සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved]

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

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 500 MARKING

2026/04/24
ictfromabc
Ravindu Bandaranayake

1. Which moral and legal ideology do the following belong to?
පහත සඳහන් කරුණු අයත් වන්නේ කුමන සදාචාරාත්මක හා නෛතික මතවාදයටද?

A. Claiming someone else's work as your own.
වෙනත් අයෙකුගේ නිර්මාණයක් තමන්ගේ සේ කියාපෑම.

B. This situation has increased with the presentation of data and information in electronic form.
දත්ත, තොරතුරු ඉලෙක්ට්‍රොනික ආකාරයට ඉදිරිපත් කිරීමත් සමගම මෙම තත්වය ඉහල ගොස් ඇත.

C. This can be avoided by citing someone else's work as a quote.
වෙනත් අයෙකුගේ නිර්මාණයක් ලබා ගන්නේ නම් එය උපුටා ගැනීමක් සේ දැක්වීම මගින් මෙය වළක්වා ගත හැක.

1) Copyright Law ප්‍රකාශන හිමිකම
2) Plagiarism රචනා චෞරත්වය
3) Phishing තතුබෑම
4) Piracy චෞරත්වය
5) Privacy පුද්ගලිකත්වය

Answer: 2)

Explanation / පැහැදිලි කිරීම:

- Copyright is the author's right to copy his material. Any copying of the material by another person may result in a copyright lawsuit.
ප්‍රකාශන හිමිකම යනු කතෘට තම ද්‍රව්‍ය පිටපත් කිරීමට ඇති අයිතියයි. වෙනත් පුද්ගලයෙකු විසින් එම ද්‍රව්‍ය පිටපත් කිරීමක් සිදුකළ හොත් ප්‍රකාශන හිමිකම යටතේ නඩු පැවරිය හැක.

- Plagiarism is the act of using someone else's work, ideas, or expressions without giving proper credit, and presenting them as your own.
වෙනත් කෙනෙකුගේ කෘති, අදහස් හෝ ප්‍රකාශන නිසි ගෞරවය ලබා නොදී භාවිතා කිරීම සහ ඒවා ඔබේම ලෙස ඉදිරිපත් කිරීම, රචනා චෞරත්වය වේ.

- Phishing is the act of stealing user details through a fake attack created on a real website.
තතුබෑම යනු සැබෑ ලෙසම පවතින් වෙබ් අඩවියක ආකෘතියකට නිර්මාණය කරන ලද ව්‍යාජ ආකෘතියක් හරහා පරිශීලක විස්තර සොරකම් කර ගැනීම වේ.

- Piracy is obtaining counterfeit copies of copyrighted software. This is also known as software piracy.
චෞරත්වය යනු හිමිකම් ආශ්‍රිත මෘදුකාංග වල ව්‍යාජ පිටපත් ලබා ගැනීම වේ. මෘදුකාංග කොල්ලකෑම ලෙසද මෙය හඳුන්වනු ලබයි.

- Privacy is the protection of personal information in computer data banks from misuse.
පුද්ගලිකත්වය යනු පරිගණක දත්ත බැංකු වල ඇති පුද්ගලික තොරතුරු වැරදි ලෙස භාවිතා කිරීමෙන් ආරක්ෂා කර ගැනීමයි.

When we consider these 3 points A, B, C, it is clear that this means about Plagiarism.
මෙම A, B, C යන කරුණු 3 ගැන සැලකූ විට මෙයින් අදහස් වන්නේ රචනා චෞරත්වය ගැන බව පැහැදිලි වේ.

2. In the IEEE single-precision floating-point standard, what does the exponent field represent?
IEEE single-precision floating-point ප්‍රමිතියෙහි, exponent ක්ෂේත්‍රය කුමක් නිරූපණය කරන්නේද?

1) The sign of the mantissa. / Mantissa හි ලකුණ
2) The fractional part of the number. / සංඛ්‍යාවේ භාගික කොටස.
3) <u>The power of 2 by which the mantissa is multiplied. / Mantissa ගුණ කරන 2 හි බලය.</u>
4) The total number of bits used to represent the number.
සංඛ්‍යාව නිරූපණය කිරීමට භාවිතා කරන මුළු බිටු ගණන.
5) The number of significant digits in the number. / සංඛ්‍යාවේ ඇති සැලකිය යුතු සංඛ්‍යාංක ගණන.

