

සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved ]

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

## Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

# 2026 QUIZ NO 489 MARKING

2026/04/13  
ictfromabc  
Ravindu Bandaranayake

1. Match What is the primary function of a NIC (Network Interface Card) in an ICT system?  
ICT පද්ධතියක NIC (භාල අතුරුමුහුණත් කාඩ්පත) හි ප්‍රධාන කාර්යය කුමක්ද?
- 1) Encrypting data for cybersecurity / සයිබර් ආරක්ෂාව සඳහා දත්ත කේතනය කිරීම.
  - 2) Translating programming languages into machine code.  
ක්‍රමලේඛන භාෂා යන්ත්‍ර කේතයට පරිවර්තනය කිරීම.
  - 3) Connecting a computer to a network / පරිගණකයක් භාලයකට සම්බන්ධ කිරීම.
  - 4) Managing memory allocation for programs.  
වැඩසටහන් සඳහා මතක වෙන් කිරීම කළමනාකරණය කිරීම.
  - 5) Assigning IP addresses to networking devices / භාලකරණ උපාංග සඳහා අප ලිපින පැවරීම.

**Answer : 3)**

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

### Option / විකල්ප 1)

Incorrect. Encryption is handled by security protocols (ex: SSL/TLS) or specialized hardware like TPMs (Trusted Platform Modules).

අසත්‍ය වේ. කේතනය හසුරුවනු ලබන්නේ ආරක්ෂක නියමාවලි (උදා: SSL/TLS) හෝ TPM (විශ්වාසදායක වේදිකා මොඩියුල) වැනි විශේෂිත දෘඩාංග මගිනි.

### Option / විකල්ප 2)

Incorrect. This is done by assemblers, compilers or interpreters.

අසත්‍ය වේ. මෙය සිදු කරනු ලබන්නේ ඇසෙම්බ්ලර්, අර්ථ විනයක හෝ පරිවර්තකයන් විසිනි.

### Option / විකල්ප 3)

Correct. A Network Interface Card (NIC) is a hardware component that allows a computer to connect to a network, either wired or wireless. Without a NIC, a computer cannot send or receive data over a local network or the internet.

සත්‍ය වේ. භාල අතුරුමුහුණත් කාඩ්පතක් (NIC) යනු පරිගණකයකට රැහැන්ගත හෝ රැහැන් රහිත භාලයකට සම්බන්ධ වීමට ඉඩ සලසන දෘඩාංග සංරචකයකි. NIC එකක් නොමැතිව, පරිගණකයකට ස්ථානීය භාලයක් හෝ අන්තර්ජාලය හරහා දත්ත යැවීමට හෝ ලබා ගැනීමට නොහැකිය.

### Option / විකල්ප 4)

Incorrect. Memory management is handled by the operating system and RAM, not by a NIC.

අසත්‍ය වේ. මතක කළමනාකරණය හසුරුවන්නේ මෙහෙයුම් පද්ධතිය සහ RAM මගින් මිස NIC එකක් මගින් නොවේ.

### Option / විකල්ප 5)

Incorrect. IP addresses are assigned by a DHCP (Dynamic Host Configuration Protocol) server, not by the NIC itself.

අසත්‍ය වේ. IP ලිපින පවරනු ලබන්නේ NIC විසින්ම නොව DHCP (ගතික ධාරක වින්‍යාස නියමාවලි) සේවාදායකයක් මගිනි.

Therefore, the correct answer is 3).

එබැවින්, නිවැරදි පිළිතුර 3) වේ.

2. What does UTF stand for in computing?

පරිගණකකරණයේදී UTF යන්නෙන් අදහස් කරන්නේ කුමක්ද?

- |                                         |                                     |
|-----------------------------------------|-------------------------------------|
| 1) Universal Text Format                | 2) Unicode Transformation Framework |
| 3) <u>Unicode Transformation Format</u> | 4) Unified Text Function            |
| 5) Unified Transformation Format        |                                     |

**Answer : 3)**

**Explanation / පැහැදිලි කිරීම**

**Option / විකල්පය 1**

Incorrect. Not a valid definition

වැරදියි. වලංගු අර්ථ දැක්වීමක් නොවේ

**Option / විකල්පය 2**

Incorrect. UTF is a format not a framework

වැරදියි. UTF යනු රාමුවක් නොව ආකෘතියකි

**Option / විකල්පය 3**

Correct. UTF stands for Unicode Transformation Format, which is a standardized method for encoding Unicode characters into binary data. Common UTF encodings include UTF-8, UTF-16, and UTF-32, each using different numbers of bytes to represent characters.

