සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved]

Ravindu Bandaranayake - ictfromabc

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 426 MARKING

2026/02/09
ictfromabc
Ravindu Bandaranayake

1. Cache memory is typically used to store which of the following?
නිහිත මතකය සාමාන්‍යයෙන් භාවිතා කරනුයේ පහතින් කුමක් ගබඩා කිරීම සඳහාද?

1) To permanently store the most frequently accessed data.
වැඩිම නිරන්තරයෙන් ප්‍රවේශ කරනු ලබන දත්ත ස්ථිරව ගබඩා කිරීමට.
2) To temporarily store the most frequently accessed data.
වැඩිම නිරන්තරයෙන් ප්‍රවේශ කරනු ලබන දත්ත තාවකාලිකව ගබඩා කිරීමට.
3) To temporarily store large volumes of data.
විශාල දත්ත පරිමාවක් තාවකාලිකව ගබඩා කිරීමට.
4) To permanently store the least frequently accessed data.
අවම නිරන්තරයෙන් ප්‍රවේශ කරනු ලබන දත්ත ස්ථිරව ගබඩා කිරීමට.
5) To temporarily store the least frequently accessed data.
අවම නිරන්තරයෙන් ප්‍රවේශ කරනු ලබන දත්ත තාවකාලිකව ගබඩා කිරීමට.

Answer : 2)

The correct answer is / නිවැරදි පිළිතුර නම්,

2) To temporarily store the most frequently accessed data.
වැඩිම නිරන්තරයෙන් ප්‍රවේශ කරනු ලබන දත්ත තාවකාලිකව ගබඩා කිරීමට.

Cache memory is a high-speed memory located close to the CPU and is used to improve processing speed by storing frequently accessed data and instructions.
නිහිත මතකය යනු CPU එකට ආසන්නව පිහිටා ඇති අධිවේගී මතකයක් වන අතර නිතර ප්‍රවේශ වන දත්ත සහ උපදෙස් ගබඩා කිරීමෙන් සැකසුම් වේගය වැඩි දියුණු කිරීමට භාවිතා කරයි.

The purpose of cache memory is to provide temporary storage for data and instructions that are frequently used by the processor, reducing the need to access slower main memory.
නිහිත මතකයේ අරමුණ වන්නේ සකසනය විසින් නිතර භාවිතා කරන දත්ත සහ උපදෙස් සඳහා තාවකාලික ගබඩාවක් සැපයීම, මන්දගාමී ප්‍රධාන මතකයට ප්‍රවේශ වීමේ අවශ්‍යතාවය අඩු කිරීමයි.

Why Option 2 is Correct / විකල්ප 2 නිවැරදි වන්නේ ඇයි,

Cache memory is temporary storage. / නිහිත මතකය තාවකාලික ගබඩාවකි.

It stores the most frequently accessed data because these data are likely to be reused soon, which helps to improve performance.
එය නිතර ප්‍රවේශ වන දත්ත ගබඩා කරන්නේ මෙම දත්ත ඉක්මනින් නැවත භාවිතා කිරීමට ඉඩ ඇති නිසා, එය කාර්ය සාධනය වැඩි දියුණු කිරීමට උපකාරී වේ.

Why the Other Options Are Incorrect, / වෙනත් විකල්ප වැරදි වන්නේ ඇයි,

1) To permanently store the most frequently accessed data
වැඩිම නිරන්තරයෙන් ප්‍රවේශ කරනු ලබන දත්ත ස්ථිරව ගබඩා කිරීමට.
Incorrect because cache memory is not used for permanent storage. It is volatile and loses data when power is turned off.
ස්ථිර ගබඩා කිරීම සඳහා නිහිත මතකය භාවිතා නොකරන නිසා වැරදියි. එය නශ්‍ය වන අතර බලය අක්‍රිය කළ විට දත්ත නැති වේ.

