

Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 525 MARKING

2026/05/19
ictfromabc
Ravindu Bandaranayake

1. Which of the following is a characteristic of a GPU compared to a CPU?

CPU එකකට කාපේක්ෂව GPU එකක ලක්ෂණයක් වන්නේ පහත සඳහන් කුමක් ද?

- 1) Lower number of cores optimized for sequential processing
අනුක්‍රමික සැකසුම් සඳහා ප්‍රශස්ත කර ඇති අඩු හර (cores) සංඛ්‍යාව
- 2) Higher number of cores optimized for parallel processing
සමාන්තර සැකසුම් සඳහා ප්‍රශස්ත කර ඇති හර (cores) ගණන වැඩි වීම
- 3) Lower memory bandwidth / අඩු මතක කලාප පළල
- 4) Greater clock speed for single-threaded tasks / single-threaded කාර්යයන් සඳහා වැඩි ස්පන්ධන වේගය
- 5) Larger cache memory / විශාල වාරක මතකය

Answer : 2)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්පය 1)

Incorrect. GPUs typically have many smaller cores, and these cores are optimized for parallel processing tasks (e.g., handling multiple operations at once). In contrast, CPUs have fewer cores that are optimized for sequential tasks (one task after the other).

වැරදියි. GPU වලට සාමාන්‍යයෙන් කුඩා හරයන් බොහොමයක් ඇති අතර, මෙම හර සමාන්තර සැකසුම් කාර්යයන් සඳහා ප්‍රශස්ත කර ඇත (උදා: එකවර මෙහෙයුම් කිහිපයක් හැසිරවීම). ඊට ප්‍රතිවිරුද්ධව, CPU වලට අනුක්‍රමික කාර්යයන් සඳහා ප්‍රශස්ත කරන ලද හරය අඩුයි (එක් කාර්යයක් පසු එකක්).

Option / විකල්පය 2)

Correct. GPUs are designed with a large number of smaller cores to perform many tasks at the same time. This architecture is ideal for applications like video rendering, machine learning, and scientific simulations, where a large amount of data can be processed in parallel. GPUs are capable of handling thousands of threads concurrently, making them highly efficient for tasks that can be divided into many smaller tasks.

නිවැරදියි. GPU නිර්මාණය කර ඇත්තේ කුඩා හර විශාල සංඛ්‍යාවක් සමඟින් බොහෝ කාර්යයන් එකවර ඉටු කිරීමටය. මෙම නිර්මාණ ශිල්පය විධියේ විද්‍යුත්කරණය, යන්ත්‍ර ඉගෙනීම සහ විද්‍යාත්මක අනුප්‍රාප්තික වැනි යෙදුම් සඳහා වඩාත් සුදුසු වන අතර එහිදී විශාල දත්ත ප්‍රමාණයක් සමාන්තරව සැකසිය හැක. GPU වලට threads දහස් ගණනක් සමගාමීව හැසිරවීමේ හැකියාව ඇති අතර, ඒවා බොහෝ කුඩා කාර්යයන් වලට බෙදිය හැකි කාර්යයන් සඳහා ඉතා කාර්යක්ෂම වේ.

Option / විකල්පය 3)

Incorrect. GPUs typically have higher memory bandwidth compared to CPUs. This is because GPUs handle large volumes of data, especially in tasks like image processing, where high bandwidth is crucial to move data quickly between the processor and memory.

වැරදියි. GPU වලට සාමාන්‍යයෙන් CPU වලට කාපේක්ෂව ඉහළ මතක කලාප පළලක් ඇත. මක්නිසාද යත් GPU විශාල දත්ත පරිමාවක් හසුරුවන බැවිනි, විශේෂයෙන් රූප සැකසීම වැනි කාර්යයන් වලදී, සකසනය සහ මතකය අතර දත්ත ඉක්මනින් ගෙනයාමට ඉහළ කලාප පළලක් ඉතා වැදගත් වේ.

Option / විකල්පය 4)

Incorrect. While GPUs have high clock speeds, CPUs are usually optimized for better single-threaded performance (processing tasks one by one). A CPU typically has a higher clock speed in comparison, which helps it handle tasks that require more focused processing.