Answer: 3)

Explanation / පැහැදිලි කිරීම:

This standard represents floating-point numbers using 32 bits is divided into three parts:
මෙම ප්‍රමිතිය බිටු 32 ක් භාවිතා කරමින් ඉපිලුම් සංඛ්‍යා නිරූපණය කරන අතර එය කොටස් තුනකට බෙදා ඇත:

- **Sign bit / ලකුණු බිටුව (1 bit):**
Indicates whether the number is positive or negative. / සංඛ්‍යාව ධන ද සෘණ ද යන්න දක්වයි.

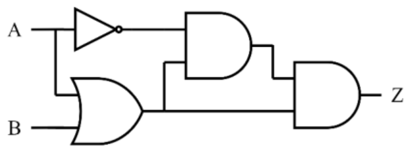
- **Exponent field / ඝාතිය ක්ෂේත්‍රය (8 bits):**
Stores the exponent, which determines the magnitude of the number.
සංඛ්‍යාවේ විශාලත්වය තීරණය කරන ඝාතය ගබඩා කරයි.

- **Mantissa field (23 bits):**
Stores the significant digits of the number. / සංඛ්‍යාවේ සැලකිය යුතු සංඛ්‍යාංක ගබඩා කරයි.

The exponent field represents the power of 2 that the mantissa is multiplied by. This allows the representation of a wide range of numbers, both very large and very small.
ඝාතිය ක්ෂේත්‍රය මගින් ජ්බඑසී හි ගුණ කරන ලද 2 හි බලය නිරූපණය කරයි. මෙය ඉතා විශාල සහ ඉතා කුඩා සංඛ්‍යා යන පුළුල් පරාසයක නිරූපණයට ඉඩ සලසයි.

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

3. What is the Boolean expression that can be obtain from the below logic circuit?
පහත තාර්කික පරිපථයෙන් ලබාගත හැකි බූලීය ප්‍රකාශනය කුමක්ද?

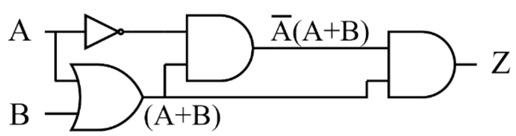


1) 1 2) <u>$\bar{A}B$</u> 3) A 4) 0 5) $(\bar{A}+B)(A.B)$

Answer: 2)

Explanation / පැහැදිලි කිරීම:

Let's obtained an expression from the circuit.
පරිපථයෙන් ප්‍රකාශනයක් ලබා ගනිමු.



Let's simplify the expression.
ප්‍රකාශනය සරල කර ගනිමු.

$Z=(A+B)(A+B)\bar{A}$

$=(AA+AB+AB+BB)\bar{A}$	Distributive law / විඝටන න්‍යාය
$=(A+AB+AB+BB)\bar{A}$	Idempotent law / තද්වභාවී න්‍යාය
$=(A+AB+BB)\bar{A}$ I	dempotent law / තද්වභාවී න්‍යාය
$=(A+AB+B)\bar{A}$	Idempotent law / තද්වභාවී න්‍යාය
$=(A(1+B)+B)\ \bar{A}$	Distributive law / විඝටන න්‍යාය
$=(A(1)+B)\bar{A}$	Identity law / සර්වසාම්‍ය න්‍යාය
$=(A+B)\bar{A}$	Identity law / සර්වසාම්‍ය න්‍යාය
$=A\bar{A}+\bar{A}B$	Distributive law / විඝටන න්‍යාය
$=0+\bar{A}B$	Inverse law / ප්‍රතිලෝම න්‍යාය
$=\bar{A}B$	Identity law / සර්වසාම්‍ය න්‍යාය

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

4. Which transition in operating system design most fundamentally enabled the move from monolithic execution to multi-user, multitasking environments, while also introducing process isolation?
මෙහෙයුම් පද්ධති නිර්මාණයේ කුමන සංක්‍රාන්තිය මොනොලිතික් ක්‍රියාත්මක කිරීමේ සිට බහු-පරිශීලක, බහුකාර්ය පරිසරයන් වෙත මාරුවීමට සහ ක්‍රියාවලි හුදකලාව හඳුන්වා දීමට මූලික වශයෙන් හැකියාව ලබා දුන්නේ ද?