3) To temporarily store large volumes of data
විශාල දත්ත පරිමාවක් තාවකාලිකව ගබඩා කිරීමට.
Incorrect because cache memory is typically small in size compared to main memory or secondary storage. It is optimized for speed, not volume.
ප්‍රධාන මතකය හෝ ද්විතියික ගබඩාවට සාපේක්ෂව නිහිත මතකය සාමාන්‍යයෙන් ප්‍රමාණයෙන් කුඩා බැවින් වැරදියි. එය පරිමාව සඳහා නොව වේගය සඳහා ප්‍රශස්තිකරණය කර ඇත.

4) To permanently store the least frequently accessed data
අවම නිරන්තරයෙන් ප්‍රවේශ කරනු ලබන දත්ත ස්ථිරව ගබඩා කිරීමට.
Incorrect because cache memory is not used for permanent storage, and it prioritizes frequently accessed data, not the least accessed.
නිහිත මතකය ස්ථිර ගබඩා කිරීම සඳහා භාවිතා නොකරන නිසා වැරදියි, සහ එය නිතර ප්‍රවේශ වන දත්ත ප්‍රමුඛත්වය දෙන්නේ, අවම වශයෙන් ප්‍රවේශ වන නිසා නොවේ.

5) To temporarily store the least frequently accessed data
අවම නිරන්තරයෙන් ප්‍රවේශ කරනු ලබන දත්ත තාවකාලිකව ගබඩා කිරීමට.
Incorrect because cache memory is specifically designed to store the most frequently accessed data, not the least.
නිහිත මතකය විශේෂයෙන් නිර්මාණය කර ඇත්තේ අවම වශයෙන් නොව නිතර ප්‍රවේශ වන දත්ත ගබඩා කිරීම සඳහා ය.
Cache memory is temporary and stores the most frequently accessed data to enhance processing speed, making 2) To temporarily store the most frequently accessed data the correct answer.
නිහිත මතකය තාවකාලික වන අතර සැකසුම් වේගය වැඩි දියුණු කිරීම සඳහා නිතර ප්‍රවේශ වන දත්ත ගබඩා කරයි, 2) නිතර ප්‍රවේශ වන දත්ත තාවකාලිකව ගබඩා කිරීම නිවැරදි පිළිතුර බවට පත් කරයි.

2. $1101_2 + 1111_2 + 1001_2 - 0111_2 = ?$

1) 11100_2 2) 100101_2 3) $\underline{11110}_2$ 4) 111111_2 5) 110111_2

Answer : 3)

Before performing this calculation, let's first convert all the numbers given in the above question into decimals.
මෙම ගණනය කිරීම සිදු කිරීමට ප්‍රථම ඉහත ප්‍රශ්නයෙහි දැක්වෙන සියළු සංඛ්‍යා දශමය බවට පරිවර්ථනය කරමු.

1101_2			
1	1	0	1
2^3	2^2	2^1	2^0
8	4	-	1
13_{10}			

1111_2			
1	1	1	1
2^3	2^2	2^1	2^0
8	4	2	1
15_{10}			

1001_2			
1	0	0	1
2^3	2^2	2^1	2^0
8	-	-	1
9_{10}			

0111_2			
0	1	1	1
2^3	2^2	2^1	2^0
0	4	2	1
7_{10}			

- $13 + 15 + 9 - 7 = 30_{10}$

30_{10}				
2^4	2^3	2^2	2^1	2^0
1	1	1	1	0
11110_2				

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

3. $\left(A.D + 1(\overline{C.B} + B.D)\right).D + D.D$

Solve this boolean expression.
මෙම බූලියානු ප්‍රකාශනය සුළු කරන්න.

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

Answer : 5)

This boolean expression can be simplfied as follows.
මෙම බූලියානු ප්‍රකාශනය පහත ආකාරයට සුළු කළ හැකිය.

