මූලාංගය `[4, 5]` ලෙස ලැයිස්තුවකි , එබැවින් මෙම මෙහෙයුමෙන් පසුව `x`, `[1, 2, 3, [4, 5]]` බවට පත්වේ. මෙයින් අදහස් කරන්නේ `[4, 5]` ලැයිස්තුව තනි මූලාංගයක් ලෙස `x` තුළ තැන්පත් කර ඇති බවයි.

`print(len(x)+24//7)`

- `len(x)`: The `len()` function returns the number of elements in the list `x`. After the `append()`, `x` contains 4 elements (1, 2, 3, and `[4, 5]`), so `len(x)` returns 4.
  `len()` ශ්‍රිතය `x` ලැයිස්තුවේ ඇති මූලාංග ගණන ලබා දෙයි. `append()`, `x` හි මූලාංග 4 ක් (1, 2, 3, සහ `[4, 5]`), අඩංගු වේ, එබැවින් `len(x)` මගින් 4 ලබා දෙයි.

- `24//7`: This is integer division of 24 by 7, which results in 3 (since 24 ÷ 7 equals 3 with a remainder of 3, but integer division discards the remainder).
  මෙය 24 න් 7 කින් නිඛිල බෙදීමකි, එහි ප්‍රතිඵලය 3 වේ (24 ÷ 7, 3 ට සමාන වන අතර 3 ක් ඉතිරි වේ, නමුත් පූර්ණ සංඛ්‍යා බෙදීම ඉතිරිය ඉවතලයි).

- `len(x) + 24//7`: This adds 4 (from `len(x)`) and 3 (from `24//7`), resulting in 7.
  මෙය 4 (`len(x)`) සහ 3 (`24//7`) එකතු කරයි, එහි ප්‍රතිඵලය 7 වේ.

So, the final output of the code is 7.
එබැවින්, කේතයේ අවසාන ප්‍රතිදානය 7 වේ.

The correct answer number is 2).
නිවැරදි පිළිතුරු අංකය 2) වේ.

7. What will the following python code output if the input is `'meditation'` ?
පහත පයිතන් කේතයේ ආදානය `'meditation'` වේ නම් ප්‍රතිදානය කුමක් වේවිද?

```
name = input()
print(list(name))
```

1) `meditation`
2) `['meditation']`
3) <u>`['m', 'e', 'd', 'i', 't', 'a', 't', 'i', 'o', 'n']`</u>
4) `['med', 'itation']`
5) `Error`

Answer : 3)

The `input()` function takes the string `'meditation'` as input.
`input()` ශ්‍රිතය `'meditation'` string එක ආදානය ලෙස ගනී.

The `list()` function then converts this string into a list of individual characters.
`list()` ශ්‍රිතය මෙම string එක තනි අක්ෂර ලැයිස්තුවක් බවට පරිවර්තනය කරයි.

Therefore, the string `'meditation'` is split into its individual characters, resulting in `['m', 'e', 'd', 'i', 't', 'a', 't', 'i', 'o', 'n']`.

එබැවින්, 'meditation' string එක එහි තනි අක්ෂරවලට බෙදී, ['m', 'e', 'd', 'i', 't', 'a', 't', 'i', 'o', 'n'] ලැබේ.

So, the output of the code when the input is 'meditation' will be 3).
එබැවින්, ආදානය 'meditation' වන විට කේතයේ ප්‍රතිදානය 3) වනු ඇත.