1) The introduction of batch processing systems in early mainframes
මුල් කාලීන ප්‍රධාන රාමු වල කණ්ඩායම් සැකසුම් පද්ධති හඳුන්වාදීම
2) The use of command-line interfaces in Unix-based systems
යුනික්ස් පාදක පද්ධතිවල විධාන රේඛා අතුරුමුහුණත් භාවිතය
3) <u>The development of time-sharing systems with memory protection</u>
මතක ආරක්ෂාව සහිත කාල විභජන පද්ධති සංවර්ධනය
4) The implementation of graphical user interfaces in personal computers
පුද්ගලික පරිගණකවල චිත්‍රක පරිශීලක අතුරුමුහුණත් ක්‍රියාත්මක කිරීම
5) The rise of mobile operating systems with app sandboxing
යෙදුම් sandboxing සමඟ ජංගම මෙහෙයුම් පද්ධතිවල නැගීම

Answer : 3)

Explanation / පැහැදිලි කිරීම:

Option / විකල්පය 1:
Incorrect. Batch processing introduced job scheduling but lacked interactive use and real-time multitasking. All tasks ran in sequence with no concept of memory protection or isolation.
වැරදියි. කණ්ඩායම් සැකසුම් මඟින් රැකියා කාලසටහන්ගත කිරීම හඳුන්වා දුන් නමුත් අන්තර්ක්‍රියාකාරී භාවිතය සහ තත්‍ය කාලීන බහුකාර්ය නොමැති විය. සියලුම කාර්යයන් මතක ආරක්ෂාව හෝ හුදකලාව පිළිබඳ සංකල්පයකින් තොරව අනුපිළිවෙලින් ක්‍රියාත්මක විය.

Option / විකල්පය 2:
Incorrect. Command-line interfaces improved user interaction but did not structurally change how the OS managed processes or memory. It was an interface innovation, not a core architecture shift.
වැරදියි. විධාන රේඛා අතුරුමුහුණත් පරිශීලක අන්තර්ක්‍රියා වැඩිදියුණු කළ නමුත් මෙහෙයුම් පද්ධතිය ක්‍රියාවලි හෝ මතකය කළමනාකරණය කරන ආකාරය ව්‍යුහාත්මකව වෙනස් කළේ නැත. එය අතුරුමුහුණත් නවෝත්පාදනයක් වූ අතර මූලික ගෘහ නිර්මාණ මාරුවක් නොවේ.

Option / විකල්පය 3:
Correct. Time-sharing systems allowed multiple users to run programs seemingly simultaneously. Combined with memory protection, these systems enforced process isolation, ensuring one process could not interfere with another. This was a foundational step toward modern multitasking operating systems.
නිවැරදි. කාල බෙදාගැනීමේ පද්ධති බහු පරිශීලකයින්ට එකවර වැඩසටහන් ක්‍රියාත්මක කිරීමට ඉඩ දුන්නේය. මතක ආරක්ෂාව සමඟ ඒකාබද්ධව, මෙම පද්ධති ක්‍රියාවලි හුදකලාව බලාත්මක කළ අතර, එක් ක්‍රියාවලියකට තවත් ක්‍රියාවලියකට බාධා කළ නොහැකි බව සහතික කළේය. මෙය නවීන බහුකාර්ය මෙහෙයුම් පද්ධති සඳහා මූලික පියවරක් විය.

Option / විකල්පය 4:
Incorrect. Graphical user interfaces improved usability but were built on top of systems that already supported multitasking. GUIs were a user-level improvement, not an architectural leap.
වැරදියි. චිත්‍රක පරිශීලක අතුරුමුහුණත් භාවිතය වැඩි දියුණු කළ නමුත් දැනටමත් බහුකාර්ය සඳහා සහය දැක්වූ පද්ධති මත ගොඩනගා ඇත. GUI යනු වාස්තු විද්‍යාත්මක පිම්මක් නොව පරිශීලක මට්ටමේ වැඩිදියුණු කිරීමකි.

Option / විකල්පය 5:
Incorrect. App sandboxing is a modern extension of process isolation principles. While important, it builds upon the core architectural concepts introduced in earlier time-sharing systems.
වැරදියි. යෙදුම් වැලිපිල්ල යනු ක්‍රියාවලි හුදකලා මූලධර්මවල නවීන දිගුවකි. වැදගත් වුවද, එය පෙර කාල බෙදාගැනීමේ පද්ධතිවල හඳුන්වා දුන් මූලික ගෘහ නිර්මාණ සංකල්ප මත ගොඩනගා ඇත.

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

5. What does a **CHECK** constraint ensure in a table column?
වගු තීරුවක **CHECK** සංරෝධකය සහතික කරන්නේ කුමක්ද?

```
CREATE TABLE employees (
            emp_id INT PRIMARY KEY,
            salary DECIMAL(10, 2) CHECK (salary >= 3000));
```

1) <u>Enforces `salary >= 3000`.</u>
2) Ignores salary values.
3) Sets salary to 3000.
4) Allows any salary.
5) Deletes rows with low salary.