$\left(A.D + 1(\overline{C.B} + B.D)\right).D + D.D$	
$(A.D + \overline{C.B} + B.D).D + D.D$	Distributive law විඝටන න්‍යාය
$(A.D + \overline{C.B} + B.D).D + D$	Idempotent law තදේවභාවී න්‍යාය
D	Redundancy law සමතිරික්ත න්‍යාය

4. Which of the following is an example of real-time processing ?
තත්‍ය කාලීන සැකසුම් සඳහා උදාහරණයක් වන්නේ පහත සඳහන් කවරක් ද?

1) Payroll calculation / වැටුප් ගණනය කිරීම.
2) Billing system / බිල්පත් පද්ධතිය.
3) Automatic production line. ස්වයංක්‍රිය නිෂ්පාදන රේඛාව.
4) Data backup / දත්ත උපස්ථය.
5) Batch data entry / කණ්ඩායම් දත්ත ඇතුළත් කිරීම.

Answer : 3)

Let's explain the options in the above question.
ඉහත ප්‍රශ්නයේ ඇති විකල්ප මෙසේ පැහැදිලි කරමු.

1) Payroll calculation / වැටුප් ගණනය කිරීම
This process typically happens periodically, such as weekly or monthly, and does not require immediate results after each data entry. Therefore, it is not an example of real-time processing.
මෙම ක්‍රියාවලිය සාමාන්‍යයෙන් සතිපතා හෝ මාසිකව කාලානුරූපව සිදු වන අතර, එක් එක් දත්ත ඇතුළත් කිරීමෙන් පසු ක්ෂණික ප්‍රතිඵල අවශ්‍ය නොවේ. එබැවින්, එය තත්‍ය කාලීන සැකසුම් සඳහා උදාහරණයක් නොවේ.

2) Billing system / බිල්පත් පද්ධතිය
Although a billing system may seem like it processes information immediately, it generally processes bills after transactions are completed. This doesn't always require real-time immediate responses but may operate in a scheduled manner.
බිල්පත් පද්ධතියක් ක්ෂණිකව තොරතුරු සකසන බවක් පෙනෙන්නට තිබුණද, එය සාමාන්‍යයෙන් ගනුදෙනු අවසන් වූ පසු බිල්පත් සකසයි. මෙය සැමවිටම තත්‍ය කාලීන ක්ෂණික ප්‍රතිචාර අවශ්‍ය නොවන නමුත් කාලසටහන්ගත ආකාරයෙන් ක්‍රියාත්මක විය හැක.

3) Automatic production line / ස්වයංක්‍රිය නිෂ්පාදන රේඛාව
This is a true example of real-time processing. In an automatic production line, the system continuously monitors and processes data to control machinery in real time. Immediate feedback and actions are required to ensure the production process runs smoothly.
මෙය තත්‍ය කාලීන සැකසුම් සඳහා සැබෑ උදාහරණයකි. ස්වයංක්‍රිය නිෂ්පාදන රේඛාවක් තුළ, පද්ධතිය තත්‍ය කාලීනව යන්ත්‍රෝපකරණ පාලනය කිරීම සඳහා දත්ත අඛණ්ඩව අධීක්ෂණය කර දත්ත සකසයි. නිෂ්පාදන ක්‍රියාවලිය සුමටව පවත්වාගෙන යාම සහතික කිරීම සඳහා ක්ෂණික ප්‍රතිපෝෂණ සහ ක්‍රියාමාර්ග අවශ්‍ය වේ.

4) Data backup / දත්ත උපස්ථය
Data backups are typically done in the background, often on a scheduled basis, and do not need real-time processing. They don't require instant response to new input.
දත්ත උපස්ථ සාමාන්‍යයෙන් පසුබිමේ සිදු කරනු ලැබේ, බොහෝ විට නියමිත පදනම මත, සහ තත්‍ය කාලීන සැකසුම් අවශ්‍ය නොවේ. ඔවුන්ට නව ආදානය සඳහා ක්ෂණික ප්‍රතිචාරයක් අවශ්‍ය නොවේ.