වැරදියි. GPU වලට ඉහළ ස්පන්ධන වේගය ඇති අතර, CPU සාමාන්‍යයෙන් වඩා හොඳ single-threaded කාර්ය සාධනයක් සඳහා ප්‍රශස්ත කර ඇත (කාර්යයන් එකින් එක සැකසීම). CPU එකකට සාමාන්‍යයෙන් කාපේක්ෂව වැඩි ස්පන්ධන වේගයක් ඇත, එය වඩාත් අවධානය යොමු කළ සැකසුම් අවශ්‍ය වන කාර්යයන් හැසිරවීමට උපකාරී වේ.

Option / විකල්පය 5)

Incorrect. CPUs typically have larger cache memory compared to GPUs. Cache memory is important for reducing the time needed to access frequently used data. Since GPUs focus on parallel processing and high throughput for massive data sets, they typically do not need as much cache memory as CPUs, which are used for tasks requiring fast access to smaller datasets.

වැරදියි. CPU වලට සාමාන්‍යයෙන් GPU වලට සාපේක්ෂව විශාල වාරක මතකයක් ඇත. නිතර භාවිතා කරන දත්ත වෙත ප්‍රවේශ වීමට ගතවන කාලය අඩු කිරීම සඳහා වාරක මතකය වැදගත් වේ. GPUs විශාල දත්ත කට්ටල සඳහා සමාන්තර සැකසුම් සහ ඉහළ ප්‍රතිදානය කෙරෙහි අවධානය යොමු කරන බැවින්, ඒවාට සාමාන්‍යයෙන් CPU තරම් වාරක මතකයක් අවශ්‍ය නොවේ, ඒවා කුඩා දත්ත කට්ටලවලට වේගවත් ප්‍රවේශයක් අවශ්‍ය කාර්යයන් සඳහා භාවිතා කරයි.

Therefore, the correct answer is 2).

එබැවින්, නිවැරදි පිළිතුර 2) වේ.

2. The time on a 12-hour clock is represented in binary as 00001100 : 01010100. What time is displayed on the clock (Hours : Minutes)?

පැය 12ක් පෙන්වන ඔරලෝසුවක වේලාව ද්විමය නිරූපණයෙන් 00001100 : 01010100 වේ. ඔරලෝසුවේ දිස්වන වේලාව කුමක්ද (පැය : විනාඩි)?

- 1) 13 : 24 2) 12 : 24 3) 12 : 84 4) 01 : 24 5) 01 : 24

Answer: 4)

Explanation / පැහැදිලි කිරීම:

First, let's convert these numbers to decimal. / පළමුව මෙම සංඛ්‍යා දශමය බවට පරිවර්තනය කරමු.

Minutes: 01010100 ₂							
0	1	0	1	0	1	0	0
2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
-	64×1	-	16×1	-	4×1	-	-
64 + 16 + 4							
84 ₁₀							

84 minutes does not appear on a clock. / ඔරලෝසුවක මිනිත්තු 84 දිස් නොවේ.

Therefore / එමනිසා,

$$84 - 60 = 24$$

The minute section displays 24. Then the 60 minutes subtracted go into the hour division as 1 hour.

මිනිත්තු කොටසේ දිස් වන්නේ 24 වේ. එවිට අඩු කරන මිනිත්තු 60, පැය 1ක් වශයෙන් පැය කොටසට ගමන් කරයි.

Hours: 00001100 ₂							
0	0	0	0	1	1	0	0
2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
-	-	-	-	8×1	4×1	-	-
8 + 4							
12 ₁₀							

Adding 1 hour to the minute part gives 13 hours. This is a 12-hour clock.

මිනිත්තු කොටසෙන් පැමිණි පැය 1ද සමග එකතු වීමෙන් පැය 13ක් වේ. මෙය පැය 12 ඔරලෝසුවකි.

Therefore / එමනිසා,

$$13 - 12 = 01$$

Then, the hour part shows 01. / එවිට, පැය කොටසේ දිස් වන්නේ 01 වේ.

Therefore, the time displayed on the clock is 01:24. / එමනිසා, ඔරලෝසුවේ දිස්වන වේලාව වන්නේ 01:24 වේ.