Answer : 1)

A CHECK constraint in a SQL table column ensures that the values inserted into that column meet a specific condition. In the given example, the CHECK constraint CHECK (`salary >= 3000`) ensures that the salary column can only contain values that are greater than or equal to 3000. If a user tries to insert a value into the salary column that is less than `3000`, the database will raise an error, preventing the insertion.
SQL වගු තීරුවක ඇති CHECK සම්බාධකය මඟින් එම තීරුවට ඇතුළු කර ඇති අගයන් නිශ්චිත කොන්දේසියක් සපුරාලන බව සහතික කරයි. ලබා දී ඇති උදාහරණයේ, CHECK සම්බාධකය CHECK (`salary >= 3000`) වැටුප් තීරුවේ අඩංගු විය හැක්කේ `3000`ට වඩා වැඩි හෝ සමාන අගයන් පමණක් බව සහතික කරයි. පරිශීලකයෙකු `3000`ට අඩු වැටුප් තීරුවට අගයක් ඇතුළු කිරීමට උත්සාහ කරන්නේ නම්, දත්ත සමුදාය දෝෂයක් මතු කරයි, ඇතුළත් කිරීම වළක්වයි.

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

6. In the following Python code, what happens if battery_percentage is 15 and charger_connected() returns False?
පහත Python කේතයේ, battery_percentage 15ක් වන අතර charger_connected() මඟින් False ලබා දෙන්නේ නම් කුමක් සිදුවේද?

```
def charger_connected():
    return False
def start_charging():
    print("Starting charging...")
def stop_charging():
    print("Stopping charging.")
battery_percentage = int(input("Enter Battery Percentage: "))
while True:
    if battery_percentage < 20:
        if not charger_connected():
            start_charging()
        else:
            stop_charging()
    else:
        if charger_connected():
            stop_charging()
        else:
            print("Battery level sufficient.")
```

1) "Starting charging..." will be printed repeatedly.
"Starting charging..." නැවත නැවතත් මුද්‍රණය කෙරේ.
2) "Stopping charging." will be printed repeatedly.
"Stopping charging." නැවත නැවතත් මුද්‍රණය කෙරේ.
3) "Battery level sufficient." will be printed repeatedly.
"Battery level sufficient." නැවත නැවතත් මුද්‍රණය කෙරේ.
4) The program will terminate after the first input.
පළමු ආදානයෙන් පසු වැඩසටහන අවසන් වේ.
5) An error will occur due to an infinite loop.
අනන්ත ලූපයක් හේතුවෙන් දෝෂයක් සිදුවනු ඇත.

Answer : 1)

Explanation / පැහැදිලි කිරීම

Code Execution with battery_percentage = 15 and charger_connected() = False
battery_percentage = 15 and charger_connected() = False සමඟ කේත ක්‍රියාත්මක කිරීම

1. battery_percentage = int(input("Enter Battery Percentage: "))
It's specified that battery_percentage is set to 15.
battery_percentage 15 ලෙස සකසා ඇති බව සඳහන් කර ඇත.

2. while True:
The program enters an infinite loop, meaning the code inside this loop will run continuously unless explicitly broken out of.
වැඩසටහන අනන්ත ලූපයකට ඇතුළු වේ, එනම් මෙම ලූපය තුළ ඇති කේතය පැහැදිලිවම කැඩී නොගියහොත් අඛණ්ඩව ක්‍රියාත්මක වේ.

3. if battery_percentage < 20:
Since battery_percentage is 15, this condition (15 < 20) is True.
battery_percentage 15 වන බැවින්, මෙම කොන්දේසිය (15 < 20) සත්‍ය වේ.

4. if not charger_connected():
charger_connected() returns False. / charger_connected() False ලබා දෙයි.
not charger_connected() evaluates to not False, which is True.
not charger_connected() False, True බවට පත් කරයි.