නිවැරදියි. UTF යනු Unicode Transformation Format වන අතර එය යුනිකෝඩ් අක්ෂර ද්වීමය දත්ත බවට කේතනය කිරීම සඳහා ප්‍රමිතිගත ක්‍රමයකි. පොදු UTF කේතන ක්‍රම අතරට UTF-8, UTF-16, UTF-32 ඇතුළත් වන අතර, ඒ සෑම එකක්ම අනුලක්ෂණ නිරූපණය කිරීම සඳහා විවිධ බයිට් සංඛ්‍යා භාවිතා කරයි.

**Option / විකල්පය 4**

Incorrect. Not a valid definition

වැරදියි. වලංගු අර්ථ දැක්වීමක් නොවේ

**Option / විකල්පය 5**

Incorrect. Not a valid definition

වැරදියි. වලංගු අර්ථ දැක්වීමක් නොවේ

Therefore, the correct answer is option 3).

එබැවින් නිවැරදි පිළිතුර විකල්පය 3) වේ.

3. Which of the following scenarios best describes a practical use of a combinational circuit?

සංයුක්ත පරිපථයක ප්‍රායෝගික භාවිතයක් වඩාත් හොඳින් විස්තර කරන්නේ පහත සඳහන් කුමන අවස්ථාවකද?

- 1) Storing a password in memory / මතකයේ මුරපදයක් ගබඩා කිරීම
- 2) Switching between two TV channels / රූපවාහිනී නාලිකා දෙකක් අතර මාරු වීම
- 3) Counting the number of vehicles passing a checkpoint  
මුරපොලක් පසු කරන වාහන ගණන ගණනය කිරීම
- 4) Performing binary addition in an ALU / ALU එකක ද්වීමය එකතු කිරීමක් සිදු කිරීම
- 5) Retaining a previous input for future use / අනාගත භාවිතය සඳහා පෙර ආදානයක් රඳවා ගැනීම

**Answer : 4)**

**Explanation / පැහැදිලි කිරීම:**

Binary addition (like half/full adders) is a combinational logic operation.

ද්වීමය එකතු කිරීම (අර්ධ/පූර්ණ ආකලක වැනි) යනු සංයෝජන තාර්කික මෙහෙයුමකි.

**Option / විකල්පය 1)**

Requires a circuit that remembers data even after the inputs are removed (e.g., an SRAM cell, flip-flop, or non-volatile memory). These are sequential circuits because they incorporate feedback or latching to hold state.

ආදාන ඉවත් කිරීමෙන් පසුව පවා දත්ත මතක තබා ගන්නා පරිපථයක් අවශ්‍ය වේ (උදා: SRAM කෙලයක්, පිළි පොළ හෝ නශ්‍ය නොවන මතකය). මේවා අනුක්‍රමික පරිපථ වන්නේ ඒවා ප්‍රතිපෝෂණ හෝ තත්ත්වය රඳවා තබා ගැනීම සඳහා අගුලු දැමීම කරන බැවිනි.

**Option / විකල්පය 2)**

At first glance a simple selector (multiplexer) could be combinational, but in a real TV the “current channel” must be maintained until the user presses a new button. That persistence comes from internal registers or latches – again a sequential characteristic.

පළමු බැල්මට සරල තේරීමක් (බහුකාර්යයක්) සංයෝජන විය හැකි නමුත්, සැබෑ රූපවාහිනියක පරිශීලකයා නව බොත්තමක් එබූ තෙක් වත්මන් නාලිකාවම පවත්වා ගත යුතුය. එම නොනැසී පැවතීම අභ්‍යන්තර රෙජිස්ටර් හෝ latch වලින් පැමිණේ - නැවතත් අනුක්‍රමික ලක්ෂණයකි.

**Option / විකල්පය 3)**

Counting means incrementing a value each time a sensor is triggered. The circuit must remember the previous count, so a counter (built from flip-flops) is required. This is classic sequential logic because past events affect future outputs.

ගණන් කිරීම යනු සංවේදකයක් ක්‍රියාත්මක වන සෑම අවස්ථාවකම අගයක් වැඩි කිරීමයි. පරිපථය පෙර ගණන් කිරීම මතක තබා ගත යුතුය, එබැවින් කවුන්ටරයක් (පිළි පොළ වලින් සාදන ලද) අවශ්‍ය වේ. මෙය සම්භාව්‍ය අනුක්‍රමික තර්කනයකි, මන්ද අතීත සිදුවීම් අනාගත ප්‍රතිදානයන්ට බලපායි.

