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 493 MARKING

2026/04/17
ictfromabc
Ravindu Bandaranayake

1. What is the main role of the system call interface in an operating system?
මෙහෙයුම් පද්ධතියක පද්ධති ඇමතුම් අතුරුමුහුණතෙහි ප්‍රධාන කාර්යභාරය කුමක්ද?

1) To provide an interface for user applications to interact with the hardware
පරිශීලක යෙදුම් සඳහා දෘඩාංග සමඟ අන්තර්ක්‍රියා කිරීමට අතුරු මුහුණතක් සැපයීමට
2) To manage network connections / ජාල සම්බන්ධතා කළමනාකරණය කිරීමට
3) To allocate memory to processes / ක්‍රියාවලි සඳහා මතකය වෙන් කිරීමට
4) To handle file system operations / ගොනු පද්ධති මෙහෙයුම් හැසිරවීමට
5) <u>To provide an interface for user applications to request services from the kernel</u>
කර්නලයෙන් සේවා ඉල්ලීමට පරිශීලක යෙදුම් සඳහා අතුරු මුහුණතක් සැපයීමට

Answer : 5)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
User applications don’t interact directly with hardware; they interact with the operating system. The system call interface provides an intermediary layer to request services indirectly, rather than directly managing hardware.
පරිශීලක යෙදුම් දෘඩාංග සමඟ සෘජුව අන්තර්ක්‍රියා නොකරයි, ඔවුන් මෙහෙයුම් පද්ධතිය සමඟ අන්තර් ක්‍රියා කරයි. පද්ධති ඇමතුම් අතුරුමුහුණත සෘජුවම දෘඩාංග කළමනාකරණය කරනවාට වඩා වක්‍රව සේවා ඉල්ලීමට අතරමැදි ස්ථරයක් සපයයි.

Option / විකල්ප 2)
Managing network connections is one of the tasks the operating system kernel can handle, but this is not the primary role of the system call interface. Network management is only one of many services that can be requested through system calls.
ජාල සම්බන්ධතා කළමනාකරණය කිරීම මෙහෙයුම් පද්ධති කර්නලයට හැසිරවිය හැකි එක් කාර්යයකි, නමුත් මෙය පද්ධති ඇමතුම් අතුරුමුහුණතේ මූලික කාර්යභාරය නොවේ. ජාල කළමනාකරණය පද්ධති ඇමතුම් හරහා ඉල්ලා සිටිය හැකි බොහෝ සේවාවන්ගෙන් එකක් පමණි.

Option / විකල්ප 3)
Memory allocation is another specific task the operating system manages, but again, it’s just one of many functions requested through system calls. The system call interface itself doesn’t manage memory; it only provides a way for applications to request memory allocation.
මතකය වෙන් කිරීම මෙහෙයුම් පද්ධතිය කළමනාකරණය කරන තවත් විශේෂිත කාර්යයකි, නමුත් නැවතත්, එය පද්ධති ඇමතුම් හරහා ඉල්ලා සිටින බොහෝ කාර්යයන්ගෙන් එකකි. පද්ධති ඇමතුම් අතුරුමුහුණත මතකය කළමනාකරණය නොකරයි, එය යෙදුම් සඳහා මතකය වෙන් කිරීම ඉල්ලා සිටීමට මාර්ගයක් පමණක් සපයයි.

Option / විකල්ප 4)
File system operations, like opening or reading files, are indeed managed through system calls. However, handling files is only one part of what system calls enable; it is not the main purpose of the system call interface.
ගොනු විවෘත කිරීම හෝ කියවීම වැනි ගොනු පද්ධති මෙහෙයුම් ඇත්ත වශයෙන්ම පද්ධති ඇමතුම් හරහා කළමනාකරණය කෙරේ. කෙසේ වෙතත්, ගොනු හැසිරවීම පද්ධති ඇමතුම් සක්‍රිය කරන දෙයින් එක් කොටසක් පමණි, එය පද්ධති ඇමතුම් අතුරුමුහුණතේ ප්‍රධාන අරමුණ නොවේ.