5. start_charging():
Because the condition if not charger_connected(): is true, the start_charging() function is called. This will print:
if not charger_connected(): කොන්දේසිය සත්‍ය බැවින්, start_charging() ශ්‍රිතය කැඳවනු ලැබේ. මෙය පහත සඳහන් දෑ මුද්‍රණය කරනු ඇත:

```
Starting charging...
```

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

7. What will be the output of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් වේද?

```
def wow(a, b=4, c=[3]):
    x = a
    c.append(x)
    c.append(5)
    return c
print(wow(1,2))
print(wow(1,2))
```

1) [3,1,5]
[3,1,5]

2) [3,1,2,5]
[3,1,2,5,1,2,5]

3) [3,1,4,5]
[3,1,4,5]

4) [3,1,5]
[3,1,5,1,5]

5) Syntax Error

Answer : 4)

Explanation / පැහැදිලි කිරීම:

The function wow(a, b=4, c=[3]) has a default list c=[3]. Since lists are mutable in Python, any modifications to c will persist across multiple function calls.
wow(a, b=4, c=[3]) ශ්‍රිතයට පෙරනිමි ලැයිස්තුවක් c=[3] ඇත. පයිතන් හි ලැයිස්තු වෙනස් කළ හැකි බැවින්, c වෙත කරන ඕනෑම වෙනස් කිරීමක් ශ්‍රිත ඇමතුම් කිහිපයක් හරහා පවතිනු ඇත.

First call / පළමු කැඳවීම : wow(1,2)

Initial Values / ආරම්භක අගයන්:
a = 1
b = 2 (not used in the function) ශ්‍රිතයේ භාවිතා නොවේ
c = [3] (from default argument) පෙරනිමි තර්කයෙන්

Modifications to c / c සඳහා වෙනස් කිරීම්:
c.append(x) → c.append(1) → c = [3, 1]
c.append(5) → c = [3, 1, 5]
Return value = [3, 1, 5]

Second call / දෙවන කැඳවීම : `wow(1,2)`
Since c is a mutable default argument, it is not reset to `[3]`. Instead, it retains changes from the first call.
c යනු වෙනස් කළ හැකි පෙරනිමි තර්කයක් බැවින්, එය `[3]` වෙත නැවත සකසනු නොලැබේ. ඒ වෙනුවට, එය පළමු කැඳවීමේ වෙනස්කම් රඳවා ගනී.

The list `c` is now `[3, 1, 5]` before this call starts. So, it appends the values of the function and updates as `[3, 1, 5, 1, 5]`.
මෙම ඇමතුම ආරම්භ වීමට පෙර ලැයිස්තුව `c` දැන් `[3, 1, 5]` වේ. එබැවින්, එය ශ්‍රිතයේ අගයන් එකතු කර `[3, 1, 5, 1, 5]` ලෙස යාවත්කාලීන කරයි.

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

8. In an office equipped with a smart management system, which scenario is least likely to happen?
සුහුරු කළමනාකරණ පද්ධතියකින් සමන්විත කාර්යාලයක, සිදුවීමට අවම සම්භාවිතාවක් ඇත්තේ කුමක්ද?

   1) Employees receive automated notifications about meeting times and changes in schedules.
   රැස්වීම් වේලාවන් සහ කාලසටහන්වල වෙනස්කම් පිළිබඳව සේවකයින්ට ස්වයංක්‍රිය දැනුම්දීම.
   2) The system automatically adjusts room temperatures based on occupancy and preferences of employees.
   සේවකයින් සිටින ස්ථානය සහ කැමැත්ත මත පදනම්ව පද්ධතිය ස්වයංක්‍රියව කාමරයේ උෂ්ණත්වය සකස් කරයි.
   3) The smart lighting adjusts based on time of day and employee presence to enhance energy efficiency.
   බලශක්ති කාර්යක්ෂමතාව ඉහළ නැංවීම සඳහා දවසේ වේලාව සහ සේවක පැමිණීම මත පදනම්ව සුහුරු ආලෝකකරණය සැකසේ.
   4) <u>Employees must manually log their attendance using a digital app that requires periodic validation of their identity.</u>
   සේවකයින් ඔවුන්ගේ අනන්‍යතාවය වරින් වර වලංගු කිරීම අවශ්‍ය වන අංකිත යෙදුමක් භාවිතයෙන් ඔවුන්ගේ පැමිණීම හස්තමය අයුරින් ලොග් විය යුතුය.
   5) The system provides real-time analytics on workspace usage, helping management optimize layouts.
   පද්ධතිය මඟින් වැඩබිම් භාවිතය පිළිබඳ තත්‍ය කාලීන විශ්ලේෂණ සැපයීමෙන් කළමනාකරණයට පිරිසැලසුම් ප්‍රශස්ත කිරීමට උපකාරී වේ.

Answer: 4)

Explanation / පැහැදිලි කිරීම:

A smart system is a technology framework that enhances efficiency and convenience through automation, connectivity, and data analysis.
සුහුරු පද්ධතියක් යනු ස්වයංක්‍රියකරණය, සම්බන්ධතාවය සහ දත්ත විශ්ලේෂණය හරහා කාර්යක්ෂමතාව සහ පහසුව වැඩි දියුණු කරන තාක්ෂණික රාමුවකි.

