සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved]

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 506 MARKING

2026/04/30
ictfromabc
Ravindu Bandaranayake

1. Which of the following is a characteristic of a GPU compared to a CPU?
   CPU එකකට සාපේක්ෂව 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) <u>Higher number of cores optimized for parallel processing</u>
      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 ස්පන්ධ 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 වීඩියෝ විදැහුම්කරණය, යන්ත්‍ර ඉගෙනීම සහ විද්‍යාත්මක අනුහුරුකරණ 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 සාපේක්ෂව ඉහල 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 සාපේක්ෂව 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. What is the decimal equivalent to the binary 101101.001_2
ද්විමය 101101.001_2 ට සමාන දශමය සංඛ්‍යාව කුමක්ද?

1) 45.5 2) 48.125 3) 65.5 4) <u>45.125</u> 5) 50.25

Answer: 4)

Explanation / පැහැදිලි කිරීම:

Let's convert the integer part into decimal first.
පළමුව නිඛිල කොටස දශම බවට පරිවර්තනය කරමු,

Conversion of binary to decimal ද්විමය, දශමයට පරිවර්තනය												
1101011.1011_2												
0	0	1	0	1	1	0	1		0	0	1	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}
128	64	32	16	8	4	2	1		0.5	0.25	0.125	0.0625
32+8+4+1 = 45									0.125			
45.125_{10}												

So, the decimal equivalent of the binary 101101.001_2 is 45.125_{10}.
එබැවින්, ද්විමය 101101.001_2 හි දශම සමානය 45.125_{10} වේ.

Therefore, the answer is the option 4).
එබැවින්, පිළිතුර විකල්පය 4) වේ.

3. Simplify / සරල කරන්න.

   X = AB + AB + AB

   1) A 2) B 3) 1 4) A+B 5) AB

   Answer: 5)

   Explanation / පැහැදිලි කිරීම:

   The expression X = AB + AB + AB can be simplified using the idempotent law of Boolean algebra, which states that A+A=A.
   X = AB + AB + AB ප්‍රකාශනය බූලියන් වීජ ගණිතයේ තදෙවභාවි නියමය භාවිතයෙන් සරල කළ හැක, එහි සඳහන් වන්නේ A+A=A බවයි.

   In this case, the term AB is repeated three times. Applying the idempotent law:
   මෙම අවස්ථාවේදී, AB යන පදය තුන් වරක් පුනරාවර්තනය වේ. තදෙවභාවි නියමය යෙදීම:

   X = (AB) + (AB) + (AB)
   X = (AB) + (AB)
   X = AB

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

4. Which of the following OS components manages all system resources?
   පහත මෙහෙයුම් පද්ධති සංරචක අතුරෙන් සියළුම සම්පත් කළමනාකරණය කරන්නේ කවරක් මගින් ද?

   1) Shell 2) Kernel 3) File system 4) Controller 5) Scheduler

   Answer: 2)

   Explanation / පැහැදිලි කිරීම:

   The correct component that manages all system resources in an operating system is the kernel. The kernel is the core part of the operating system, responsible for controlling and coordinating all hardware and software resources. It manages tasks such as memory management, process scheduling, and input/output operations, ensuring that each application and process gets the necessary resources to function properly. The kernel acts as a bridge between the hardware and the software, making it the most critical component of the operating system.
   මෙහෙයුම් පද්ධතියක සියලුම පද්ධති සම්පත් කළමනාකරණය කරන නිවැරදි සංරචකය වන්නේ කර්නලයයි. කර්නලය යනු සියලුම දෘඩාංග සහ මෘදුකාංග සම්පත් පාලනය කිරීම සහ සම්බන්ධීකරණය කිරීම සඳහා වගකිව යුතු මෙහෙයුම් පද්ධතියේ මූලික කොටසයි. එය මතක කළමනාකරණය, ක්‍රියාවලි උපලේඛනගත කිරීම සහ ආදාන/ප්‍රතිදාන මෙහෙයුම් වැනි කාර්යයන් කළමනාකරණය කරයි, එක් එක් යෙදුම සහ ක්‍රියාවලිය නිසි ලෙස ක්‍රියාත්මක වීමට අවශ්‍ය සම්පත් ලබා ගන්නා බව සහතික කරයි. කර්නලය දෘඩාංග සහ මෘදුකාංග අතර පාලමක් ලෙස ක්‍රියා කරයි, එය මෙහෙයුම් පද්ධතියේ වඩාත්ම තීරණාත්මක අංගය බවට පත් කරයි.

