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 463 MARKING

2026/03/18
ictfromabc
Ravindu Bandaranayake

1. In a modern gaming console, which of the following components commonly employs CMOS technology for high-speed, temporary data storage that is crucial for real-time rendering and gameplay?
නවීන gaming කොන්සෝලයක, තථ්‍ය කාලීන විදැහුම්කරණය සහ gaming සඳහා ඉතා වැදගත් වන අධිවේගී, තාවකාලික දත්ත ගබඩා කිරීම සඳහා CMOS තාක්ෂණය බහුලව භාවිතා කරන්නේ පහත සඳහන් කුමන සංරචකයද?

    1) The Blu-ray disc drive / Blu-ray තැටි ධාවකය.
    2) The internal hard disk drive or solid-state drive
    අභ්‍යන්තර දෘඪ තැටි ධාවකය හෝ ඝන-තත්ව ධාවකය.
    3) The main graphics processing unit's (GPU) dedicated memory.
    ප්‍රධාන ග්‍රැෆික් සැකසුම් ඒකකයේ (GPU) මතකය.
    4) The power supply unit / බලශක්ති සැපයුම් ඒකකය.
    5) The network interface card / ජාල අතුරුමුහුණත් කාඩ්පත.

**Answer: 3)**

**Explanation / පැහැදිලි කිරීම:**

CMOS (Complementary Metal-Oxide-Semiconductor) technology is widely used in memory chips due to its low power consumption and high integration density. This makes it ideal for creating large and fast memory modules.
CMOS තාක්ෂණය එහි අඩු බල පරිභෝජනය සහ ඉහළ ඒකාබද්ධ ඝනත්වය හේතුවෙන් මතක චිප වල බහුලව භාවිතා වේ. මෙය විශාල හා වේගවත් මතක මොඩියුල නිර්මාණය කිරීම සඳහා වඩාත් සුදුසු වේ.

In a gaming console, the GPU is responsible for rendering complex visuals in real-time. To do this efficiently, it requires very fast access to a large amount of texture data, frame buffers, and other graphical information. This dedicated, high-speed memory directly connected to the GPU is typically implemented using CMOS-based memory technologies like GDDR (Graphics Double Data Rate) SDRAM.
Gaming කොන්සෝලයකදී, සංකීර්ණ දෘශ්‍ය තත්‍ය කාලීනව විදැහුම් කිරීම සඳහා GPU වගකිව යුතුය. මෙය කාර්යක්ෂමව කිරීමට, එයට වයනය දත්ත (texture data), රාමු බෆර සහ අනෙකුත් චිත්‍රක තොරතුරු විශාල ප්‍රමාණයකට ඉතා වේගවත් ප්‍රවේශයක් අවශ්‍ය වේ. GPU වෙත සෘජුවම සම්බන්ධ කර ඇති මෙම කැපවූ, අධිවේගී මතකය සාමාන්‍යයෙන් ක්‍රියාත්මක කරනු ලබන්නේ GDDR, SDRAM වැනි CMOS මත පදනම් වූ මතක තාක්ෂණයන් භාවිතා කරමිනි.

Therefore, the correct answer is option 3).
එබැවින් නිවැරදි පිළිතුර විකල්පය 3) වේ.

2. What is the answer of subtraction given below?
පහත දක්වා ඇති අඩුකිරීමේ පිළිතුර කුමක්ද?

$8B_{16}$ - $31_8$ = ………………

1) <u>$162_8$</u> 2) $62_{16}$ 3) $1101110_2$ 4) $166_8$ 5) $6C_{16}$

**Answer: 1)**

**Explanation / පැහැදිලි කිරීම:**

| Converting Hexadecimal into Decimal<br>ෂඩ් දශම දශමයට පරිවර්තනය කිරීම | |
|---|---|
| $8B_{16}$ | |
| $16^1$ | $16^0$ |
| 16 × 8 = 128 | 1 × 11 = 11 |
| $8B_{16} = 139_{10}$ | |

| Converting Octal into Decimal<br>අෂ්ටක දශමයට පරිවර්තනය කිරීම | |
|---|---|
| $31_8$ | |
| $8^1$ | $8^0$ |
| 3 × 8 = 24 | 1 × 1 = 1 |
| $31_8 = 25_{10}$ | |