**Option / විකල්පය 4)**

An adder’s outputs (Sum and Carry) depend only on the present operand bits and not on any stored history. Half-adders and full-adders are purely combinational networks of XOR, AND, and OR/NAND gates, making this the correct choice.

ආකලකයක ප්‍රතිදානයන් (එකතුව සහ රැගෙන යාම) රඳා පවතින්නේ වර්තමාන operand බිටු මත පමණක් වන අතර ගබඩා කළ ඉතිහාසයක් මත නොවේ. අර්ධ ආකලක සහ සම්පූර්ණ ආකලක XOR, AND, සහ OR/NAND ද්වාර වල තනිකරම සංයෝජන ජාල වන අතර එය නිවැරදි තේරීම කරයි.

**Option / විකල්පය 5)**

The phrase “retaining” explicitly signals memory. Implementations use latches or flip-flops, making it sequential rather than combinational.

“රඳවා ගැනීම” යන වාක්‍ය බණ්ඩය මතකය පැහැදිලිව සංඥා කරයි. ක්‍රියාත්මක කිරීම් අගුල් හෝ පිළි පොළ භාවිතා කරයි, එය සංයෝජන වෙනුවට අනුක්‍රමික කරයි.

**Key distinction / ප්‍රධාන වෙනස**

Combinational logic: output =  $f$  (current inputs) — no stored state.

ඒකාබද්ධ තර්කනය: ප්‍රතිදානය =  $f$  (වත්මන් ආදාන) - ගබඩා කළ තත්වයක් නොමැත.

Sequential logic: output =  $f$  (current inputs, past history) — needs storage elements.

අනුක්‍රමික තර්කනය: ප්‍රතිදානය =  $f$  (වත්මන් ආදාන, අතීත ඉතිහාසය) - ගබඩා අංග අවශ්‍ය වේ.

Only Option 4 fits the definition without involving any memory or feedback.

කිසිදු මතකයක් හෝ ප්‍රතිපෝෂණයක් සම්බන්ධ නොකර අර්ථ දැක්වීමට ගැලපෙන්නේ විකල්ප 4 පමණි.

Therefore, the answer is 4)

එබැවින් පිළිතුර 4) වේ.

4. Which of the following is an example of a single-user, multi-tasking operating system?

පහත සඳහන් ඒවායින් ඒක පරිශීලක, බහු කාර්ය මෙහෙයුම් පද්ධතියකට උදාහරණයක් වන්නේ කුමක්ද?

- 1) Windows 10      2) MS-DOS      3) IMB Mainframe      4) Linux Server      5) None of the above  
ඉහත කිසිවක් නොවේ

**Answer : 1)**

**Explanation / පැහැදිලි කිරීම**

A single-user, multi-tasking operating system allows one user to run multiple applications at the same time. Windows 10 is designed for personal computers and supports multitasking efficiently.

තනි පරිශීලක, බහු කාර්ය මෙහෙයුම් පද්ධතියක් මගින් එක් පරිශීලකයෙකුට එකවර යෙදුම් කිහිපයක් ක්‍රියාත්මක කිරීමට ඉඩ සලසයි. Windows 10 පුද්ගලික පරිගණක සඳහා නිර්මාණය කර ඇති අතර කාර්යක්ෂමව බහු කාර්යයන් සඳහා සහය දක්වයි.

**Option/ විකල්පය 2**

MS-DOS (Microsoft Disk Operating System) is a single-user, single-tasking OS. It can run only one task at a time, meaning the user must close one program before opening another.

MS-DOS (Microsoft Disk Operating System) යනු තනි පරිශීලක, තනි කාර්ය මෙහෙයුම් පද්ධතියකි. එයට එකවර ක්‍රියාත්මක කළ හැක්කේ එක් කාර්යයක් පමණි, එනම් පරිශීලකයා තවත් වැඩසටහනක් විවෘත කිරීමට පෙර වසා දැමිය යුතුය.

**Option/ විකල්පය 3**

IBM Mainframe OS is a multi-user, multi-tasking operating system designed for large-scale enterprise use. It allows multiple users to work on the system at the same time.