8. What is output of the following python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
print ([1,2,3,4,5,6] [-1:2] [::-2])
```

1) [1,5]　　2) [5,1]　　3) [1,3,5]　　4) [5,3,1]　　5) None of above.
ඉහත කිසිවක් නොවේ.

Answer : 5)

The output of print([1,2,3,4,5,6][-1:2][::-2]) is [] because the slice [-1:2] attempts to retrieve elements starting from index -1 (the last element) up to index 2, but Python slicing works from left to right, so this range is invalid and results in an empty list. When [::-2] is applied to this empty list, it still returns an empty list, since there are no elements to reverse or step through.
print([1,2,3,4,5,6][-1:2][::-2]) හි ප්‍රතිදානය [] වේ. මන්ද, slice [-1:2] උත්සාහ කරයි. index -1
(අවසාන මූලද්‍රව්‍යය) සිට index 2 දක්වා මූලද්‍රව්‍ය ලබා ගැනීමට, නමුත් පයිතන් slice වමේ සිට දකුණට ක්‍රියා කරයි, එබැවින් මෙම පරාසය වලංගු නොවන අතර හිස් ලැයිස්තුවක් ඇති කරයි. මෙම හිස් ලැයිස්තුවට [::-2] යෙදූ විට, ආපසු හැරවීමට හෝ පියවර හරහා යාමට මූලද්‍රව්‍ය නොමැති බැවින් එය තවමත් හිස් ලැයිස්තුවක් ප්‍රතිදානය ලෙස ලබා දෙයි.

So, the correct answer is 5).
එබැවින්, නිවැරදි පිළිතුර 5) වේ.

9. How can you prevent the print function from automatically adding a newline character at the end ?
print ශ්‍රිතය අවසානයේ නව පේළියකින් ස්වයංක්‍රියව ආරම්භ කිරීමේ කාර්යය වළක්වා ගන්නේ කෙසේද?

1) Using the newline=False parameter.
newline=False පරාමිතිය භාවිතා කිරීම.
2) Using the end='' parameter.
end='' පරාමිතිය භාවිතා කිරීම.
3) Using the end='nonewline' parameter.
end='nonewline' පරාමිතිය භාවිතා කිරීම.
4) Using the end='\n' parameter.
end='\n' පරාමිතිය භාවිතා කිරීම.
5) Using the newline='Null' parameter.
newline='Null' පරාමිතිය භාවිතා කිරීම.

Answer : 2)

Let's explain the statements in the above question.
ඉහත ප්‍රශ්නයේ ඇති ප්‍රකාශ මෙසේ පැහැදිලි කරමු.

- **Statement / ප්‍රකාශය 1)**

This option suggests using newline=False, but newline is not a valid parameter for the print function to control the end character behavior.
මෙම විකල්පය මඟින් newline=False භාවිතා කිරීමට යෝජනා කරයි, නමුත් නව රේඛාව යනු අවසාන අක්ෂර හැසිරීම පාලනය කිරීම සඳහා print කාර්යය සඳහා වලංගු පරාමිතියක් නොවේ.

- **Statement / ප්‍රකාශය 2)**

This option correctly suggests using end='' as the parameter. When you specify end='', it tells the print function not to append anything after the printed content, effectively preventing the addition of a newline character.
මෙම විකල්පය නිවැරදිව end='' පරාමිතිය ලෙස භාවිතා කිරීමට යෝජනා කරයි. ඔබ end='' සඳහන් කළ විට, එය මුද්‍රිත අන්තර්ගතයට පසුව කිසිවක් එකතු නොකරන ලෙස මුද්‍රණ ශ්‍රිතයට කියයි, නව රේඛා අක්ෂරයක් එකතු කිරීම ඵලදායී ලෙස වළක්වයි.

- **Statement / ප්‍රකාශය 3)**

This is not a valid parameter for the print function. end should be a string that specifies what should be printed at the end of the output.
මෙය මුද්‍රණ කාර්යය සඳහා වලංගු පරාමිතියක් නොවේ. end යනු ප්‍රතිදානය අවසානයේ මුද්‍රණය කළ යුතු දේ සඳහන් කරන තන්තුවක් විය යුතුය.

- **Statement / ප්‍රකාශය 4)**

This would actually cause the print function to explicitly add a newline character (\n) at the end of the printed output, which is opposite to what we want.
මෙය ඇත්ත වශයෙන්ම මුද්‍රණ ශ්‍රිතය මඟින් අපට අවශ්‍ය දේට ප්‍රතිවිරුද්ධ වන මුද්‍රිත ප්‍රතිදානයේ අවසානයේ නව රේඛා අක්ෂරයක් (\n) පැහැදිලිව එක් කිරීමට හේතු වේ.

- **Statement / ප්‍රකාශය 5)**

Similar to option 3, newline is not a valid parameter for print, and 'Null' is not a recognized value for controlling the end character behavior.
විකල්ප 3 ට සමානව, newline යනු මුද්‍රණය සඳහා වලංගු පරාමිතියක් නොවන අතර "Null" යනු අවසාන අක්ෂර හැසිරීම් පාලනය සඳහා පිළිගත් අගයක් නොවේ.

Therefore, the correct answer among the options provided is 2) Using the end='' parameter. This option correctly specifies using end='' to prevent the print function from adding a newline character at the end of the output.
එබැවින්, ලබා දී ඇති විකල්ප අතරින් නිවැරදි පිළිතුර වන්නේ 2) Using the end='' පරාමිතිය භාවිතා කිරීමයි. මුද්‍රණ ශ්‍රිතය ප්‍රතිදානය අවසානයේ නව රේඛා අක්ෂරයක් එක් කිරීමෙන් වැළැක්වීමට මෙම විකල්පය නිවැරදිව end='' භාවිතා කරයි.

10. Which operator is overloaded by the `__add__()` method?
`__add__()` ක්‍රමය ලෙස දක්වා ඇත්තේ කුමන මෙහෙයවනයද?

1) `-` 2) `*` 3) `/` 4) `+` 5) `%`

Answer : 4)

Explanation of the options / විකල්ප පැහැදිලි කිරීම:

Option / විකල්පය 1)
The - operator is overloaded by the `__sub__()` method. This method defines the behavior for subtraction between objects of a class.
- මෙහෙයවනය `__sub__()` ක්‍රමය මඟින් දක්වා ඇත. මෙම ක්‍රමය පන්තියක (class) වස්තු අතර අඩුකිරීමේ හැසිරීම නිර්වචනය කරයි.

Example:

```
class Number:
    def __init__(self, value):
        self.value = value
    def __sub__(self, other):
        return self.value - other.value
```

Option / විකල්පය 2)
The * operator is overloaded by the `__mul__()` method. This method defines the behavior for multiplication between objects of a class.
* ක්‍රියාකරු `__mul__()` ක්‍රමය මඟින් දක්වා ඇත. මෙම ක්‍රමය පන්තියක (class) වස්තූන් අතර ගුණ කිරීම සඳහා හැසිරීම නිර්වචනය කරයි.

Example:

```
class Number:
    def __init__(self, value):
        self.value = value
    def __mul__(self, other):
        return self.value * other.value
```

Option / විකල්පය 3)
The / operator is overloaded by the `__truediv__()` method. This method defines the behavior for true division between objects of a class.
/ ක්‍රියාකරු `__truediv__()` ක්‍රමය මඟින් දක්වා ඇත. මෙම ක්‍රමය පන්තියක (class) වස්තූන් අතර සැබෑ බෙදීම සඳහා හැසිරීම නිර්වචනය කරයි.

Example:

```
class Number:
    def __init__(self, value):
        self.value = value
    def __truediv__(self, other):
        return self.value / other.value
```

Option / විකල්පය 4)
The + operator is overloaded by the `__add__()` method. This method defines the behavior for addition between objects of a class.
`__add__()` ක්‍රමය මඟින් + ක්‍රියාකරු දක්වා ඇත. මෙම ක්‍රමය පන්තියක (class) වස්තූන් අතර එකතු කිරීම සඳහා හැසිරීම නිර්වචනය කරයි.

Example:

```
class Number:
    def __init__(self, value):
        self.value = value
    def __add__(self, other):
        return self.value + other.value
num1 = Number(10)
num2 = Number(20)
print(num1 + num2)  # Output: 30
```

Option / විකල්පය 5)
The % operator is overloaded by the `__mod__()` method. This method defines the behavior for the modulus operation between objects of a class.
% ක්‍රියාකරු `__mod__()` ක්‍රමය මඟින් දක්වා ඇත. මෙම ක්‍රමය පන්තියක (class) වස්තූ අතර මොඩියුල ක්‍රියාකාරිත්වය සඳහා හැසිරීම නිර්වචනය කරයි.

Example:

```
class Number:
    def __init__(self, value):
        self.value = value
    def __mod__(self, other):
        return self.value % other.value
```

Therefore, the correct answer for the operator that is overloaded by the `__add__()` method in Python is 4) +.
එමනිසා, __add__() ක්‍රමය මඟින් දක්වා ඇති ක්‍රියාකරු සඳහා නිවැරදි පිළිතුර වන්නේ 4) +.