Option / විකල්ප 5)
The primary role of the system call interface is to allow user applications to request services from the kernel, enabling them to perform operations like file management, memory allocation, and network connections through system calls.
පද්ධති ඇමතුම් අතුරුමුහුණතෙහි මූලික කාර්යභාරය වනුයේ පරිශීලක යෙදුම්වලට කර්නලයෙන් සේවා ඉල්ලීමට ඉඩ දීම, පද්ධති ඇමතුම් හරහා ගොනු කළමනාකරණය, මතකය වෙන් කිරීම සහ ජාල සම්බන්ධතා වැනි මෙහෙයුම් සිදු කිරීමට ඔවුන්ට හැකියාව ලබා දීමයි.

Therefore, the correct answer is option 5), To provide an interface for user applications to request services from the kernel.
එබැවින්, නිවැරදි පිළිතුර විකල්පය 5), කර්නලයෙන් සේවා ඉල්ලීම සඳහා පරිශීලක යෙදුම් සඳහා අතුරුමුහුණතක් සැපයීම වේ.

2. Which of the following advancements is most likely to define sixth-generation computers?
පහත සඳහන් දියුණුව අතුරින් හයවන පරම්පරාවේ පරිගණක නිර්වචනය කිරීමට වඩාත්ම ඉඩ ඇත්තේ කුමක්ද?

1) Full integration of quantum computing and artificial intelligence
quantum පරිගණනය සහ කෘතිම බුද්ධිය පූර්ණ ලෙස ඒකාබද්ධ කිරීම
2) Use of nanotechnology for ultra-fast processing
අතිශය වේගවත් සැකසුම් සඳහා නැනෝ තාක්ෂණය භාවිතය
3) Development of bio-computing and DNA-based storage
ජෛව පරිගණකකරණය සහ DNA පාදක ගබඩා කිරීම සංවර්ධනය කිරීම
4) Enhanced neural networks mimicking human brain functionality
මිනිස් මොළයේ ක්‍රියාකාරිත්වය අනුකරණය කරන වැඩිදියුණු කළ ස්නායුක ජාල
5) Complete elimination of cloud computing in favor of local processing
ස්ථානීය සැකසුම් සඳහා පක්ෂව වලාකුළු පරිගණකකරණය සම්පූර්ණයෙන්ම ඉවත් කිරීම.

Answer: 1)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්පය 1)
This is correct. The sixth generation is expected to be dominated by quantum computing and AI advancements. Current trends indicate that this combination is the most probable direction for the next generation of computers.
මෙය නිවැරදියි. හයවන පරම්පරාව quantum පරිගණකකරණය සහ AI දියුණුවෙන් ආධිපත්‍යය දරනු ඇතැයි අපේක්ෂා කෙරේ. වත්මන් ප්‍රවණතා පෙන්නුම් කරන්නේ මෙම සංයෝජනය ඊළඟ පරම්පරාවේ පරිගණක සඳහා වඩාත්ම සම්භාවිතාවක් ඇති දිශාව බවයි.

Option / විකල්පය 2)
This is possible, but not primary. Nanotechnology will likely play a role in enhancing hardware performance and reducing power consumption. But it is not the defining feature of the sixth generation.
මෙය විය හැකිය, නමුත් ප්‍රධාන නොවේ. දෘඩාංග කාර්ය සාධනය වැඩි දියුණු කිරීම සහ බලශක්ති පරිභෝජනය අඩු කිරීම සඳහා නැනෝ තාක්ෂණය බොහෝ විට කාර්යභාරයක් ඉටු කරනු ඇත. නමුත් එය හයවන පරම්පරාවේ නිර්වචන ලක්ෂණය නොවේ.