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

5. Which of the following SQL statement/s correctly retrieves the names of employees who work in the 'Sales' department but excludes those with a salary above 50000?
'Sales' දෙපාර්තමේන්තුවේ සේවය කරන නමුත් 50000 ට වැඩි වැටුපක් ඇති අය බැහැර කරන සේවකයින්ගේ නම් පහත සඳහන් SQL ප්‍රකාශනවලින් නිවැරදිව ලබා ගන්නේ කුමක් ද?

A.
```
SELECT name FROM employees
WHERE department = 'Sales' AND NOT salary > 50000;
```
B.
```
SELECT name FROM employees
WHERE department LIKE 'Sales' AND salary > 50000;
```
C.
```
SELECT name FROM employees
WHERE department = 'Sales' AND salary BETWEEN 0 AND 50000;
```

1) A only. A පමණි.
2) B only. B පමණි.
3) C only. C පමණි.
4) A and C only. A හා C පමණි.
5) B and C only. B හා C පමණි.

Answer: 4)

Explanation of each statement / එක් එක් ප්‍රකාශය පැහැදිලි කිරීම:

Statement / ප්‍රකාශය A
This condition filters employees who belong to the Sales department. The expression NOT salary > 50000 means the salary is not greater than 50,000, effectively selecting employees earning 50,000 or less.
මෙම කොන්දේසිය Sales දෙපාර්තමේන්තුවට අයත් සේවකයින් පෙරහන් කරයි. NOT salary > 50000 ප්‍රකාශනය යන්නෙන් අදහස් වන්නේ වැටුප 50,000 ට වඩා වැඩි නොවන බවයි, ඵලදායී ලෙස 50,000 හෝ ඊට අඩු වැටුපක් ලබන සේවකයින් තෝරා ගනී.

This matches exactly what the question asks for: employees in Sales excluding those with salaries above 50,000. Therefore, Option A is logically correct and will retrieve the desired results.
මෙය ප්‍රශ්නය අසන දේට හරියටම ගැලපේ: 50,000ට වැඩි වැටුප් ඇති අය හැර විකුණුම්වල සේවකයින්. එබැවින්, විකල්පය A තාර්කිකව නිවැරදි වන අතර අපේක්ෂිත ප්‍රතිඵල ලබා ගනී.

Statement / ප්‍රකාශය B
Here, the condition checks for employees in the Sales department, with LIKE 'Sales' behaving like an exact match because no wildcards are used. However, the salary condition salary > 50000 selects employees who earn more than 50,000, which is the opposite of what the question wants.
මෙහි, Sales දෙපාර්තමේන්තුවේ සේවකයින් සඳහා කොන්දේසිය පරීක්ෂා කරයි, LIKE 'Sales' වයිල්ඩ්කාඩ් භාවිතා නොකරන නිසා නිශ්චිත ගැලපීමක් ලෙස හැසිරේ. කෙසේ වෙතත්, වැටුප් තත්ත්වය salary > 50000 මගින් 50,000ට වඩා උපයන සේවකයින් තෝරා ගනී, එය ප්‍රශ්නයට අවශ්‍ය දෙයට ප්‍රතිවිරුද්ධයයි.