$8B_{16} – 31_8 = 139_{10} – 25_{10} = 114_{10}$.

| Converting Decimal into Hexadecimal<br>දශම ෂඩ් දශම බවට පරිවර්තනය කිරීම | |
|---|---|
| $114_{10}$ | |
| $16^1$ | $16^0$ |
| 16 | 1 |
| 16 × 7 | 1 × 2 |
| $114_{10} = 72_{16}$ | |

| Converting Decimal into Octal<br>දශම අෂ්ටක බවට පරිවර්තනය කිරීම | | |
|---|---|---|
| $114_{10}$ | | |
| 82 | 81 | 80 |
| 64 | 8 | 1 |
| 64 × 1 | 8 × 6 | 1 × 2 |
| $114_{10} = 162_8$ | | |

| Converting Decimal into Binary<br>දශම ද්විමය බවට පරිවර්තනය කිරීම | | | | | | | |
|---|---|---|---|---|---|---|---|
| $114_{10}$ | | | | | | | |
| 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 128 × 0 | 64 × 1 | 32 × 1 | 16 × 1 | 8 × 0 | 4 × 0 | 2 × 1 | 1 × 0 |
| $114_{10} = 01110010_2$ | | | | | | | |

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

3. Which of the following statements about full adders and half adders is true?
පූර්ණ ආකලකය සහ අර්ධ ආකලකය පිළිබඳ පහත සඳහන් කුමන ප්‍රකාශය සත්‍ය ද?

1) A half adder can add two binary digits and includes a carry in input.
අර්ධ ආකලකයට ද්විමය ඉලක්කම් දෙකක් එකතු කළ හැකි අතර carry in ආදානයක් ඇතුළත් වේ.

2) A full adder can add two binary digits but does not include a carry in input.
පූර්ණ ආකලකයට ද්විමය ඉලක්කම් දෙකක් එකතු කළ හැකි නමුත් carry in ආදානයක් ඇතුළත් නොවේ.

3) A half adder produces a sum and a carry out output but cannot handle a carry in input.
අර්ධ ආකලකය එකතුවක් සහ carry out ප්‍රතිදානයක් නිපදවන නමුත් carry in ආදානයක් හැසිරවිය නොහැක.

4) A full adder can only add two binary digits and cannot handle a carry in input.
පූර්ණ ආකලකයට ද්විමය ඉලක්කම් දෙකක් පමණක් එකතු කළ හැකි අතර carry in ආදානයක් හැසිරවිය නොහැක.

5) Both full adders and half adders require three inputs: two significant bits and a carry in bit.
පූර්ණ ආකලකය සහ අර්ධ ආකලකය යන දෙකටම ආදාන තුනක් අවශ්‍ය වේ. (සැලකිය යුතු බිටු දෙකක් සහ carry in බිට් එකක්)

**Answer: 3)**

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

**Statement / ප්‍රකාශය 1)**
This is incorrect. A half adder adds two binary digits but does not include a carry in input. It only produces a sum and a carry out.
මෙය නිවැරදි නොවේ. අර්ධ ආකලකයක් ද්විමය ඉලක්කම් දෙකක් එකතු කරන නමුත් carry in ආදානය ඇතුළත් නොවේ. එය නිෂ්පාදනය කරන්නේ එකතුවක් සහ carry out එකක් පමණි.

**Statement / ප්‍රකාශය 2)**
This is incorrect. A full adder can add three binary digits and does include a carry in input, which allows it to handle carry-over from previous additions.
මෙය නිවැරදි නොවේ. පූර්ණ ආකලකයකට ද්විමය ඉලක්කම් තුනක් එකතු කළ හැකි අතර එයට carry in ආදානය ඇතුළත් වේ, එය පෙර එකතු කිරීම් වලින් carry-over හැසිරවීමට ඉඩ සලසයි.