Option / විකල්පය 3)
This is possible future development, but not exclusive to sixth generation. Bio-computing uses biological molecules for computing instead of traditional silicon-based chips. DNA storage has massive potential for long-term, high-density data storage. However, this field is still in early research stages and may take longer to become a mainstream technology.
මෙය අනාගත සංවර්ධනයක් සිදුවිය හැකි නමුත් හයවන පරම්පරාවට සීමා නොවේ. ජෛව පරිගණකකරණයේදී සාම්ප්‍රදායික සිලිකන් පාදක චිප්ස් වෙනුවට ජීව විද්‍යාත්මක අණු පරිගණකකරණය සඳහා භාවිතා කරයි. ෘභ් ගබඩා කිරීම තුළ දිගු කාලීන, ඉහළ ඝනත්ව දත්ත ගබඩා කිරීම සඳහා දැවැන්ත විභවයක් ඇත. කෙසේ වෙතත්, මෙම ක්ෂේත්‍රය තවමත් මුල් පර්යේෂණ අවධියේ පවතින අතර ප්‍රධාන ධාරාවේ තාක්ෂණයක් බවට පත්වීමට වැඩි කාලයක් ගතවනු ඇත.

Option / විකල්පය 4)
This is not the defining feature. Neural networks and brain-inspired computing (neuromorphic computing) will continue to evolve. However, this is a subset of AI advancements, making it an important but not exclusive characteristic of the sixth generation.
මෙය නිර්වචන ලක්ෂණය නොවේ. ස්නායුක ජාල සහ මොළයේ ආභාෂය ලත් පරිගණකකරණය (ස්නායුරූපී පරිගණකකරණය) අඛණ්ඩව පරිණාමය වනු ඇත. කෙසේ වෙතත්, මෙය AI දියුණුවේ උප කුලකයකි, එය හයවන පරම්පරාවේ වැදගත් නමුත් සුවිශේෂී නොවන ලක්ෂණයක් බවට පත් කරයි.

Option / විකල්පය 5)
Cloud computing is evolving, not disappearing.
වලාකුළු පරිගණකකරණය පවතින්නේ පරිණාමය වෙමිනි, අතුරුදහන් වෙමින් නොවේ.

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

3. Which of the following is an invalid Boolean law?
පහත සඳහන් ඒවායින් අවලංගු බූලියන් නීතියක් කුමක්ද?

1) $A.A = A$
2) $\overline{(A.B)} = \overline{A} + \overline{B}$
3) $\underline{A.(A + B) = B}$
4) $(A + B) + C = A + (B + C)$
5) $A + (B.C) = (A + B).(A + C)$

Answer : 3)

Explanation / පැහැදිලි කිරීම

According to Redundancy Law, $A.(A + B) = A$
සමතිරික්ත න්‍යායට අනුව, $A.(A + B) = A$

Option / විකල්පය 1)
This proves the Idempotent Law
මෙය තදේවභාවී න්‍යාය සනාථ කරයි

Option / විකල්පය 2)
This proves the De Morgan's Law
මෙය ඩි මෝගන්ගේ නීතිය සනාථ කරයි

Option / විකල්පය 4)
This proves the Associative Law
මෙය සංඝටන න්‍යාය සනාථ කරයි

Option / විකල්පය 5)
This proves the Distributive Law
මෙය විඝටන න්‍යාය සනාථ කරයි

Therefore, the correct answer is option 3).
එබැවින් නිවැරදි පිළිතුර විකල්පය 3) 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 ක්‍රියාවලින් RAM සහ තැටිය අතර ගෙන යයි, ස්ථාවර බ්ලොක් 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 only matching rows from both tables / වගු දෙකෙන්ම ගැලපෙන පේළි පමණක් සම්බන්ධ වීමට
2) To join all rows from both tables / වගු දෙකෙන්ම සියලුම පේළි එක් කිරීමට
3) To join only unique rows from both tables / වගු දෙකෙන්ම අනන්‍ය 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: 1)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
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 ගැළපෙන අගයන් ඇති පේළි පමණි. ගැළපීමක් නොමැති නම්, එම පේළි ප්‍රතිඵල කට්ටලයෙන් බැහැර කරනු ලැබේ.

Option / විකල්ප 2)
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 3)
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 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 කථා කරයි, එය DISTINCT මූල පදයෙන් සිදු කරනු ලැබේ, වගු එකතු කිරීමෙන් නොවේ.