IBM Mainframe OS යනු මහා පරිමාණ ව්‍යවසාය භාවිතය සඳහා නිර්මාණය කර ඇති බහු-පරිශීලක, බහු කාර්ය මෙහෙයුම් පද්ධතියකි. එය බහු පරිශීලකයින්ට එකවර පද්ධතියේ වැඩ කිරීමට ඉඩ සලසයි.

**Option/ විකල්පය 4**

Linux Server OS is mainly designed for multi-user environments where multiple users can access the system and perform tasks simultaneously.

Linux Server OS ප්‍රධාන වශයෙන් නිර්මාණය කර ඇත්තේ බහු පරිශීලකයින්ට පද්ධතියට ප්‍රවේශ විය හැකි සහ එකවර කාර්යයන් ඉටු කළ හැකි බහු-පරිශීලක පරිසරයන් සඳහා ය.

**Option/ විකල්පය 5**

Since Windows 10 correctly fits the definition, this option is incorrect.

Windows 10 අර්ථ දැක්වීමට නිවැරදිව ගැලපෙන බැවින්, මෙම විකල්පය වැරදිය.

Therefore, the correct answer is option 1).

එබැවින් නිවැරදි පිළිතුර විකල්ප 1) වේ.

5. What is happening with this SQL Query?

මෙම SQL විධානය මගින් සිදුවන්නේ කුමක්ද?

`SELECT * FROM employees WHERE last_name LIKE 'S%';`

- 1) Selects all employees whose last name start with 'S'.  
last name 'S' වලින් ආරම්භ වන සියලුම සේවකයින් තෝරා ගනී.
- 2) Selects all employees whose last name contain 'S'.  
last name තුළ 'S' අඩංගු සියලුම සේවකයින් තෝරා ගනී.
- 3) Selects all employees whose last name end with 'S'.  
last name 'S' වලින් අවසන් වන සියලුම සේවකයින් තෝරා ගනී.
- 4) Selects all employees whose last name are exactly 'S'.  
last name හරියටම 'S' වන සියලුම සේවකයින් තෝරා ගනී.
- 5) None of the above  
ඉහත කිසිවක් නොවේ.

**Answer: 1)**

**Explanation / පැහැදිලි කිරීම:**

```
SELECT * FROM employees WHERE last_name LIKE 'S%';
```

Uses the LIKE operator with the pattern 'S%'. In SQL, the LIKE operator is used to search for a specified pattern in a column.

'S%' රටාව සමඟ LIKE මෙහෙයවනය භාවිතා කරයි. SQL හි, තීරුවක නිශ්චිත රටාවක් සෙවීමට LIKE මෙහෙයවනය භාවිතා කරයි.

The pattern 'S%' has a specific meaning. / 'S%' රටාවට නිශ්චිත අර්ථයක් ඇත.

- S is a literal character, indicating that the string should start with the letter 'S'.  
S යනු වචනාර්ථ අක්ෂරයකි, string එක 'S' අක්ෂරයෙන් ආරම්භ විය යුතු බව පෙන්නුම් කරයි.
- % is a wildcard character that represents zero or more characters following 'S'.
- % යනු 'S' පහත අනුලක්ෂණ ගුණය හෝ වැඩි ගණනක් නියෝජනය කරන Wildcard අක්ෂරයකි.

Therefore, 'S%' matches any string that begins with the letter 'S', followed by any number of other characters (or no characters at all).

එමනිසා, 'S%' අකුරෙන් ආරම්භ වන ඕනෑම තත්ත්වයට ගැලපේ, පසුව වෙනත් ඕනෑම අක්ෂර ගණනකින් (හෝ අක්ෂර නොමැත).

- 1) Selects all employees whose last name start with 'S'.  
අවසාන නම 'S' වලින් ආරම්භ වන සියලුම සේවකයින් තෝරා ගනී.

This is correct because the pattern 'S%' means that the last name must start with the letter 'S' and can be followed by any other characters.

මෙය නිවැරදි වන්නේ 'S%' රටාවෙන් අදහස් වන්නේ අවසාන නම 'S' අක්ෂරයෙන් ආරම්භ විය යුතු අතර වෙනත් ඕනෑම අක්ෂරයකින් අනුගමනය කළ හැකි බවයි.

- 2) Selects all employees whose last name contain 'S'.  
අවසාන නම 'S' අඩංගු සියලුම සේවකයින් තෝරා ගනී.

This would be represented by the pattern '%S%', which matches any last name containing 'S' anywhere in the string.