Statement / ප්‍රකාශය C
This option filters employees who are in the Sales department and whose salaries range from 0 to 50,000 inclusive. The BETWEEN clause here explicitly captures all salaries up to and including 50,000, perfectly matching the question's requirement.
මෙම විකල්පය Sales දෙපාර්තමේන්තුවේ සේවය කරන සහ වැටුප් 0 සිට 50,000 දක්වා පරාසයක සිටින සේවකයින් පෙරහන් කරයි, ඇතුළුව. මෙහි BETWEEN වගන්තිය 50,000 දක්වා සහ ඇතුළුව සියලුම වැටුප් පැහැදිලිව ග්‍රහණය කර ගන්නා අතර, ප්‍රශ්නයේ අවශ්‍යතාවයට පරිපූර්ණව ගැලපේ.

Therefore the answer is 4) A and C only
එබැවින් පිළිතුර 4) A සහ C පමණි.

6. Which type of Programming does Python support?
පයිතන් සහාය දක්වන්නේ කුමන ආකාරයේ ක්‍රමලේඛණයට ද?

1) Object-oriented programming / වස්තු-නැඹුරු ක්‍රමලේඛණය
2) Structured programming / ව්‍යුහගත ක්‍රමලේඛණය
3) Functional programming / කාර්යබද්ධ ක්‍රමලේඛණය
4) All of the above / ඉහත සියල්ලම.
5) None of the above / ඉහත කිසිවක් නොවේ.

Answer : 4)

Python supports all of the mentioned programming paradigms:
පයිතන් සඳහන් කර ඇති සියලුම ක්‍රමලේඛනවලට ආදර්ශ සහය දක්වයි:

- **Object-oriented programming / වස්තු-නැඹුරු ක්‍රමලේඛණය (OOP)**
  Python supports OOP, allowing you to define classes, create objects, and utilize concepts like inheritance, encapsulation, and polymorphism.
  පයිතන් OOP සඳහා සහය දක්වයි, ඔබට පන්ති නිර්වචනය කිරීමට, වස්තු නිර්මාණය කිරීමට සහ උරුමය, ආවරණය කිරීම සහ බහුරූපතාව වැනි සංකල්ප භාවිතා කිරීමට ඉඩ සලසයි.

- **Structured programming / ව්‍යුහගත ක්‍රමලේඛණය**
  Python supports structured programming, which emphasizes the use of clear and logical control structures, such as loops and conditionals.
  පයිතන් ව්‍යුහගත ක්‍රමලේඛනයට සහය දක්වයි, එය ලූප සහ කොන්දේසි වැනි පැහැදිලි සහ තාර්කික පාලන ව්‍යුහයන් භාවිතා කිරීම අවධාරණය කරයි.

- **Functional programming / කාර්යබද්ධ ක්‍රමලේඛණය**
  Python also supports functional programming, which allows you to use functions as first-class objects, utilize higher-order functions, and work with concepts like map, filter, and reduce.
  පයිතන් කාර්යබද්ධ ක්‍රමලේඛනයට ද සහය දක්වයි, එමඟින් ඔබට ශ්‍රිතයන් පළමු පන්තියේ වස්තු ලෙස භාවිතා කිරීමට, ඉහළ අනුපිළිවෙලෙහි කාර්යයන් භාවිතා කිරීමට සහ සිතියම, පෙරහන සහ අඩු කිරීම වැනි සංකල්ප සමඟ වැඩ කිරීමට ඉඩ සලසයි.

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

7. What does the following code print?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් වේද?

```
def g(x, y=None):
    if y is None:
        y = []
    y += [x]
    return y

a = g(1)
b = g(2, a)
c = g(3)
s = a
s = s * 2
t = (tuple(s),)
d = {t: len(a)}
d[t] += 1
print(a, b, c, d[t])
```

1) `[1] [2] [3] 1`
2) `[1, 2] [2] [3] 2`
3) <u>`[1, 2] [1, 2] [3] 3`</u>
4) `[1, 2] [1, 2] [3] 2`
5) `[1, 2] [1, 2, 3] [3] 3`

Answer: 3)

Explanation / පැහැදිලි කිරීම:

`g(1)` - y is `None`, so `y = []`. Then `y += [1]` mutates `y` to `[1]`. Return value stored in `a` → `a = [1]`.
`g(1)` - y යනු `None`, එබැවින් `y = []`. එවිට `y += [1]`, y, `[1]` ලෙස විකෘති කරයි. `a` → `a = [1]` හි ගබඩා කර ඇති ආපසු ලබා දෙයි.