Let's find the option that is least aligned with the principles of a smart management system.
සුහුරු කළමනාකරණ පද්ධතියක මූලධර්ම සමඟ අවම වශයෙන් සමපාත වන විකල්පය සොයා ගනිමු.

Option / විකල්පය 1)
This is likely to happen in a smart management system, as automation is a key feature for improving communication and efficiency.
සන්නිවේදනය සහ කාර්යක්ෂමතාව වැඩි දියුණු කිරීම සඳහා ස්වයංක්‍රියකරණය ප්‍රධාන අංගයක් වන බැවින් මෙය සුහුරු කළමනාකරණ පද්ධතියක සිදු විය හැකිය.

Option / විකල්පය 2)
This scenario is common in smart management systems, utilizing sensors and automation for environmental control.
පරිසර පාලනය සඳහා සංවේදක සහ ස්වයංක්‍රියකරණය භාවිතා කරන සුහුරු කළමනාකරණ පද්ධතිවල මෙම අවස්ථාව බහුලව දක්නට ලැබේ.

Option / විකල්පය 3)
This is also likely, as smart lighting systems often use sensors to adjust lighting for energy efficiency and comfort.
බලශක්ති කාර්යක්ෂමතාව සහ සුවපහසුව සඳහා ආලෝකය සකස් කිරීම සඳහා සුහුරු ආලෝක පද්ධති බොහෝ විට සංවේදක භාවිතා කරන බැවින් මෙය ද විය හැකිය.

Option / විකල්පය 4)
This is least likely in a smart management system, which typically focuses on automation and efficiency. A smart system would likely have automatic attendance tracking through sensors or biometrics rather than requiring manual logging.
සාමාන්‍යයෙන් ස්වයංක්‍රියකරණය සහ කාර්යක්ෂමතාව කෙරෙහි අවධානය යොමු කරන සුහුරු කළමනාකරණ පද්ධතියක මෙය සිදු වීමේ සම්භාවිතාවය අවම වේ. සුහුරු පද්ධතියකට හස්තමය අයුරින් ඇතුළුවීම අවශ්‍ය වනවාට වඩා සංවේදක හෝ ජෛවමිතික හරහා ස්වයංක්‍රිය ඇතුළුවීම සිදු විය හැකි ය.

Option / විකල්පය 5)
This scenario is highly probable, as smart systems often include analytics to improve workspace efficiency and management decisions.
සුහුරු පද්ධතිවල බොහෝ විට වැඩ අවකාශයේ කාර්යක්ෂමතාව සහ කළමනාකරණ තීරණ වැඩි දියුණු කිරීම සඳහා විශ්ලේෂණ ඇතුළත් වන බැවින්, මෙම අවස්ථාව සිදුවීමට. බොහෝ දුරට ඉඩ ඇත.

Therefore, the answer is 4).
එබැවින්, නිවැරදි පිළිතුර 4) 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) <u>This is a shorthand property in CSS / මෙය CSS හි shorthand ගුණාංගයකි.</u>
3) The above code will not change anything on the website.
ඉහත කේතය වෙබ් අඩවියේ කිසිවක් වෙනස් නොකරයි.
4) It is a CSS code / එය CSS කේතයකි.
5) This code will make the CSS file longer to load / මෙම කේතය CSS ගොනුව පූරණය කිරීමට දිගු කරයි.

Answer: 2)

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 2).
එබැවින්, නිවැරදි පිළිතුර වන්නේ 2) වේ.

10. Which of the following steps is NOT typically involved in the process of collecting data from an HTML form and inserting that data into a database?
HTML පෝරමයකින් දත්ත රැස්කර එම දත්ත දත්ත සමුදායකට ඇතුළත් කිරීමේ ක්‍රියාවලියට සාමාන්‍යයෙන් සම්බන්ධ නොවන්නේ පහත පියවරවලින් කවරක්ද?

1) Retrieving the data from the HTML form using PHP.
PHP භාවිතයෙන් HTML පෝරමයෙන් දත්ත ලබා ගැනීම.
2) Sending the data to the server using an HTTP request
HTTP ඉල්ලීමක් භාවිතයෙන් දත්ත සේවාදායකයට යැවීම.
3) Parsing and validating the data on the server side.
සේවාදායක අන්තයේ දත්ත විග්‍රහ කිරීම සහ වලංගු කිරීම.
4) Directly updating the database schema without modifying the table structure.
වගු ව්‍යුහය වෙනස් නොකර දත්ත සමුදා ක්‍රමය සෘජුවම යාවත්කාලීන කිරීම.
5) Inserting the validated data into the database using a server-side script (Ex. PHP, Node.js).
සේවාදායක අන්තයේ ස්ක්‍රිප්ට් (උදා. PHP" Node'js) භාවිතයෙන් දත්ත සමුදායට වලංගු දත්ත ඇතුළත් කිරීම.