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

3. What is the final output of the given Boolean expression?
දී ඇති බූලියානු ප්‍රකාශනයේ අවසාන ප්‍රතිදානය කුමක් ද?

$$D. (C.B + 0.B) + B.D$$

- 1) D.B.C 2) 1 3) 0 4) B.C+D 5) B.D

Answer: 5)

Let's simplify the given Boolean expression.
ලබා දී ඇති බූලියානු ප්‍රකාශනය සරල කරමු.

$$\begin{aligned} &= D.(C.B+0.B)+B.D \\ &= D.(C.B+0)+B.D && \text{Identity Law / සර්වකාමය න්‍යාය} \\ &= D.C.B+B.D && \text{Distributive Law / විකටන න්‍යාය} \\ &= B.D && \text{Identity Law / සර්වකාමය න්‍යාය} \end{aligned}$$

Therefore, the correct answer is 5).
එමනිසා, නිවැරදි පිළිතුර 5) වේ.

4. Which memory management technique divides memory into fixed-size blocks?
මතකය ස්ථාවර ප්‍රමාණයේ කොටස් වලට බෙදන මතක කළමනාකරණ තාක්ෂණය කුමක්ද?

- 1) Paging / පිටුකරණය
2) Segmentation / ඛණ්ඩනීකරණය
3) Contiguous allocation / යාබද විභාජනය
4) Swapping / ප්‍රතිහරනය
5) Dynamic partitioning / ගතික කොටස් කිරීම

Answer : 1)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)

Paging divides memory into fixed-size blocks called pages and maps them to frames in physical memory. පිටුකරණය මඟින් මතකය පිටු ලෙස හඳුන්වන ස්ථාවර ප්‍රමාණයේ කොටස් වලට බෙදා භෞතික මතකයේ රාමු වලට සිතියම්ගත කරයි.

Option / විකල්ප 2)

Segmentation divides memory into variable-sized segments. ඛණ්ඩනීකරණය මඟින් මතකය විචල්‍ය ප්‍රමාණයේ කොටස් වලට බෙදයි.

Option / විකල්ප 3)

Contiguous allocation requires adjacent memory blocks, unlike paging. පිටුකරණය මෙන් නොව, යාබද විභාජනය සඳහා යාබද මතක කොටස් අවශ්‍ය වේ.

Option / විකල්ප 4)

Swapping moves entire processes between RAM and disk, not fixed blocks. ප්‍රතිහරනය මඟින් සම්පූර්ණ ක්‍රියාවලීන් RAM සහ තැටිය අතර ගෙන යයි, ස්ථාවර ඩිලොක් අතර නොවේ.

Option / විකල්ප 5)

Dynamic partitioning divides memory into variable-sized partitions instead of fixed-size blocks, leading to external fragmentation. ගතික කොටස් කිරීම මඟින් මතකය ස්ථාවර ප්‍රමාණයේ කොටස් වෙනුවට විචල්‍ය ප්‍රමාණයේ කොටස් වලට බෙදන අතර එමඟින් බාහිර ඛණ්ඩනයට මග පාදයි.

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

5. What is the purpose of the INNER JOIN clause in SQL?
SQL හි INNER JOIN වගන්තියේ අරමුණ කුමක්ද?

- 1) To join all rows from both tables / වගු දෙකෙහිම සියලුම පේළි එක් කිරීමට
- 2) To join only unique rows from both tables / වගු දෙකෙහිම අනන්‍ය පේළි පමණක් සම්බන්ධ වීමට
- 3) To join only matching rows from both tables / වගු දෙකෙහිම ගැලපෙන පේළි පමණක් සම්බන්ධ වීමට
- 4) To join all rows from one table and only matching rows from the other table
එක් වගුවකින් සියලුම පේළි හා අනෙක් වගුවෙන් ගැලපෙන පේළි පමණක් සම්බන්ධ කිරීමට
- 5) To join rows and remove duplicates / පේළි එක් කිරීමට සහ අනුපිටපත් ඉවත් කිරීමට

Answer: 3)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)

This would describe a FULL OUTER JOIN, which includes all rows from both tables, even if there is no match.

ගැලපීමක් නොමැතිවුවද වගු දෙකෙහිම සියලුම පේළි ඇතුළත් වන FULL OUTER JOIN මෙය විස්තර කරයි.