මෙය '%S%' රටාවෙන් නිරූපණය වනු ඇත, එය තත්ත්වයේ ඕනෑම තැනක 'S' අඩංගු ඕනෑම වාසගමකට ගැලපේ.

- 3) Selects all employees whose last name end with 'S'.  
අවසාන නම 'S' වලින් අවසන් වන සියලුම සේවකයින් තෝරා ගනී.

This would be represented by the pattern '%S', which matches any last name that ends with 'S'.

මෙය 'S' වලින් අවසන් වන ඕනෑම වාසගමකට ගැලපෙන '%S' රටාවෙන් නියෝජනය වේ.

- 4) Selects all employees whose last name are exactly 'S'.  
අවසාන නම හරියටම 'S' වන සියලුම සේවකයින් තෝරා ගනී.

This would be represented by the pattern 'S', which matches last names that are exactly 'S' without any additional characters.

මෙය 'S' රටාවෙන් නිරූපණය වන අතර එය අතිරේක අක්ෂර නොමැතිව හරියටම 'S' යන අවසාන නම් වලට ගැලපේ.

- 5) None of the above. / ඉහත කිසිවක් නොවේ.

This is incorrect because option 1) accurately describes what the query does.

- 1) විකල්පය විමසුමෙන් කරන්නේ කුමක්ද යන්න නිවැරදිව විස්තර කරන බැවින් මෙය වැරදිය.

In summary, the pattern 'S%' specifically matches last names that start with 'S', which is why option 1) is the correct answer.

සාරාංශයක් ලෙස, 'S%' රටාව 'S' වලින් ආරම්භ වන අවසාන නම් වලට විශේෂයෙන් ගැලපේ, එබැවින් විකල්ප 1) නිවැරදි පිළිතුර වේ.

6. What will the following code output.

පහත කේත ප්‍රතිදානය කුමක් වනු ඇත්ද.

```
def foo(bar=[]):
    bar.append("baz")
    return bar
print(foo())
print(foo())
```

1) ['baz'] and ['baz']

2) ['baz'] and ['baz', 'baz']

3) ['baz', 'baz'] and ['baz', 'baz']

4) [] and ['baz']

5) ['baz'] and []

**Answer : 2)**

**Explanation / පැහැදිලි කිරීම**

In Python, default argument values are evaluated only once, at the time the function is defined, not each time it is called.

Python හි, පෙරනිමි තර්ක අගයන් ඇගයීමට ලක් කරනු ලබන්නේ ශ්‍රිතය අර්ථ දැක්වන අවස්ථාවේදී පමණි, එය කැඳවන සෑම අවස්ථාවකම නොවේ.

This means that if you use a mutable default argument (like a list), that same object is reused across function calls.

මෙයින් අදහස් කරන්නේ ඔබ වෙනස් කළ හැකි පෙරනිමි තර්කයක් (ලැයිස්තුවක් වැනි) භාවිතා කරන්නේ නම්, එම වස්තුවම ශ්‍රිත ඇමතුම් හරහා නැවත භාවිතා කරන බවයි.

Step-by-step breakdown / පියවරෙන් පියවර බිඳවැටීම:

**First call / පළමු ඇමතුම: foo()**

bar is the default list []

"baz" is appended → bar = ['baz']

It returns / එය ලබා දෙයි: ['baz']

**Second call / දෙවන ඇමතුම: foo()**

Now bar is not a new list, it's the same list from the first call: ['baz']

දැන් bar නව ලැයිස්තුවක් නොවේ, එය පළමු ඇමතුමේ ඇති එකම ලැයිස්තුවයි: ['baz']

"baz" is appended again / නැවත "baz" append කරයි. → bar = ['baz', 'baz']

It returns ['baz', 'baz']

එය ['baz', 'baz'] ආපසු ලබා දෙයි

So, the output will be / එබැවින් ප්‍රතිදානය වනුයේ:

['baz']

['baz', 'baz']

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

7. What will be the output of the following Python code?

පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් වේද?

```
x = 5
for i in range(x):
    if i % 2 == 0:
        continue
    for j in range(i):
        if j == 3:
            break
    print(i, j)
```

- |                      |                             |                                    |                                           |                             |
|----------------------|-----------------------------|------------------------------------|-------------------------------------------|-----------------------------|
| 1) 1 0<br>3 0<br>3 1 | 2) 1 0<br>3 0<br>3 1<br>5 0 | 3) 1 0<br>3 0<br>3 1<br>5 0<br>5 1 | 4) 1 0<br>3 0<br>3 1<br>5 0<br>5 1<br>5 2 | 5) 1 0<br>3 0<br>3 1<br>3 2 |
|----------------------|-----------------------------|------------------------------------|-------------------------------------------|-----------------------------|