**Statement / ප්‍රකාශය 3)**
This is correct. A half adder takes two binary inputs, produces a sum and a carry out, but does not have a carry in input. It is only suitable for adding two single bits without considering any carry from previous additions.
මෙය නිවැරදි වේ. අර්ධ ආකලකයක් ද්විමය ආදාන දෙකක් ගනී, එකතුවක් සහ carry out එකක් නිපදවයි, නමුත් carry in ආදානයක් නොමැත. එය යෝග්‍ය වන්නේ පෙර එකතු කිරීම් වලින් කිසිදු රැගෙන යාමක් නොසලකා තනි බිටු දෙකක් එකතු කිරීම සඳහා පමණි.

**Statement / ප්‍රකාශය 4)**
This is incorrect. A full adder can handle two binary digits and also includes a carry in input, making it capable of handling multi-bit binary addition by taking into account carry-over from previous additions.
මෙය නිවැරදි නොවේ. පූර්ණ ආකලකයකට ද්විමය ඉලක්කම් දෙකක් හැසිරවිය හැකි අතර carry in ආදානයක් ද ඇතුළත් වේ, පෙර එකතු කිරීම් වලින් රැගෙන යාම සැලකිල්ලට ගනිමින් බහු-බිටු ද්විමය එකතු කිරීම හැසිරවීමේ හැකියාව එයට ඇත.

**Statement / ප්‍රකාශය 5)**
This is incorrect. While a full adder requires three inputs (two significant bits and a carry in), a half adder only requires two inputs (the two binary digits) and does not have a carry in input.
මෙය නිවැරදි නොවේ. පූර්ණ ආකලකයකට ආදාන තුනක් (සැලකිය යුතු බිටු දෙකක් සහ carry in) අවශ්‍ය වන අතර, අර්ධ ආකලකයකට අවශ්‍ය වන්නේ ආදාන දෙකක් (ද්විමය ඉලක්කම් දෙක) පමණක් වන අතර carry in ආදානයක් නොමැත.

So, the correct answer is indeed 3).
එබැවින්, නිවැරදි පිළිතුර 3) වේ.

4. Which scheduling algorithm allows each process to run for a fixed time slice before moving to the next process?
මීළඟ ක්‍රියාවලියට යාමට පෙර එක් එක් ක්‍රියාවලිය ස්ථාවර කාල කොටසක් සඳහා ක්‍රියාත්මක කිරීමට ඉඩ දෙන නියමකරණ ඇල්ගොරිතමය කුමක් ද?

1) First-Come, First-Served
2) Shortest Job First
3) Round Robin Algorithm
4) Priority Scheduling
5) Multilevel Queue Scheduling

**Answer: 3)**

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 First-Come, First-Served (FCFS) algorithm schedules processes in the order they arrive without fixed time slices.
මෙම ප්‍රකාශය අනුව එය අසත්‍ය වේ. First-Come, First-Served (FCFS) ඇල්ගොරිතම නියමිත කාල කොටස් නොමැතිව ඔවුන් පැමිණෙන අනුපිළිවෙලට ක්‍රියාවලි නියමකරණය කරයි.

**Statement / ප්‍රකාශය 2)**
According to this statement, it's false. The Shortest Job First (SJF) algorithm schedules processes based on the shortest execution time needed but does not use fixed time slices.
මෙම ප්‍රකාශය අනුව එය අසත්‍ය වේ. Shortest Job First (SJF) ඇල්ගොරිතම අවශ්‍ය කෙටිම ක්‍රියාත්මක කිරීමේ කාලය මත පදනම්ව ක්‍රියාවලි නියමකරණය කරන නමුත් ස්ථාවර කාල කොටස් භාවිතා නොකරයි.

**Statement / ප්‍රකාශය 3)**
According to this statement, it's true. The Round Robin (RR) scheduling algorithm allocates a fixed time slice or quantum to each process in a cyclic order before moving to the next process.
මෙම ප්‍රකාශය අනුව එය සත්‍ය වේ. Round Robin (RR) නියමකරණ ඇල්ගොරිතම ඊළඟ ක්‍රියාවලියට යාමට පෙර එක් එක් ක්‍රියාවලියට චක්‍රීය අනුපිළිවෙලකට ස්ථාවර කාල කොටසක් හෝ ඉතා කුඩා කාලයක් (quantum) වෙන් කරයි.