Option / විකල්ප 2)

This would be closer to the behavior of DISTINCT (removing duplicates), but INNER JOIN does not remove duplicates inherently.

මෙය DISTINCT (අනුපිටපත් ඉවත් කිරීම) හි හැසිරීමට සමීප වනු ඇත, නමුත් INNER JOIN අනුපිටපත් සහජයෙන්ම ඉවත් නොකරයි.

Option / විකල්ප 3)

This describes an INNER JOIN. An INNER JOIN only returns rows that have matching values in both tables based on the condition specified in the ON clause. If there's no match, those rows are excluded from the result set.

මෙය INNER JOIN එකක් විස්තර කරයි. INNER JOIN එකක් ලබා දෙන්නේ ON වගන්තියේ දක්වා ඇති කොන්දේසිය මත පදනම්ව වගු දෙකෙහිම ගැලපෙන අගයන් ඇති පේළි පමණි. ගැලපීමක් නොමැති නම්, එම පේළි ප්‍රතිඵල කට්ටලයෙන් බැහැර කරනු ලැබේ.

Option / විකල්ප 4)

This could describe a LEFT JOIN (or RIGHT JOIN), where all rows from one table are included, and only matching rows from the other.

මෙය LEFT JOIN (හෝ RIGHT JOIN) විස්තර කළ හැක, එහිදී එක් වගුවක සියලුම පේළි ඇතුළත් වන අතර අනෙක් වගුවේ සිට ගැලපෙන පේළි පමණි.

Option / විකල්ප 5)

This is not related to INNER JOIN, as it talks about removing duplicates, which is done with the DISTINCT keyword, not by joining tables.

මෙය INNER JOIN හා සම්බන්ධ නොවේ, මන්ද එහි අනුපිටපත් ඉවත් කිරීම ගැන කතා කරයි, එය DISTINCT මූල පදයෙන් සිදු කරනු ලැබේ, වගු එකතු කිරීමෙන් නොවේ.

So, the correct answer number is 3).

එබැවින්, නිවැරදි පිළිතුරු අංකය 3) වේ.

6. What does the mode 'r+' do when opening a file?

ගොනුවක් විවෘත කිරීමේදී 'r+' මගින් කරන්නේ කුමක්ද?

- 1) Opens a file for reading only.
කියවීම සඳහා පමණක් ගොනුවක් විවෘත කරයි.
- 2) Opens a file for writing only
ගොනුවක් ලිවීම සඳහා පමණක් විවෘත කරයි,
- 3) Opens a file for writing only, creating a new file if it doesn't exist.
ලිවීම සඳහා පමණක් ගොනුවක් විවෘත කරයි, එය නොමැති නම් නව ගොනුවක් නිර්මාණය කරයි.
- 4) Opens a file for reading and writing, starting at the beginning of the file.
ගොනුවේ ආරම්භයේ සිට කියවීම සහ ලිවීම සඳහා ගොනුවක් විවෘත කරයි.
- 5) Opens a file for reading only, creating a new file if it doesn't exist
කියවීම සඳහා පමණක් ගොනුවක් විවෘත කරයි, එය නොමැති නම් නව ගොනුවක් නිර්මාණය කරයි.

Answer : 4)

Explanation / පැහැදිලි කිරීම:

The mode r+ in file handling opens a file for both reading and writing, starting at the beginning of the file. It does not create a new file if it doesn't exist, unlike some other modes. This means you can read from and write to the file without truncating its contents.

ගොනු හැසිරවීමේදී r+ මාදිලිය ගොනුවේ ආරම්භයේ සිට කියවීම සහ ලිවීම යන දෙකටම ගොනුවක් විවෘත කරයි. එය වෙනත් සමහර මාතයන් මෙන් නොව එය නොපවතියි නම් එය නව ගොනුවක් නිර්මාණය නොකරයි. මෙයින් අදහස් කරන්නේ ඔබට එහි අන්තර්ගතය කප්පාදු නොකර ගොනුවෙන් කියවීමට සහ ලිවීමට හැකි බවයි.

Therefore, the answer is, 4).

එබැවින් පිළිතුර වන්නේ, 4) වේ

7. `dictionary = {'a': 1, 'b': 2, 'c': 3}`
`x = dictionary.keys()`
`print(x)`

What is the output of the above Python code?

ඉහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් වේද?

- 1) {'a', 'b', 'c'}
- 2) dict keys(['a', 'b', 'c'])
- 3) {'a': 1, 'b': 2, 'c': 3}
- 4) {1,2,3}
- 5) syntax error

Answer: 2)

Let's consider the process of this Python code step by step.

මෙම පයිතන් කේතයේ ක්‍රියාවලිය පියවරෙන් පියවර සලකා බලමු.

dictionary = {'a': 1, 'b': 2, 'c': 3}

This line creates a dictionary named 'dictionary' with three key-value pairs:

මෙම පේළිය යතුරු-අගය යුගල තුනක් සහිත 'dictionary' නම් dictionary එකක් නිර්මාණය කරයි:

'a' is the key with the value 1.

'a' යනු 1 අගය සහිත යතුරයි.

'b' is the key with the value 2.

'b' යනු 2 අගය සහිත යතුරයි.

'c' is the key with the value 3.

'c' යනු 3 අගය සහිත යතුරයි.

x = dictionary.keys()

This line calls the keys() method on the dictionary object. The keys() method returns a view object that displays a list of all the keys in the dictionary. This view object is then assigned to the variable x.

මෙම ජේලිය dictionary හි keys() ක්‍රමය ලෙස හඳුන්වයි. keys() ක්‍රමය dictionary එකෙහි ඇති සියලුම යතුරු ලැයිස්තුවක් පෙන්වන view object එකක් ලබා දෙයි. මෙම view object එක පසුව x විචල්‍යයට පවරනු ලැබේ.

print(x)

This line prints the value of x to the console. Since x is a dictionary view object containing the keys, the output will be:

මෙම ජේලිය කොන්සෝලය මත x අගය මුද්‍රණය කරයි. x යනු යතුරු අඩංගු dictionary එකෙහි view object එකක් බැවින්, ප්‍රතිදානය වනුයේ:

```
dict_keys(['a', 'b', 'c'])
```

So, the correct answer is 2).

එබැවින්, නිවැරදි පිළිතුර 2) වේ.

8. What is represented by data flows in a data flow diagram?

දත්ත ගැලීම් සටහනක දත්ත ප්‍රවාහ මගින් නිරූපණය කරනු ලබන්නේ කුමක්ද?

- 1) Just a stream of data / දත්ත ගලනයක් පමණි.
- 2) Data flows and their directions / දත්ත ගලනයක් හා ඒවායේ දිශාවන්ය.
- 3) Only the data going to the database / දත්ත ගබඩාවට යන දත්ත පමණි.
- 4) Only data coming from an external entity / බාහිර භූතාථියක් මගින් එන දත්ත පමණි.
- 5) Only the direction of data flow / දත්ත ගලන දිශාව පමණි.

Answer : 2)

The correct answer is 4) Data flows and their directions.

නිවැරදි පිළිතුර නම් 4) දත්ත ගලන හා ඒවායේ දිශාවන්ය.

In a Data Flow Diagram (DFD), data flows represent the movement of data between different components of a system, including processes, data stores, and external entities. The direction of the flow is important because it shows where the data is coming from and where it is going, helping to visualize how data moves through the system.

දත්ත ප්‍රවාහ රූප සටහනක (DFD) දත්ත ප්‍රවාහයන් මගින් ක්‍රියාවලි, දත්ත ගබඩා සහ බාහිර ආයතන ඇතුළුව පද්ධතියක විවිධ සංරචක අතර දත්ත වලනය නියෝජනය කරයි. ප්‍රවාහයේ දිශාව වැදගත් වන්නේ එය දත්ත පැමිණෙන්නේ කොතැනින්ද සහ එය යන්නේ කොතැනටද යන්නත්, පද්ධතිය හරහා දත්ත වලනය වන ආකාරය දැශයමාන කිරීමට උපකාරී වන නිසාත් ය.

Why Other Options Are Incorrect / වෙනත් විකල්ප වැරදි ඇයි,

Option / විකල්පය 1

False, while data flows represent data, they also include the direction of movement, not just the data itself. අසත්‍ය වේ, දත්ත ප්‍රවාහයන් දත්ත නියෝජනය කරන අතරම, ඒවාට දත්ත පමණක් නොව වලනය වන දිශාවද ඇතුළත් වේ.