Answer : 4)

Updating the database schema without modifying the table structure is not typically involved in the process of collecting data from an HTML form and inserting it into a database. The database schema defines the structure of tables, fields, and relationships, which is usually predefined during the design phase. When inserting data, the schema is not modified unless there's a need for structural changes like adding new columns or tables, which is unrelated to routine data collection.
වගු ව්‍යුහය වෙනස් නොකර දත්ත සමුදා ක්‍රමය යාවත්කාලීන කිරීම සාමාන්‍යයෙන් HTML පෝරමයකින් දත්ත රැස් කර දත්ත සමුදායකට ඇතුල් කිරීමේ ක්‍රියාවලියට සම්බන්ධ නොවේ. දත්ත සමුදා යෝජනා ක්‍රමය මඟින් වගු, ක්ෂේත්‍ර සහ සම්බන්ධතා වල ව්‍යුහය නිර්වචනය කරයි, එය සාමාන්‍යයෙන් සැලසුම් අවධියේදී පූර්ව නිර්වචනය කර ඇත. දත්ත ඇතුළු කරන විට, සාමාන්‍ය දත්ත රැස් කිරීම හා සම්බන්ධ නොවන, නව තීරු හෝ වගු එකතු කිරීම වැනි ව්‍යුහාත්මක වෙනස්කම් සඳහා අවශ්‍ය වන්නේ නම් මිස ක්‍රමලේඛනය වෙනස් නොවේ.

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

Other Options / වෙනත් විකල්ප:

Statement / ප්‍රකාශය 1
This is a common step in processing form data. PHP (or another server-side language) retrieves the form data using methods like `$_POST` or `$_GET`.
මෙය ආකෘති දත්ත සැකසීමේ පොදු පියවරකි. PHP (හෝ වෙනත් සේවාදායක පැත්තේ භාෂාවක් `$_POST` හෝ `$_GET` වැනි ක්‍රම භාවිතයෙන් පෝරම දත්ත ලබා ගනී.

Statement / ප්‍රකාශය 2
The data from the HTML form is typically sent to the server via an HTTP request (ex: POST or GET), which is a fundamental step in web form processing.
HTML පෝරමයේ දත්ත සාමාන්‍යයෙන් HTTP ඉල්ලීමක් හරහා සේවාදායකයට යවනු ලැබේ (උදා: POST හෝ GET) එය වෙබ් පෝරමය සැකසීමේ මූලික පියවරකි.

Statement / ප්‍රකාශය 3
Data validation and parsing ensure that the collected data is in the correct format and meets all required conditions before being inserted into the database.
දත්ත වලංගුකරණය සහ විග්‍රහ කිරීම මඟින් එකතු කරන ලද දත්ත නිවැරදි ආකෘතියෙන් පවතින බවත් දත්ත සමුදායට ඇතුළු කිරීමට පෙර අවශ්‍ය සියලු කොන්දේසි සපුරාලන බවත් සහතික කරයි.

Statement / ප්‍රකාශය 5
After validation, the data is inserted into the database using a server-side script. This is a critical step for ensuring that the data is stored securely and accurately.
සත්‍යාපනය කිරීමෙන් පසු, දත්ත සමුදාය තුළට server-side ස්ක්‍රිප්ට් එකක් භාවිතයෙන් ඇතුල් කරනු ලැබේ. දත්ත ආරක්ෂිතව සහ නිවැරදිව ගබඩා කර ඇති බව සහතික කිරීම සඳහා මෙය තීරණාත්මක පියවරකි.