**Statement / ප්‍රකාශය 4)**
According to this statement, it's false. Priority Scheduling selects processes based on priority levels and does not use fixed time slices.
මෙම ප්‍රකාශය අනුව එය අසත්‍ය වේ. Priority Scheduling ප්‍රමුඛතා මට්ටම් මත පදනම්ව ක්‍රියාවලි තෝරා ගන්නා අතර ස්ථාවර කාල පෙති භාවිතා නොකරයි.

**Statement / ප්‍රකාශය 5)**
According to this statement, it's false. Multilevel Queue Scheduling organizes processes into different queues based on priority or other criteria but does not inherently use fixed time slices.
මෙම ප්‍රකාශය අනුව එය අසත්‍ය වේ. බහු මට්ටමේ පෝලිම් උපලේඛනගත කිරීම ප්‍රමුඛතාවය හෝ වෙනත් නිර්ණායක මත පදනම්ව විවිධ පෝලිම් වලට ක්‍රියාවලි සංවිධානය කරන නමුත් සහජයෙන්ම ස්ථාවර කාල කොටස් භාවිතා නොකරයි.

Therefore, the correct answer is 3).
එබැවින්, නිවැරදි පිළිතුර නම් 3) ය.

5. Given tables Employees and Departments with a foreign key `dept_id` in Employees referencing Departments, which SQL query correctly retrieves all employees and their respective department names, including employees without a department?
ලබා දී ඇති වගු, සේවකයින් යොමු කරන දෙපාර්තමේන්තු වල ආගන්තුක යතුර `dept_id` සහිත සේවකයින් සහ දෙපාර්තමේන්තු, විමසුමෙන් දෙපාර්තමේන්තුවක් නොමැති සේවකයින් ඇතුළුව සියලුම සේවකයින් සහ ඔවුන්ගේ අදාළ දෙපාර්තමේන්තු නම් නිවැරදිව ලබා ගැනීමට අදාළ නිවැරදි SQL ප්‍රකාශනය වන්නේ කුමක්ද?

   1) `SELECT Employees.name, Departments.dept_name FROM Employees`
   `INNER JOIN Departments ON Employees.dept_id = Departments.dept_id;`
   2) `SELECT Employees.name, Departments.dept_name FROM Employees`
   `LEFT JOIN Departments ON Employees.dept_id = Departments.dept_id;`
   3) `SELECT Employees.name, Departments.dept_name FROM Employees`
   `RIGHT JOIN Departments ON Employees.dept_id = Departments.dept_id;`
   4) `SELECT Employees.name, Departments.dept_name FROM Employees`
   `FULL JOIN Departments ON Employees.dept_id = Departments.dept_id;`
   5) `SELECT Employees.name, Departments.dept_name FROM Employees`
   `CROSS JOIN Departments ON Employees.dept_id = Departments.dept_id;`

**Answer : 2)**

**Explanation / පැහැදිලි කිරීම:**

This query uses a `LEFT JOIN`, which retrieves all employees, regardless of whether they have a matching department in the `Departments` table. For employees without a department, the `dept_name` will be `NULL`, ensuring that no employees are excluded from the results. This is the correct query to include all employees and their department names, including those without a department.
මෙම විමසුම `LEFT JOIN` භාවිතා කරයි. එය `Departments` වගුවේ ගැලපෙන දෙපාර්තමේන්තුවක් තිබේද යන්න නොසලකා සියලුම සේවකයින් ලබා ගනී. දෙපාර්තමේන්තුවක් නොමැති සේවකයින් සඳහා, දෙපාර්තමේන්තුවේ නම (`dept_name`) `NULL` වනු ඇත, එමඟින් කිසිදු සේවකයෙකු ප්‍රතිඵල වලින් බැහැර නොවන බව සහතික කෙරේ. දෙපාර්තමේන්තුවක් නොමැති අය ඇතුළුව සියලුම සේවකයින් සහ ඔවුන්ගේ දෙපාර්තමේන්තු නම් ඇතුළත් කිරීමට මෙය නිවැරදි විමසුමයි.

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

6. Which method in dictionaries will return a list of all the keys in the dictionary?
dictionary වල ඇති සියලුම keys ලැයිස්තුවක් ලබා දෙන්නේ කුමන ක්‍රමය ද?