**Answer: 5)**

Step-by-Step Execution: / පියවරෙන් පියවර ක්‍රියාත්මක කිරීම:

- $i = 0$ :
  - $0 \% 2 == 0$ , so continue  $\rightarrow$  Skipped / මඟ හැරේ
- $i = 1$ :
  - $1 \% 2 != 0$ , so it proceeds. / එබැවින් එය ඉදිරියට යයි.
  - for  $j$  in range(1)  $\rightarrow j = 0$
  - $j == 3$  condition is not met, so it prints: / කොන්දේසිය සපුරා නොමැති නිසා, එය මුද්‍රණය කරයි:

Output: / ප්‍රතිදානය: 1 0

- $i = 2$ :
  - $2 \% 2 == 0$ , so continue  $\rightarrow$  Skipped / මඟ හැරේ
- $i = 3$ :
  - $3 \% 2 != 0$ , so it proceeds. / එබැවින් එය ඉදිරියට යයි.
  - for  $j$  in range(3)  $\rightarrow j = 0, 1, 2$
  - $j == 3$  is never met, so it prints: / කිසිසේත්ම නොවන බැවින්, එය මුද්‍රණය කරයි:

Output: / ප්‍රතිදානය:

```
3 0
3 1
3 2
```

- $i = 4$ :
  - $4 \% 2 == 0$ , so continue  $\rightarrow$  Skipped / මඟ හැරේ

Final Output: / අවසන් ප්‍රතිදානය:

```
1 0
3 0
3 1
3 2
```

Iteration Table: / පුනරාවර්තන වගුව:

| Iteration (i) | Condition (i % 2 == 0?) | Action          | Inner Loop (j range(i)) | Break Condition (j == 3?) | Printed Output    |
|---------------|-------------------------|-----------------|-------------------------|---------------------------|-------------------|
| 0             | 0 % 2 == 0 → True       | continue (skip) | -                       | -                         | -                 |
| 1             | 1 % 2 == 0 → False      | Enter loop      | j = 0                   | j != 3 (continue)         | 1 0               |
| 2             | 2 % 2 == 0 → True       | continue (skip) | -                       | -                         | -                 |
| 3             | 3 % 2 == 0 → False      | Enter loop      | j = 0, 1, 2             | j != 3 (continue)         | 3 0<br>3 1<br>3 2 |
| 4             | 4 % 2 == 0 → True       | continue (skip) | -                       | -                         | -                 |

8. Which of the following is a part of a closed system of the human body?

පහත සඳහන් දේවලින් මිනිස් සිරුරේ සංවෘත පද්ධතියක කොටසක් වන්නේ කුමක්ද?

- 1) Ear / කන්      2) Heart / හදවත      3) Eye / ඇස      4) Lung / පෙනහළු      5) Kidney / වකුගඩු

**Answer: 2)**

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

**Statement / ප්‍රකාශය 1)**

According to this statement, it's false. The ear is part of the sensory system and not specifically part of a closed system in the body.

මෙම ප්‍රකාශය අනුව එය අසත්‍ය වේ. කන යනු සංවේදී පද්ධතියේ කොටසක් වන අතර විශේෂයෙන් ශරීරයේ සංවෘත පද්ධතියක කොටසක් නොවේ.

**Statement / ප්‍රකාශය 2)**

According to this statement, it's true. The heart is part of the closed circulatory system, which includes the heart and blood vessels that circulate blood throughout the body.

මෙම ප්‍රකාශය අනුව එය සත්‍ය වේ. හදවත යනු සංවෘත සංසරණ පද්ධතියේ කොටසකි, ශරීරය පුරා රුධිරය සංසරණය වන හෘදය සහ රුධිර වාහිනී ඇතුළත් වේ.

**Statement / ප්‍රකාශය 3)**

According to this statement, it's false. The eye is part of the sensory system, specifically related to vision, not a closed system.

මෙම ප්‍රකාශය අනුව එය අසත්‍ය වේ. ඇස යනු සංවෘත පද්ධතියක් නොව, විශේෂයෙන් දර්ශනය හා සම්බන්ධ සංවේදී පද්ධතියේ කොටසකි.

**Statement / ප්‍රකාශය 4)**

According to this statement, it's false. The lung is part of the respiratory system and not part of a closed system; it interacts with the environment by exchanging gases.