So, the correct answer number is 1).
එබැවින්, නිවැරදි පිළිතුරු අංකය 1) වේ.

6. What will be the output of the following code?
පහත කේතයේ ප්‍රතිදානය කුමක් වේද?

```
x = 3
y = 10
if x < 5:
    if y > 5:
        if x + y > 15:
            print("Case 1")
        else:
            print("Case 2")
    else:
        print("Case 3")
else:
    print("Case 4")
```

1) Case 1 2) <u>Case 2</u> 3) Case 3 4) Case 4 5) Error

Answer : 2)

Explanation/ පැහැදිලි කිරීම,

The given code checks a series of conditions using nested if statements. Initially, x = 3 and y = 10. The first condition x < 5 is true because 3 is less than 5. Inside this block, the next condition y > 5 is also true because 10 is greater than 5. Finally, the code evaluates x + y > 15, which translates to 3 + 10 > 15. Since this condition is false (13 is not greater than 15), the else block of this condition is executed, printing "Case 2". Thus, the output of the program is Case 2.
දී ඇති කේතය nested if ප්‍රකාශ භාවිතා කරමින් කොන්දේසි මාලාවක් පරීක්ෂා කරයි. මුලදී, x = 3 සහ y = 10 වේ. පළමු කොන්දේසිය x < 5 සත්‍ය වේ, මන්ද 3, 5 ට වඩා අඩුය. මෙම කොටස තුළ, ඊළඟ කොන්දේසිය y > 5 ද සත්‍ය වේ, මන්ද 10 5 ට වඩා වැඩිය. අවසාන වශයෙන්, කේතය x + y > 15 ඇගයීමට ලක් කරයි, එය 3 + 10 > 15 ලෙස පරිවර්තනය වේ. මෙම කොන්දේසිය අසත්‍ය බැවින් (13, 15 ට වඩා වැඩි නොවේ), මෙම කොන්දේසියේ else කොටස ක්‍රියාත්මක කර, "Case 2" මුද්‍රණය කරයි. මේ අනුව, වැඩසටහනේ ප්‍රතිදානය "Case 2" වේ.

7. In the magical world of Pythonia, a wizard stores his spells in a file called spellbook.txt. Each line in the file contains a spell name and its power level separated by a colon (:). The wizard wants to find the most powerful spell in his spellbook. Complete the following code to help him achieve this.
Pythonia හි මායාකාරී ලෝකයේ, මායාකාරයෙක් තම මන්ත්‍ර ගබඩා කරන්නේ spellbook.txt නම් ගොනුවක ය. ගොනුවේ සෑම පේළියකම මන්ත්‍ර නාමයක් සහ එහි බල මට්ටම colon (:) වෙන් කර ඇත. මායාකරයාට ඔහුගේ මන්ත්‍ර පොතේ ඇති බලවත්ම මන්ත්‍රය සොයා ගැනීමට අවශ්‍ය වේ. මෙය සාක්ෂාත් කර ගැනීමට ඔහුට උපකාර කිරීම සඳහා පහත කේතය සම්පූර්ණ කරන්න.

```
most_powerful_spell = ""
max_power = __a)___
with open("spellbook.txt", __b)___) as file:
    for line in file:
        spell, power = line.strip().split(":")
        power = int(power)
        if power > __c)___:
            max_power = __d)___
            most_powerful_spell = __e)___
print(f"The most powerful spell is: {most_powerful_spell} with power {max_power}")
```

	a)	b)	c)	d)	e)
1)	0	"r"	max_power	power	spell
2)	-1	"w"	power	max_power	power
3)	None	"a""	spell	spell	max_power
4)	100	"x"	line	line	line
5)	-1	"x"	max_power	power	line

Answer : 1)

Explanation/ පැහැදිලි කිරීමල

Initialization: / ආරම්භ කිරීම:
"max_power" is set to 0 to track the highest power level. "most_powerful_spell" is an empty string to store the name of the most powerful spell.
ඉහළම බල මට්ටම නිරීක්ෂණය කිරීම සඳහා max_power, 0 ලෙස සකසා ඇත. "most_powerful_spell" යනු වඩාත්ම බලගතු මන්ත්‍රයේ නම ගබඩා කිරීම සඳහා වූ empty string එකකි.