   1) `getKeys()` 2) `keys()` 3) `values()` 4) `items()` 5) `listKeys()`

**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. There is no `getKeys()` method in Python dictionaries.
  මෙම ප්‍රකාශය අනුව එය අසත්‍ය වේ. පයිතන් ශබ්දකෝෂවල `getKeys()` ක්‍රමයක් නොමැත.
- **Statement / ප්‍රකාශය 2)**
  According to this statement, it's true. The `keys()` method returns a view object that displays a list of all the keys in the dictionary.
  මෙම ප්‍රකාශය අනුව එය සත්‍ය වේ. `keys()` ක්‍රමය ශබ්දකෝෂයේ ඇති සියලුම යතුරු ලැයිස්තුවක් පෙන්වන දර්ශන වස්තුවක් ලබා දෙයි.
- **Statement / ප්‍රකාශය 3)**
  According to this statement, it's false. The `values()` method returns a view object that displays a list of all the values in the dictionary, not the keys.
  මෙම ප්‍රකාශය අනුව එය අසත්‍ය වේ. `values()` ක්‍රමය මඟින් යතුරු නොව ශබ්දකෝෂයේ ඇති සියලුම අගයන් ලැයිස්තුවක් පෙන්වන දර්ශන වස්තුවක් ලබා දෙයි.

- **Statement / ප්‍රකාශය 4)**
  According to this statement, it's false. The `items()` method returns a view object that displays a list of tuples, each containing a key-value pair from the dictionary.
  මෙම ප්‍රකාශය අනුව එය අසත්‍ය වේ. `items()` ක්‍රමය මඟින් tuple ලැයිස්තුවක් පෙන්වන දර්ශන වස්තුවක් ආපසු ලබා දෙයි, ඒ සෑම එකක්ම ශබ්දකෝෂයේ යතුරු-අගය යුගලයක් අඩංගු වේ.
- **Statement / ප්‍රකාශය 5)**
  According to this statement, it's false. There is no `listKeys()` method in Python dictionaries.
  මෙම ප්‍රකාශය අනුව එය අසත්‍ය වේ. පයිතන් ශබ්දකෝෂවල `listKeys()` ක්‍රමයක් නොමැත.

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

7. What is the output of below python code?
   පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
def foo(x):
    x[0] = ['def']
    x[1] = ['abc']
    return id(x)
q = ['abc', 'def']
print(id(q) == foo(q))
```

1) `2`
2) <u>`True`</u>
3) `False`
4) `None`
5) `Error`

**Answer: 2)**

**Explanation / පැහැදිලි කිරීම:**

The function `foo(x)` modifies the contents of the list `x` by changing its first two elements and then returns the memory address (`id(x)`) of the list. Since Python lists are mutable, any changes made to the list inside the function do not affect its memory location. The variable `q` is a list that is passed to the function `foo(q)`. Although the contents of `q` are modified, its memory address remains the same.
`foo(x)` ශ්‍රිතය `x` ලැයිස්තුවේ පළමු මූලාංග දෙක වෙනස් කිරීමෙන් එහි අන්තර්ගතය වෙනස් කර ලැයිස්තුවේ මතක ලිපිනය (`id(x)`) ලබා දෙයි. පයිතන් ලැයිස්තු විකෘති වන බැවින්, ශ්‍රිතය තුළ ඇති ලැයිස්තුවට කරන ලද ඕනෑම වෙනස්කමක් එහි මතක ස්ථානයට බලපාන්නේ නැත. `q` විචල්‍යය යනු `foo(q)` ශ්‍රිතයට යවන ලැයිස්තුවකි. `q` හි අන්තර්ගතය වෙනස් කර ඇතත්, එහි මතක ලිපිනය එලෙසම පවතී.

The comparison `id(q) == foo(q)` evaluates to True because the `id(q)` before and after calling the function is identical, showing that the list retains its original memory location despite being altered.
`id(q) == foo(q)` සංසන්දනය සත්‍ය ලෙස ඇගයීමට ලක් කරන්නේ ශ්‍රිතය ඇමතීමට පෙර සහ පසු `id(q)` එක සමාන වන නිසා, ලැයිස්තුව වෙනස් කර තිබියදීත් එහි මුල් මතක ස්ථානය රඳවා ගන්නා බව පෙන්වයි.

Output / ප්‍රතිදානය: `True`

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

8. How many characters print by this python code?
මෙම පයිතන් කේතය මගින් අනුලක්ෂණ කොපමණ ප්‍රමාණයක් මුද්‍රණය කරන්නේද?