1.(a) A student creates a simple Python program that allows a character to activate special powers using energy points.
ශිෂ්‍යයෙක් බලශක්ති ලකුණු (energy points) භාවිතා කරමින් චරිතයකට සුවිශේෂී බලයන් සක්‍රිය කිරීමට ඉඩ සලසන සරල පයිතන් වැඩසටහනක් නිර්මාණය කරයි.

```
powers = {
    "Fire": 50,
    "Speed": 30,
    "Strength": 40
}

energy = int(input("Enter energy: "))
chosen = input("Choose power: ")

if chosen in powers:
    if energy >= powers[chosen]:
        print("Power activated!")
        energy -= powers[chosen]
    else:
        print("Not enough energy!")
else:
    print("Invalid power")

print("Remaining energy:", energy)
```

i. What will be the output if the user enters:
පරිශීලකයා විසින් පහත දෑ ඇතුළත් කළහොත් ලැබෙන ප්‍රතිදානය කුමක්ද?

```
Enter energy: 60
Choose power: Fire
```

Answer / පිළිතුර:

```
Power activated!
Remaining energy: 10
```

ii. Explain what will happen if the user enters a power that is not in the dictionary.
පරිශීලකයා dictionary එකේ නොමැති බලයක් ඇතුළත් කළහොත් කුමක් සිදුවේදැයි පැහැදිලි කරන්න.

If the entered power is not in the dictionary, the condition becomes False and the program executes the `else` block, printing `"Invalid power"`. No energy is deducted, so the remaining energy stays the same.
ඇතුළත් කළ බලය ශබ්දකෝෂයේ නොමැති නම්, කොන්දේසිය අසත්‍ය (False) වන අතර වැඩසටහන `else` කොටස ක්‍රියාත්මක කර `"Invalid power"` යන්න මුද්‍රණය කරයි. බලශක්තිය අඩු නොවන බැවින්, ඉතිරි බලශක්ති ප්‍රමාණය නොවෙනස්ව පවතී.

Answer / පිළිතුර:

```
Invalid power
Remaining energy: <original energy>
```

iii. Modify the program so that the user can activate multiple powers continuously until they choose to stop. Write the improved code.
පරිශීලකයාට තමන් නවත්වන තෙක් අඛණ්ඩව බලයන් කිහිපයක් සක්‍රිය කළ හැකි වන පරිදි වැඩසටහන වෙනස් කරන්න. වැඩිදියුණු කළ කේතය ලියන්න.

```
while True:
    chosen = input("Choose power (or type 'stop' to exit): ")
    if chosen == "stop":
        break
```

(b) A school library system is being developed to manage book records. The following Python program is used.
පොත් වාර්තා කළමනාකරණය කිරීම සඳහා පාසල් පුස්තකාල පද්ධතියක් සංවර්ධනය වෙමින් පවතී. පහත පයිතන් වැඩසටහන භාවිතා වේ.

Fill the five blanks (----------------------) of the following Python code which is written to count the number of vowels in a given word and print "Valid" if the count is greater than 2.
ලබා දී ඇති වචනයක ඇති ස්වර (vowels) ගණන ගණනය කර, එම සංඛ්‍යාව 2 ට වඩා වැඩි නම් "Valid" ලෙස මුද්‍රණය කිරීම සඳහා ලියා ඇති පහත පයිතන් කේතයේ හිස්තැන් පහ (----------------------) පුරවන්න.

Note: Vowels are: a, e, i, o, u (both lowercase and uppercase).
සටහන: ස්වර යනු: a, e, i, o, u (Simple අකුරු සහ capital අකුරු දෙකම).

```
text = input("Enter a word: ")
count = 0
i = 0

while i _______(A)______ len(text):
    if text[i].lower() _______(B)______ ['a','e','i','o','u']:
        ______(C)_______
    i += 1

if count ______(D)_______ 2:
    print("Valid")
else:
    print("Invalid")
```

This program reads a word from the user and counts how many vowels it contains using a `while` loop. Each character is converted to lowercase and checked against a list of vowels, and the count is increased when a match is found. After scanning the whole word, it checks whether the vowel count is greater than 2. If so, it prints "Valid"; otherwise, it prints "Invalid".
මෙම වැඩසටහන මඟින් පරිශීලකයාගෙන් වචනයක් කියවා while ලූපයක් භාවිතයෙන් එහි අඩංගු ස්වර ගණන ගණනය කරයි. එක් එක් අක්ෂරය කුඩා අකුරුවලට (lowercase) පරිවර්තනය කර ස්වර ලැයිස්තුවක් සමඟ පරීක්ෂා කරන අතර, ගැලපීමක් හමු වූ විට ගණනය (count) වැඩි කරයි. මුළු වචනයම පරීක්ෂා කිරීමෙන් පසුව, ස්වර සංඛ්‍යාව 2 ට වඩා වැඩි දැයි පරීක්ෂා කරයි. එසේ නම්, එය "Valid" ලෙස මුද්‍රණය කරයි; එසේ නොමැති නම් "Invalid" ලෙස මුද්‍රණය කරයි.

Answers / පිළිතුරු:

A = `<`
B = `in`
C = `count += 1`
D = `>`