Option / විකල්පය 3

False, data flows represent all data movements, not just those going to a database. They can represent data moving between any part of the system, such as between processes or external entities.

අසත්‍ය වේ, දත්ත ප්‍රවාහයන් දත්ත සමුදායකට යන ඒවා පමණක් නොව සියලුම දත්ත චලනයන් නියෝජනය කරයි. ක්‍රියාවලි හෝ බාහිර භූතාර්ථ අතර වැනි පද්ධතියේ ඕනෑම කොටසක් අතර චලනය වන දත්ත ඒවාට නියෝජනය කළ හැකිය.

Option / විකල්පය 4

False, data flows show movement between all parts of the system, including internal processes and data stores, not just from external entities.

අසත්‍ය වේ, දත්ත ප්‍රවාහයන් බාහිර භූතාර්ථවලින් පමණක් නොව අභ්‍යන්තර ක්‍රියාවලි සහ දත්ත ගබඩා ඇතුළුව පද්ධතියේ සියලුම කොටස් අතර චලනය පෙන්වයි.

Option / විකල්පය 5

False, data flows represent both the data itself and the direction it travels in, not just the direction.

අසත්‍ය වේ, දත්ත ප්‍රවාහයන් දත්තම සහ දිශාව පමණක් නොව එය ගමන් කරන දිශාව යන දෙකම නියෝජනය කරයි.

So, the correct representation in a DFD includes both the data and its direction.

එබැවින්, DFD හි නිවැරදි නිරූපණයට දත්ත සහ එහි දිශාව යන දෙකම ඇතුළත් වේ.

9. What is the result of this expression?

මෙම ප්‍රකාශනයේ ප්‍රතිඵලය කුමක්ද?

```
$arr = ['a' => 1, 'b' => 2, 'c' => 3];
echo array_values($arr)[1];
```

- 1) a 2) b 3) 1 4) 2 5) Error

Answer : 4)

Explanation / පැහැදිලි කිරීම

Step 1: \$arr = ['a' => 1, 'b' => 2, 'c' => 3]

This is an associative array

මෙය associative array එකක් වේ.

This means / මෙයින් අදහස් වන්නේ

\$arr['a'] is 1

\$arr['b'] is 2

\$arr['c'] is 3

Step 2: array_values(\$arr)

The function array_values() removes the keys and returns just the values in the order they appear
array_values() ශ්‍රිතය යතුරු ඉවත් කර ඒවා දිස්වන අනුපිළිවෙලට අගයන් පමණක් ආපසු ලබා දෙයි

array_values(\$arr) returns: [1, 2, 3]

Now, the resulting array is indexed numerically

දැන්, ප්‍රතිඵලයක් ලෙස ලැබෙන අරාට සංඛ්‍යාත්මකව අනුක්‍රමණය කර ඇත.

index 0 → 1

index 1 → 2

index 2 → 3

Step 3: (\$arr)[1]

The [1] accesses the second element (index 1) in the array, which is 2.

[1] , අරාටෙහි දෙවන මූලද්‍රව්‍යය(index 1) වෙත ප්‍රවේශ වන අතර එය 2 වේ.

Final Output / අවසාන ප්‍රතිදානය: 2

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

10. What is the correct code for applying a background image using CSS?

CSS මගින් පසුබිම් රූපයක් යෙදීමට අදාළ නිවැරදි කේතය කුමක්ද?

- 1) `<style type="text/css">body{bgimage:url(ictfromabc.jpg);}</style>`
- 2) `<style>body body{backgroundimage: url(ictfromabc.jpg);}</style>`
- 3) `<style type="text/css">body{background-image:url(ictfromabc.jpg);}</style>`
- 4) `<style type="text/css"><body background image:url(ictfromabc.jpg)></style>`
- 5) `<style type="text/css"><body bgimage: url(ictfromabc.jpg)></style>`

Answer : 3)

The background-image property sets one or more background images for an element. By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.

background-image මගින් එක් මූලාංගයක් සඳහා පසුබිම් රූප එකක් හෝ කිහිපයක් සකසයි. සම්මත ආකාරයට අනුව, background-image මූලාංගයක ඉහළ වම් කෙළවරේ සිට සිරස් අතට සහ තිරස් අතට පුනර්කරණය වේ.