```
for i in range (1,5):
    for k in range (1,i+2):
       print (k*i,end='_')
```

1) 14 2) 28 3) 32 4) 29 5) 31

**Answer: 3)**

Let's calculate the total characters printed:
මුද්‍රණය කර ඇති සම්පූර්ණ අක්ෂර ගණනය කරමු:

When `i = 1`, the inner loop iterates once:
`i = 1` වන විට, අභ්‍යන්තර ලූපය එක්වරක් පුනරාවර්තනය වේ:

```
1 * 1 = 1_
1 * 2 = 2_
```

Number of characters when `i = 1` is 4
`i = 1` වන විට අක්ෂර ගණන: 4

When `i = 2`, the inner loop iterates twice:
`i = 2` වන විට, අභ්‍යන්තර ලූපය දෙවරක් පුනරාවර්තනය වේ:

```
1 * 2 = 2_
2 * 2 = 4_
3 * 2 = 6_
```

Number of characters when `i = 2` is 6
`i = 2` වන විට අක්ෂර ගණන: 6

When `i = 3`, the inner loop iterates three times:
`i = 3` වන විට, අභ්‍යන්තර ලූපය තෙවරක් පුනරාවර්තනය වේ:

```
1 * 3 = 3_
2 * 3 = 6_
3 * 3 = 9_
4 * 3 = 12_
```

Number of characters when `i = 3` is 9
`i = 3` වන විට අක්ෂර ගණන: 9

When `i = 4`, the inner loop iterates four times:
`i = 4` වන විට, අභ්‍යන්තර ලූපය සිව්වරක් පුනරාවර්තනය වේ:

```
1 * 4 = 4_
2 * 4 = 8_
3 * 4 = 12_
4 * 4 = 16_
5 * 4 = 20_
```

Number of characters when `i = 4` is 13
`i = 4` වන විට අක්ෂර ගණන: 13

Adding all these up:
මේ සියල්ල එකතු කිරීම:

Total number of characters: 4+6+9+13 = 32
මුළු අක්ෂර ගණන: 4+6+9+13 = 32

So, the correct answer should be: (3) 32
එබැවින්, නිවැරදි පිළිතුර විය යුත්තේ (3) 32 වේ.

9. Which of the following CSS rules will create a blue text shadow? (that is 3px to the right, 3px down, with no blur)
පහත සඳහන් කුමන CSS රීතිය මඟින් නිල් පැහැති පාඨ සෙවනැල්ලක් නිර්මාණය කළ හැකිද? (එහි දකුණට 3px, පහළට 3px, කිසිදු බොඳවීමක් නොමැත)

1) `text-shadow: blue 3px 3px 0px;` 2) `text-shadow: 3px 3px 0px blue;`
3) `text-shadow: (3px, 3px, 0px, blue);` 4) `text-shadow: [3px 3px 0px] blue;`
5) `text-shadow: 3px 3px blue;`

**Answer: 2)**

```
text-shadow: 3px 3px 0px blue;
```

The correct syntax for the `text-shadow` property in CSS is:
CSS හි `text-shadow` ගුණාංගය සඳහා නිවැරදි වාක්‍ය ඛණ්ඩය වන්නේ:

```
text-shadow: horizontal-offset  vertical-offset  blur-radius color;
```

3px - Moves the shadow 3px to the right.correct
සෙවනැල්ල 3px දකුණට ගෙන යයි. නිවැරදියි

3px - Moves the shadow 3px down.correct
සෙවනැල්ල 3px පහළට ගෙන යයි. නිවැරදියි

0px - No blur (sharp shadow).correct
blur වී නැත. නිවැරදියි

blue - Shadow color.correct
සෙවනැලි වර්ණය. නිවැරදියි

Why the Others Are Wrong? / අනෙක් ඒවා වැරදි වන්නේ ඇයි ?

**1)`text-shadow: blue 3px 3px 0px;`**
Wrong order! The **color** should be at the end, not the beginning.
වැරදි අනුපිළිවෙලකි! වර්ණය ආරම්භයට නොව අවසානයේ තිබිය යුතුය.

**3) `text-shadow: (3px, 3px, 0px, blue);`**
Wrong syntax! **No parentheses** are used in text-shadow.
වැරදි වාක්‍ය ඛණ්ඩයකි! text-shadow හි වරහන් භාවිතා නොකෙරේ.

**4) `text-shadow: [3px 3px 0px] blue;`**
Wrong syntax! **No brackets** ([]) are used in CSS properties.
CSS ගුණාංගවල **වරහන්** ([]) භාවිතා නොවේ.

**5) `text-shadow: 3px 3px blue;`**
Missing the `blur-radius` value (should be 0px for no blur).
`blur-radius` අගය මග හැරී ඇත (බොඳ නොවීම සඳහා 0px විය යුතුය).

10. Which of the following CSS selectors is correctly used to style all HTML elements with the class name `"highlight"`?
`"highlight"` පන්ති නාමය සහිත සියලුම HTML මූලාංග මෝස්තර කිරීමට පහත සඳහන් CSS තේරීම්කාරක අතරින් නිවැරදිව භාවිතා කරන්නේ කුමක්ද?

    1) `#highlight {background-color: yellow;}`
    2) `.highlight {background-color: yellow;}`
    3) `highlight {background-color: yellow;}`
    4) `*highlight {background-color: yellow;}`
    5) `highlight. {background-color: yellow;}`

    **Answer: 2)**

    **Explanation / පැහැදිලි කිරීම:**

    In CSS, class selectors are used to target HTML elements that have a specific class attribute. Class selectors are prefixed with a dot (.).
    CSS හි, නිශ්චිත පන්ති ගුණාංගයක් ඇති HTML මූලාංග ඉලක්ක කිරීමට පන්ති තේරීම් භාවිතා කරයි. පන්ති තේරීම් තිතක් (.) සමඟ උපසර්ග කර ඇත.

    1) **`#highlight {background-color: yellow;}`**:
    This is an ID selector. ID selectors are used to target a single, unique HTML element with a specific `id` attribute. They are prefixed with a hash symbol (#).
    මෙය ID තේරීමකි. ID තේරීම් භාවිතා කරනු ලබන්නේ නිශ්චිත ID ගුණාංගයක් සහිත තනි, අනන්‍ය HTML මූලාංගයක් ඉලක්ක කිරීමටයි. ඒවා හැෂ් සංකේතයක් (#) සමඟ උපසර්ග කර ඇත.

    3) **`highlight {background-color: yellow;}`**:
    This is an element selector (or type selector). Element selectors target all HTML elements of a specific type (ex: `div`, `p`, `span`).
    මෙය මූලාංග තේරීමකි. මූලාංග තේරීම්කරුවන් නිශ්චිත වර්ගයක සියලුම HTML මූලාංග ඉලක්ක කරයි.

    4) **`*highlight {background-color: yellow;}`**:
    This is not a standard CSS selector. The asterisk (*) is the universal selector, which targets all HTML elements. While you can combine it with other selectors, this specific syntax is incorrect for targeting a class.
    මෙය සම්මත CSS තේරීමක් නොවේ. තරු ලකුණ (*) යනු සියලුම HTML මූලාංග ඉලක්ක කරන විශ්විය තේරීමයි. ඔබට එය අනෙකුත් තේරීම් සමඟ ඒකාබද්ධ කළ හැකි වුවද, පන්තියක් ඉලක්ක කිරීම සඳහා මෙම නිශ්චිත වාක්‍ය ඛණ්ඩය වැරදිය.

    5) **`highlight. {background-color: yellow;}`**:
    This is also incorrect syntax for a class selector. The dot must directly precede the class name.
    මෙය පන්ති තේරීමක් සඳහා වැරදි වාක්‍ය ඛණ්ඩයකි. තිත පන්ති නාමයට කෙළින්ම ඉදිරියෙන් තිබිය යුතුය.

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