`g(2, a)` - we pass the list a as `y`. Inside `g`, `y += [2]` mutates that same list, so a becomes `[1, 2]`.
`g(2, a)` - අපි list a, `y` ලෙස සම්මත කරමු. ට ඇතුළත, `y += [2]` එම ලැයිස්තුවම විකෘති කරයි, එබැවින් `a, [1, 2]` බවට පත්වේ.

The returned value is the same list object and assigned to b. So a and b are the same list: [1, 2].
ආපසු ලබා දුන් අගය එකම ලැයිස්තු වස්තුව වන අතර `b` වෙත පවරා ඇත. එබැවින් `a` සහ `b` එකම ලැයිස්තුව වේ: `[1, 2]`.

`g(3)` - new call with no `y`, so a fresh list is created and mutated to `[3]`, returned and assigned to `c` → `c = [3]`.
`g(3)` - `y` නොමැති නව ඇමතුමකි, එබැවින් නැවුම් ලැයිස්තුවක් නිර්මාණය කර `[3]` වෙත විකෘති කර, ආපසු ලබා දී `c` → `c = [3]` වෙත පවරා ඇත.

`s = "a"; s = s * 2` → s becomes `"aa"`. `t = (s,)` → `t` is the tuple (`'aa',`).
`s = "a"; s = s * 2` → s `"aa"` බවට පත් වේ. `t = (s,)` → `t` tuple එකකි (`'aa',`).

`d = {t: len(a)}` → `len(a)` is 2 so `d[t] == 2`.
`d = {t: len(a)}` → `len(a)` 2කි. එබැවින් `d[t] == 2`.

`d[t] += 1` → integers are immutable, so this computes `2 + 1 = 3` and reassigns `d[t] = 3`.
`d[t] += 1` → නිඛිල වෙනස් කළ නොහැකි බැවින් මෙය `2 + 1 = 3` ගණනය කර `d[t] = 3` නැවත පවරයි.

`print(a, b, c, d[t])` prints the current values: `a` → `[1, 2]`, `b` → `[1, 2]`, `c` → `[3]`, `d[t]` → 3.
`print(a, b, c, d[t])` වත්මන් අගයන් මුද්‍රණය කරයි: `a` → `[1, 2]`, `b` → `[1, 2]`, `c` → `[3]`, `d[t]` → 3.
Hence the output is / එබැවින් ප්‍රතිදානය: `[1, 2] [1, 2] [3] 3`.

Therefore, the answer is Option 3)
එබැවින් පිළිතුර විකල්පය 3) වේ

8. Which of the following is a key characteristic of an agile development approach?
සුවලය සංවර්ධන ආකෘතියක ප්‍රධාන ලක්ෂණය වන්නේ පහත සඳහන් කුමක්ද?

1) Strict adherence to initial requirements / මූලික අවශ්‍යතා දැඩි ලෙස පිළිපැදීම.
2) Sequential and linear development process / අනුක්‍රමික හා රේඛීය සංවර්ධන ක්‍රියාවලිය.
3) Flexibility and iterative development / නම්‍යශීලිතාවය සහ පුනරාවර්තන සංවර්ධනය.
4) Long development cycles / දිගු සංවර්ධන චක්‍ර.
5) Limited user involvement / සීමිත පරිශීලක සහභාගිත්වය.

Answer: 3)

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

Option / විකල්පය 1
This is incorrect for agile development. Agile is designed to be adaptable to changing requirements. Instead of strictly adhering to the initial plan, agile teams continuously adjust to feedback and new information.
සුවලය සංවර්ධනය සඳහා මෙය වැරදිය. සුවලය ආකෘතිය නිර්මාණය කර ඇත්තේ වෙනස්වන අවශ්‍යතා වලට අනුවර්තනය වීමටය. ආරම්භක සැලැස්මට දැඩි ලෙස අනුගත වීම වෙනුවට, කඩිසර කණ්ඩායම් ප්‍රතිපෝෂණ සහ නව තොරතුරු වෙත අඛණ්ඩව හැඩගැසෙයි.