File Operations: / ගොනු මෙහෙයුම්:
The file is opened in read mode ("r"). A for loop reads each line in the file.
ගොනුව කියවීමේ ආකාරයෙන් විවෘත කර ඇත ("r"). for ලූපය ගොනුවේ සෑම පේළියක්ම කියවයි.

Processing Each Line: / සෑම පේළියක්ම සැකසීම:
Each line is split into spell (name) and power (level) using split(":"). The power is converted to an integer for comparison.
split(":") භාවිතා කර සෑම පේළියක්ම, මන්ත්‍රය (නම) සහ බලය (මට්ටම) ලෙස බෙදා ඇත. සංසන්දනය සඳහා බලය පූර්ණ සංඛ්‍යාවක් බවට පරිවර්තනය වේ.

Finding the Most Powerful Spell: / වඩාත්ම බලවත් මන්තරය සොයා ගැනීම:
If the current spell's power is greater than max_power, update max_power and most_powerful_spell.
වත්මන් මන්ත්‍රයේ බලය "max_power" වඩා වැඩි නම්, "max_power" සහ "most_powerful_spell" යාවත්කාලීන කරයි.

Output: / ප්‍රතිදානය:
After processing all lines, the program prints the most powerful spell and its power level.
සියලුම පේළි සැකසීමෙන් පසු, වැඩසටහන වඩාත්ම බලගතු මන්ත්‍රය සහ එහි බල මට්ටම මුද්‍රණය කරයි.

Therefore, the answer is, 1).
එබැවින් පිළිතුර වන්නේ, 1) වේ.

8. Which type of system is best characterized by its ability to self-regulate and maintain stability?
ස්වයං-නියාමනය කිරීමට සහ ස්ථාවරත්වය පවත්වා ගැනීමට ඇති හැකියාව මගින් වඩාත් හොඳින් සංලක්ෂිත වන්නේ කුමන ආකාරයේ පද්ධතිය ද?

1) Open systems / විවෘත පද්ධති
2) Closed systems / සංවෘත පද්ධති
3) Living systems / ජීවන පද්ධති
4) Physical systems / භෞතික පද්ධති
5) Manmade systems / මිනිසා විසින් සාදන ලද පද්ධත

Answer : 3)

The correct answer is 3) Living systems.
නිවැරදි පිළිතුර 3) ජීවන පද්ධති වේ.

Living systems, like organisms, are best characterized by their ability to self-regulate and maintain stability through homeostasis. They can adjust their internal processes to adapt to changes in the environment, ensuring that conditions within the system remain stable. This self-regulating ability is essential for survival and is a defining feature of living systems.
ජීවීන් වැනි ජීව පද්ධති, හෝමියස්ටැසිස් හරහා ස්වයං-නියාමනය කිරීමට සහ ස්ථාවරත්වය පවත්වා ගැනීමට ඇති හැකියාව මගින් වඩාත් හොඳින් සංලක්ෂිත වේ. පරිසරයේ වෙනස්කම් වලට අනුවර්තනය වීමට ඔවුන්ගේ අභ්‍යන්තර ක්‍රියාවලින් සකස් කළ හැකි අතර, පද්ධතිය තුළ ඇති තත්වයන් ස්ථාවරව පවතින බව සහතික කරයි. මෙම ස්වයං-නියාමන හැකියාව පැවැත්ම සඳහා අත්‍යවශ්‍ය වන අතර එය ජීව පද්ධතිවල නිර්වචන ලක්ෂණයකි.