The CSS code related to it can be written as follows.

එයට අදාළ CSS කේතය පහත ආකාරයට ලියා දැක්විය හැකිය.

```
<style type="text/css">
    body{background-image:url(filepath\filename.extension);}
</style>
```

Accordingly, the correct answer can be taken as number 3.

ඒ අනුව නිවැරදි පිළිතුර අංක 3 ලෙස ගත හැකිය.

1.(a) Consider the following Python code

පහත පයිතන් කේතය සලකා බලන්න

```
num_students = int(input("Enter the number of students: "))
num_subjects = int(input("Enter the number of subjects per student: "))

for i in range(1, num_students + 1):
    print(f"\nStudent {i}:")
    total_marks = 0

    for j in range(1, num_subjects + 1):
        marks = int(input(f" Enter marks for Subject {j}: "))
        total_marks += marks

    average = total_marks / num_subjects

    if average >= 90:
        grade = 'A'
    elif average >= 75:
        grade = 'B'
    elif average >= 50:
        grade = 'C'
    else:
        grade = 'F'

    if grade == 'F':
        remark = "Failed. Needs improvement."
    else:
        remark = "Passed."

    print(f"Average Marks: {average:.2f}")
    print(f"Grade: {grade}")
    print(f"Remark: {remark}")
```

- i. What will be if a student has marks 85, 70, and 90 in three subjects, what will be the average, grade, and remark printed by the program?

ගිණයයෙකු විෂයන් තුනක් සඳහා ලකුණු 85, 70, සහ 90 ලබාගෙන තිබේ නම්, වැඩසටහන මගින් මුද්‍රණය කරනු ලබන සාමාන්‍යය (average) ශ්‍රේණිය (grade) සහ සටහන (remark) කුමක්ද?

Average Marks: 81.67

Grade: B

Remark: Passed.

- ii. Modify the program so that 5 extra marks are added to each subject before calculating the average. Show the updated Python code.

සාමාන්‍යය (average) ගණනය කිරීමට පෙර සෑම විෂයකටම අමතර ලකුණු 5 බැගින් එකතු වන සේ වැඩසටහන වෙනස් කරන්න. යාවත්කාලීන කරන ලද Python කේතය පෙන්වන්න.

```
marks = int(input(f" Enter marks for Subject {j}: "))
marks += 5 # Add 5 extra marks before adding to total
total_marks += marks
```

iii. In the statement: `print(f"Average Marks: {average:.2f}")`

Explain the purpose of `.2f` in the output.

`print(f"Average Marks: {average:.2f}")` යන ප්‍රකාශයෙහි ප්‍රතිදානය තුළ `.2f` මගින් සිදුවන කාර්යය පැහැදිලි කරන්න.

In the statement `print(f"Average Marks: {average:.2f}")`, the `.2f` is a formatting instruction used inside an f-string. The colon `:` introduces the format specification, and `.2f` means the value should be displayed as a floating-point number rounded to two decimal places. This ensures the average is printed neatly, for example, if the value is 81.6666, it will appear as 81.67 in the output.

`print(f"Average Marks: {average:.2f}")` යන ප්‍රකාශයෙහි, `.2f` යනු (f-string) එකක් තුළ භාවිතා වන හැඩසවි ගැන්වීමේ උපදෙසකි (formatting instruction) මෙහි ඇති තිත් දෙක (`:`) මගින් හැඩසවි පිරිවිතර (format specification) හඳුන්වා දෙන අතර, `.2f` යන්නෙන් අදහස් වන්නේ අදාළ අගය දශමස්ථාන දෙකකට වටයන ලද ඉපිලෙන ලක්ෂ්‍ය සංඛ්‍යාවක් (floating-point number) ලෙස ප්‍රදර්ශනය කළ යුතු බවයි. මෙය සාමාන්‍ය අගය ඉතා පිළිවෙළට මුද්‍රණය කිරීම සහතික කරයි. උදාහරණයක් ලෙස, අගය 81.6666 නම්, එය ප්‍රතිදානය තුළ පෙන්වනු ලබන්නේ 81.67 ලෙසය.