මෙම ප්‍රකාශය අනුව එය අසත්‍ය වේ. පෙනහළු ශ්වසන පද්ධතියේ කොටසක් වන අතර සංවෘත පද්ධතියක කොටසක් නොවේ එය වායු හුවමාරුව මගින් පරිසරය සමඟ අන්තර් ක්‍රියා කරයි.

**Statement / ප්‍රකාශය 5)**

According to this statement, it's false. The kidneys are part of the urinary system and are involved in filtering blood but are not specifically part of a closed system.

මෙම ප්‍රකාශය අනුව එය අසත්‍ය වේ. වකුගඩු මුත්‍රා පද්ධතියේ කොටසක් වන අතර රුධිරය පෙරීමට සම්බන්ධ වන නමුත් විශේෂයෙන් සංවෘත පද්ධතියක කොටසක් නොවේ.

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

9. Which of the following statements about CSS specificity and inheritance is TRUE?  
CSS විශේෂත්වය සහ උරුමය පිළිබඳ පහත සඳහන් ප්‍රකාශවලින් සත්‍ය වන්නේ කුමක්ද?

- 1) The !important declaration has the lowest priority in CSS specificity.  
CSS විශේෂත්වය තුළ !important ප්‍රකාශනයට අවම ප්‍රමුඛතාවයක් ඇත.
- 2) The universal selector (\*) has higher specificity than type selectors.  
Universal තේරීම්කාරක (\*) සඳහා වර්ග තේරීම්කාරකවන්ට වඩා ඉහළ නිශ්චිතතාවයක් ඇත.
- 3) ID selectors override class selectors regardless of importance or inline styles.  
වැදගත්කම හෝ පේළිගත විලාස නොසලකා ID තේරීම්කාරක පන්ති තේරීම්කාරක අභිබවා යයි.
- 4) Inherited properties always override styles explicitly applied to a child element.  
උරුම වූ ගුණාංග සෑම විටම child මූලාංගයකට පැහැදිලිව යොදන ලද විලාසයන් අභිබවා යයි.
- 5) Inline styles override both ID and class selectors in CSS specificity.  
CSS විශේෂත්වය තුළ පේළිගත විලාසයන් මගින් ID සහ පන්ති තේරීම්කාරක දෙකම අභිබවා යයි.

**Answer: 5)**

Let's consider each option one by one.

එක් එක් විකල්පය එකින් එක සලකා බලමු.

#### Option/ විකල්ප 1)

This is false. The !important declaration gives the highest priority to a rule, overriding inline styles, ID selectors, and everything else unless another rule also has !important. So, this option is incorrect because it states the exact opposite.

මෙය අසත්‍යයි. !important ප්‍රකාශනය රීතියකට ඉහළම ප්‍රමුඛතාවය ලබා දෙන අතර, තවත් රීතියකට !important තිබේ නම් මිස, inline styles, ID selectors සහ අනෙකුත් සියල්ල අභිබවා යයි. එබැවින්, මෙම විකල්පය වැරදියි, මන්ද එය හරියටම ප්‍රතිවිරුද්ධ දේ ප්‍රකාශ කරයි.

#### Option/ විකල්ප 2)

This is also false. The universal selector \* has the lowest specificity of all. Type selectors (like div, p, h1) have a higher specificity than \*.

මෙය ද අසත්‍යයි. Universal වරකය \* සියල්ලටම වඩා අඩුම නිශ්චිතතාව ඇත. වර්ග තේරීම් (div, p, h1 වැනි) \* ට වඩා ඉහළ නිශ්චිතතාවයක් ඇත.

#### Option/ විකල්ප 3)

This is not always true. While ID selectors have higher specificity than class selectors, inline styles actually have even more specificity than ID selectors. Also, any rule with !important will override ID selectors too. So this statement is misleading.

මෙය සැමවිටම සත්‍ය නොවේ. ID තේරීම්කාරකවන්ට පන්ති තේරීම් වලට වඩා ඉහළ නිශ්චිතතාවයක් ඇති අතර, inline styles ඇත්ත වශයෙන්ම ID තේරීම් වලට වඩා වැඩි නිශ්චිතතාවයක් ඇත. එසේම, !important සහිත ඕනෑම රීතියක් ID තේරීම් ද අභිබවා යනු ඇත. එබැවින් මෙම ප්‍රකාශය නොමඟ යවන සුළුය.