Explanation of other options / වෙනත් විකල්ප පැහැදිලි කිරීම:

Option / විකල්පය 1)
Open systems interact with their environment by exchanging energy, matter, or information. While they can adapt to external changes, they don't have the same intrinsic self-regulation mechanisms as living systems. They rely on external inputs to maintain their stability.
විවෘත පද්ධති බලශක්තිය, පදාර්ථය හෝ තොරතුරු හුවමාරු කර ගැනීමෙන් ඔවුන්ගේ පරිසරය සමඟ අන්තර් ක්‍රියා කරයි. බාහිර වෙනස්කම් වලට අනුවර්තනය වීමට ඔවුන්ට හැකි වුවද, ජීවන පද්ධතිවලට සමාන සහජ ස්වයං-නියාමක යාන්ත්‍රණ නොමැත. ඔවුන්ගේ ස්ථාවරත්වය පවත්වා ගැනීම සඳහා ඔවුන් බාහිර යෙදවුම් මත රඳා පවතී.

Option / විකල්පය 2)
Closed systems are isolated from their environment and do not exchange energy or matter with their surroundings. They lack the ability to self-regulate and maintain stability over time, as they cannot adapt to changes in external conditions.
සංවෘත පද්ධති ඒවායේ පරිසරයෙන් හුදකලා වන අතර ඒවායේ වටපිටාව සමඟ ශක්තිය හෝ පදාර්ථ හුවමාරු නොකරයි. බාහිර තත්වයන්හි වෙනස්කම් වලට අනුවර්තනය වීමට ඔවුන්ට නොහැකි බැවින්, කාලයත් සමඟම ස්වයං-නියාමනය කිරීමට සහ ස්ථාවරත්වය පවත්වා ගැනීමට ඔවුන්ට හැකියාවක් නැත.

Option / විකල්පය 4)
Physical systems refer to non-living systems, such as mechanical or chemical systems. While they may exhibit stability under certain conditions, they do not possess the complex self-regulation abilities seen in living systems.
භෞතික පද්ධති යනු යාන්ත්‍රික හෝ රසායනික පද්ධති වැනි අජීවී පද්ධති වෙත යොම වන පද්ධතු වේ. ඔවුන් යම් යම් තත්වයන් යටතේ ස්ථාවරත්වය ප්‍රදර්ශනය කළ හැකි වුවද, ජීවන පද්ධතිවල දක්නට ලැබෙන සංකීර්ණ ස්වයං-නියාමනය කිරීමේ හැකියාවන් ඔවුන් සතුව නොමැත.

Option / විකල්පය 5)
Manmade systems, like machines or software, can be designed to maintain stability through control mechanisms, but they do not inherently self-regulate in the same way living systems do. Their stability is typically a result of human design rather than an intrinsic characteristic.
යන්ත්‍ර හෝ මෘදුකාංග වැනි මිනිසා විසින් සාදන ලද පද්ධති, පාලන යාන්ත්‍රණ හරහා ස්ථායිතාවය පවත්වා ගැනීමට සැලසුම් කළ හැකි නමුත්, ඒවා ජීව පද්ධති කරන ආකාරයටම සහජයෙන්ම ස්වයං-නියාමනය නොකරයි. ඔවුන්ගේ ස්ථාවරත්වය සාමාන්‍යයෙන් ආවේනික ලක්ෂණයකට වඩා මානව නිර්මාණයේ ප්‍රතිඵලයකි.

9. Which of the following is the correct syntax to change the background-color of all paragraph elements to yellow?
සියලුම ජේද මුලාංගවල පසුබිම් වර්ණය කහ පැහැයට වෙනස් කිරීම සඳහා පහත දැක්වෙන නිවැරදි කේත ඛණ්ඩය කුමක්ද?
   1) `p{background-color:#yellow;}`
   2) `p{background-color:yellow;}`
   3) `all{background-color:yellow;}`
   4) `p{background-color:*yellow;}`
   5) `p{bgcolor:yellow;}`

Answer: 2)

The correct syntax to change the background color of all paragraph elements to yellow is `p{background-color:yellow;}`.
සියලුම ජේද මුලාංගවල පසුබිම් වර්ණය කහ පැහැයට වෙනස් කිරීමට නිවැරදි වාක්‍ය ඛණ්ඩය `p{background-color:yellow;}`.