Option / විකල්පය 2
This is a characteristic of the traditional Waterfall method, not agile. Agile follows an iterative approach, where development is broken down into smaller, manageable increments.
මෙය සාම්ප්‍රදායික දියඇලි ආකෘතියේ ලක්ෂණයක් මිස සුවලය ආකෘතිය නොවේ. සුවලය ආකෘතිය පුනරාවර්තන ප්‍රවේශයක් අනුගමනය කරයි, එහිදී සංවර්ධනය කුඩා, කළමනාකරණය කළ හැකි වර්ධකවලට කැඩී යයි.

Option / විකල්පය 3
This is correct. Agile emphasizes flexibility, allowing teams to iterate on a project by delivering small, functional parts in short cycles (called sprints), then improving them based on user feedback.
මෙය නිවැරදියි. සුවලය ආකෘතිය නම්‍යශීලී බව අවධාරණය කරයි, කුඩා, ක්‍රියාකාරී කොටස් කෙටි චක්‍රවල (ස්ප්‍රින්ට් ලෙස හැඳින්වේ) බෙදා හැරීමෙන්, පසුව පරිශීලක ප්‍රතිපෝෂණ මත පදනම්ව ඒවා වැඩිදියුණු කිරීමෙන් ව්‍යාපෘතියක් නැවත නැවත කිරීමට කණ්ඩායම්වලට ඉඩ සලසයි.

Option / විකල්පය 4
Agile focuses on short, incremental cycles, often lasting 2–4 weeks. Long development cycles are typical of Waterfall or other traditional methods, not agile.
සුවලය ආකෘතිය කෙටි, වර්ධක චක්‍ර කෙරෙහි අවධානය යොමු කරයි, බොහෝ විට සති 2-4 ක් පවතී. දිගු සංවර්ධන චක්‍ර දියඇලි හෝ වෙනත් සම්ප්‍රදායික ක්‍රමවල සාමාන්‍ය වේ, සුවලය ආකෘතියේ එසේ නොවේ.

Option / විකල්පය 5
Agile encourages frequent collaboration with users, ensuring their involvement throughout the project. Limited user involvement would be more common in Waterfall, where feedback is often sought only at the end.
සුවලය ආකෘතිය පරිශීලකයන් සමඟ නිරන්තර සහයෝගිතාවය දිරිමත් කරයි, ව්‍යාපෘතිය පුරාවට ඔවුන්ගේ සහභාගිත්වය සහතික කරයි. බොහෝ විට ප්‍රතිපෝෂණ සොයන්නේ අවසානයේ දී පමණක් වන දියඇලි ආකෘතියේ සීමිත පරිශීලක සහභාගිත්වය වඩාත් සුලභ වනු ඇත.

So, the correct answer number is 3.
එමනිසා, නිවැරදි පිළිතුරු අංකය 3 වේ.

9. Choose the best statement about the code given below,
පහත දක්වා ඇති කේතය පිළිබඳ වඩා හොඳින්ම විස්තර කෙරෙන ප්‍රකාශය තෝරන්න,

```
body {background: #ffffff url("Cheems.png") no-repeat right top;}
```

1) The above code will give an error / ඉහත කේතය දෝෂයක් ලබා දෙනු ඇත.
2) The above code won't change anything on website. / ඉහත කේතය වෙබ් අඩවියේ කිසිවක් වෙනස් නොකරයි.
3) It is a CSS code / එය CSS කේතයකි.
4) This code will make the CSS file longer to load / මෙම කේතය CSS ගොනුව පූරණය කිරීමට දිගු කරයි.
5) This is a shorthand property in CSS / මෙය CSS හි shorthand ගුණාංගයකි.