#### Option/ විකල්ප 4)

In CSS, inheritance only applies when no specific style is directly applied to the child element. If a child element has a style defined directly (via selector, inline style, etc.), that explicit style will take precedence over any inherited value from the parent.

CSS හි, උරුමය අදාළ වන්නේ ළමා මූලාංගයට නිශ්චිත ශෛලියක් සෘජුවම යොදන්නේ නැති විට පමණි. ළමා මූලාංගයකට සෘජුවම අර්ථ දක්වා ඇති ශෛලියක් තිබේ නම් (තේරීම්කාරකය, පේළි විලාසය, ආදිය හරහා), එම පැහැදිලි විලාසය මාපියන්ගෙන් උරුම වූ ඕනෑම අගයකට වඩා ප්‍රමුඛත්වය ගනී.

#### Option/ විකල්ප 5)

This is the correct answer. Inline styles (e.g., style="color: red;") have higher specificity than both ID (#id) and class (.class) selectors, and are only overridden by !important rules. This is a fundamental concept in CSS.

මෙය නිවැරදි පිළිතුරයි. පේළිගත විලාස (උදා: style="color: red;) ID (#id) සහ පන්ති (.class) තේරීම් දෙකටම වඩා ඉහළ නිශ්චිතතාවයක් ඇති අතර, !important නීති මගින් පමණක් අභිබවා යයි. මෙය CSS හි මූලික සංකල්පයකි.

10. Which of the following code generates a hyperlink that directs to a different section of the same web page?  
එකම වෙබ් පිටුවේ වෙනත් කොටසකට යොමු කරන අධි සබැඳියක් උත්පාදනය කරන්නේ පහත සඳහන් කුමන කේතයද?

- 1) `<a href="contact-us">Contact Us</a>`
- 2) `<a href=".contact-us">Contact Us</a>`
- 3) `<a href="#contact-us">Contact Us</a>`
- 4) `<a href="_contact-us">Contact Us</a>`
- 5) `<a href="http://contact-us">Contact Us</a>`

**Answer: 3)**

#### **Explanation / පැහැදිලි කිරීම**

To link to a different section of the same web page, you use an HTML anchor (#) followed by the ID of the element you want to jump to.

එකම වෙබ් පිටුවේ වෙනත් කොටසකට සම්බන්ධ කිරීමට, ඔබ HTML anchor (#) භාවිතා කර ඔබට යාමට අවශ්‍ය මූලාංගයේ ID භාවිතා කරයි.

Example:

Suppose somewhere else in the HTML you have:

ඔබ සතුව ඇති HTML හි වෙනත් ස්ථානයක ඇතැයි සිතන්න

```
<div id="contact-us">
  <!-- Contact section content -->
</div>
```

```
<a href="#contact-us">Contact Us</a>
```

This will scroll to (or jump to) that <div> with the ID "contact-us" on the same page.

එම පිටුවෙහිම ID "contact-us" සමඟ <div> වෙත අනුවලනය කරනු ඇත.

#### **Option / විකල්පය 1)**

This tries to go to a different webpage named contact-us, not a section of the same page.

මෙය එකම පිටුවේ කොටසකට නොව, contact-us නම් වෙනත් වෙබ් පිටුවකට යාමට උත්සාහ කරයි.

#### **Option / විකල්පය 2)**

This assumes .contact-us is a class selector, but HTML anchor links don't use class names — they use IDs with #.

මෙය .contact-us class තේරීමක් ලෙස උපකල්පනය කරයි, නමුත් HTML anchor සබැඳි class නම් භාවිතා නොකරයි - ඒවා # සහිත ID භාවිතා කරයි.

#### **Option / විකල්පය 3)**

This is the correct option / මෙය නිවැරදි විකල්පයයි.

#### **Option / විකල්පය 4)**

Tries to go to a file or path called \_contact-us, not a section within the same page.

එකම පිටුව තුළ ඇති කොටසකට නොව \_contact-us නම් ගොනුවකට හෝ මාර්ගයකට යාමට උත්සාහ කරයි.

#### **Option / විකල්පය 5)**

This is a full URL to an external site or path http://contact-us, not internal navigation.

මෙය බාහිර වෙබ් අඩවියකට හෝ http://contact-us මාර්ගයකට යන සම්පූර්ණ URL එකකි, අභ්‍යන්තර සංචාලනය නොවේ.

Therefore, the answer is 3).

එබැවින් පිළිතුර 3) වේ.