Explanation / පැහැදිලි කිරීම:

In CSS, you use a selector to target the elements you want to style, in this case, the `p` selector targets all paragraph elements. The `background-color` property sets the background color, and yellow is the color value. The other options are incorrect due to syntax errors or incorrect properties. For instance, `#yellow` is not a valid color code, `all` is not a CSS selector, `*yellow` is not a valid color value, and `bgcolor` is an outdated HTML attribute rather than a CSS property.
CSSහි, ඔබට මෝස්තර කිරීමට අවශ්‍ය මූලාංගය ඉලක්ක කිරීමට ඔබ තේරීම්කාරකයක් භාවිතා කරයි, මෙම අවස්ථාවේදී, `p` තේරීම් කාරකය සියලුම ජේද මූලාංගය ඉලක්ක කරයි. `background-color` ගුණාංගය පසුබිම් වර්ණය සකසයි, සහ `yellow` යනු වර්ණ අගයයි. වාක්‍ය ඛණ්ඩ දෝෂ හෝ වැරදි ගුණාංග හේතුවෙන් අනෙකුත් විකල්ප වැරදිය. උදාහරණයක් ලෙස, `#yellow` යනු වලංගු වර්ණ කේතයක් නොවේ, `all` CSS තේරීම්කාරකයක් නොවේ, `*yellow` යනු වලංගු වර්ණ අගයක් නොවේ, සහ `bgcolor` යනු CSS ගුණාංගයකට වඩා යල් පැන ගිය HTML ගුණාංගයකි.

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

10. What is the purpose of the action attribute in an HTML form?
HTML ආකෘතියක action උපලක්ෂණයේ අරමුණ කුමක්ද?
   1) It specifies the form's target audience
      එය පෝරමයේ ඉලක්කගත ප්‍රේක්ෂකයින් නියම කරයි.
   2) It defines the CSS styles to apply to the form
      එය පෝරමයට යෙදිය යුතු CSS styles නිර්වචනය කරයි.
   3) It sets the URL where the form data should be submitted'
      එය පෝරම දත්ත ඉදිරිපත් කළ යුතු URL එක සකසයි.
   4) It sets the data validation rules
      එය දත්ත වලංගු කිරීමේ නීති සකසයි.
   5) It defines the programming language used in the form.
      එය පෝරමයේ භාවිතා කරන ක්‍රමලේඛන භාෂාව නිර්වචනය කරයි.

Answer: 3)

Explanation / පැහැදිලි කිරීම:

The action attribute in an HTML form defines the URL where the form data will be sent when the user submits the form. It points to a server-side script or application that processes the submitted data, such as saving it in a database or sending it via email.
HTML පෝරමයක ඇති action උපලක්ෂණය පරිශීලකයා පෝරමය ඉදිරිපත් කරන විට පෝරම දත්ත යවනු ලබන ඔසෑබ නිර්වචනය කරයි. එය දත්ත සමුදායක සුරැකීම හෝ විද්‍යුත් තැපෑලෙන් යැවීම වැනි ඉදිරිපත් කරන ලද දත්ත සකසන සේවාදායක පාර්ශවීය ස්ක්‍රිප්ට් හෝ යෙදුමක් වෙත යොමු කරයි.

- When the form is submitted, the browser sends the form data to this specified URL using the method defined (usually GET or POST).
  පෝරමය ඉදිරිපත් කළ විට, වෙබ් අතිරික්සුව විසින් නිර්වචනය කර ඇති (සාමාන්‍යයෙන් ධෑඹ හෝ ජඩිඹ) ක්‍රමය භාවිතා කරමින් මෙම නිශ්චිත ඔසෘබ වෙත පෝරම දත්ත යවයි.
- If the action attribute is empty or omitted, the form data is submitted to the current page URL. Action උපලක්ෂණ හිස් නම් හෝ මඟ හැර තිබේ නම්, පෝරම දත්ත වත්මන් පිටුවේ ඔසෘබ වෙත ඉදිරිපත් කෙරේ.

The correct answer is 3) It sets the URL where the form data should be submitted.
නිවැරදි පිළිතුර 3) එය පෝරම දත්ත ඉදිරිපත් කළ යුතු ඔසෘබ එක සකසයි.