Answer : 5)

Explanation / පැහැදිලි කිරීම:

This won't give any errors, the code will change the background colour and add a background image if the image is in the same folder as the HTML file, it is a CSS code but there is a more suitable answer, this code won't make the CSS file longer to load rather it will make it faster.
මෙමගින් කිසිදු දෝෂයක් ඇති නොවේ, කේතය පසුබිම් වර්ණය වෙනස් කර පින්තූරය HTML ගොනුවේ ඇති ෆෝල්ඩරයේම තිබේ නම් පසුබිම් රූපයක් එක් කරයි, එය CSS කේතයක් වන නමුත් වඩා සුදුසු පිළිතුරක් ඇත, මෙම CSS කේතය වෙබ්පිටුව පූරණය කිරීමට දිගු කාලයක් නොගන්නා අතර එය වේගවත් කරයි.

So, the most suitable answer is that this is a shorthand property in CSS.
එබැවින්, වඩාත්ම සුදුසු පිළිතුර මෙය CSS හි shorthand property එකකි යන්නයි

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

10. `<legend>` tag is used for,
`<legend>` උසුලනය භාවිතා කරනුයේ,

A. To provide a border for a group of related form elements.
පෝරමයකට අදාල මූලාංග සමූහයක් සඳහා මායිමක් සැපයීමට.

B. When you have a set of form controls that belong together, like a group of radio buttons or checkboxes, The `<legend>` element then provides a label or title for that group.
ඔබට රේඩියෝ බොත්තම් හෝ පිරික්සුම් කොටු සමූහයක් වැනි එකට අයත් වන ආකෘති පාලන කට්ටලයක් ඇති විට, `<legend>` මූලාංග එම කණ්ඩායම සඳහා ලේබලයක් හෝ මාතෘකාවක් සපයයි.

C. It is used to insert a space between elements that cannot be broken by word wrap or line break.
එය වචන word wrap එකකින් හෝ line break එකකින් බිඳිය නොහැකි මූලාංග අතර අවකාශයක් ඇතුළු කිරීමට භාවිතා කරයි.

1) A only. A පමණි.	2) B only. B පමණි.	3) C only. C පමණි.	4) A and B only. A හා B පමණි.	5) A and C only. A හා C පමණි.

Answer: 2)

A. To provide a border for a group of related form elements
ගුණාංග සමූහයක් සඳහා මායිමක් සැපයීමට

While using a <fieldset> element with CSS can create a border-like effect, the <legend> tag itself doesn't control borders.
CSS සමඟින් <fieldset> ගුණාංගය භාවිතා කිරීමෙන් මායිම් වැනි බලපෑමක් ඇති කළ හැකි අතර, <legend> උසුලනය මායිම් පාලනය නොකරයි.

B. When you have a set of form controls that belong together, like a group of radio buttons or checkboxes, The <legend> element then provides a label or title for that group.
ඔබට රේඩියෝ බොත්තම් හෝ පිරික්සුම් කොටු සමූහයක් වැනි එකට අයත් වන පෝරම පාලන කට්ටලයක් ඇති විට, <legend> මූලද්‍රව්‍යය එම කණ්ඩායම සඳහා ලේබලයක් හෝ මාතෘකාවක් සපයයි.

This is the correct usage of the <legend> tag.
<legend> උසුලනයේ නිවැරදි භාවිතය මෙයයි.

C. It is used to insert a space between elements that cannot be broken by word wrap or line break.
එය වචන එතුමකින් හෝ රේඛා බිඳීමෙන් බිඳිය නොහැකි මූලාංග අතර අවකාශයක් ඇතුළු කිරීමට භාවිතා කරයි.

The <legend> tag doesn't affect spacing. Use CSS properties like margin or padding for spacing control.
<legend> උසුලනය පරතරයට බලපාන්නේ නැත. පරතරය පාලනය සඳහා margin හෝ padding වැනි CSS ගුණාංග භාවිතා කරන්න.

Therefore, only option B accurately describes the purpose of the <legend> tag.
එබැවින්, B විකල්පය පමණක් <legend> උසුලනයේ අරමුණ නිවැරදිව විස්තර කරයි.