5) Batch data entry / කණ්ඩායම් දත්ත ඇතුළත් කිරීම
Batch processing handles large amounts of data at once, typically after data is collected. This method is not real-time because it processes data in groups rather than instantly.
කණ්ඩායම් සැකසීම සාමාන්‍යයෙන් දත්ත රැස් කිරීමෙන් පසු විශාල දත්ත ප්‍රමාණයක් එකවර හසුරුවයි. මෙම ක්‍රමය තත්‍ය කාලීන නොවේ, මන්ද එය ක්ෂණිකව නොව කණ්ඩායම් වශයෙන් දත්ත සකසන බැවිනි.

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

5. In which memory management scheme does the operating system keep track of all free and occupied memory blocks?
මෙහෙයුම් පද්ධතිය සියලුම නිදහස් සහ නිදහස් නොවන මතක කොටස් පිළිබඳ වාර්තා තබා ගන්නේ කුමන මතක කළමනාකරණ ක්‍රමවේදය තුළද?

1) Paging / පිටුකරණය
2) Segmentation / ඛණ්ඩනය
3) Contiguous Memory Allocation / යාබද මතක විභජනය
4) Virtual Memory / අතථ්‍ය මතකය
5) Dynamic Memory Allocation / ගතික මතක විභජනය

Answer : 3)

Explanation / පැහැදිලි කිරීම

1) Paging / පිටුකරණය
Divides physical memory into fixed-sized blocks called frames and logical memory into blocks of the same size called pages. It doesn't require contiguous allocation, but it doesn't directly track free and occupied blocks in the way you described.
භෞතික මතකය රාමු ලෙස හැඳින්වෙන ස්ථාවර ප්‍රමාණයේ කොටස් වලට සහ තාර්කික මතකය පිටු ලෙස හඳුන්වන එකම ප්‍රමාණයේ කොටස් වලට බෙදයි. එයට එක දිගට විභජනය කිරීමක් අවශ්‍ය නොවේ, නමුත් එය විස්තර කළ ආකාරයට නිදහස් සහ නිදහස් නොවන කොටස් සෘජුවම නිරීක්ෂණය නොකරයි.

2) Segmentation / ඛණ්ඩනය
Divides logical memory into variable-sized segments. It also doesn't directly track free and occupied blocks in the way you described.
තාර්කික මතකය විචල්‍ය ප්‍රමාණයේ කොටස් වලට බෙදයි. එය ඔබ විස්තර කළ ආකාරයෙන් නිදහස් සහ නිදහස් නොවන කොටස් සෘජුවම නිරීක්ෂණය නොකරයි.

3) Contiguous Memory Allocation / අඛණ්ඩ මතක විභජනය කිරීම
Allocates a contiguous block of memory to a process. This scheme requires the operating system to keep track of free and occupied memory blocks to allocate and deallocate memory efficiently.
ක්‍රියාවලියකට එක දිගට මතක කොටස විභජනය කරයි. මෙම යෝජනා ක්‍රමයට මෙහෙයුම් පද්ධතියට මතකය කාර්යක්ෂමව විභජනය කිරීමට සහ බෙදා හැරීමට නිදහස් සහ නිදහස් නොවන මතක කොටස් පිළිබඳ වාර්තාවක් තබා ගැනීමට අවශ්‍ය වේ.

4) Virtual Memory / අතථ්‍ය මතකය
Combines hardware and software techniques to provide a larger address space than physical memory. It doesn't directly track free and occupied blocks in the way you described, but it relies on underlying memory management techniques like paging or segmentation.
භෞතික මතකයට වඩා විශාල ලිපින ඉඩක් සැපයීමට දෘඪාංග සහ මෘදුකාංග ශිල්පීය ක්‍රම ඒකාබද්ධ කරයි. එය ඔබ විස්තර කර ඇති ආකාරයට නිදහස් සහ නිදහස් නොවන කොටස් සෘජුවම හඹා නොයයි, නමුත් එය පිටුකරණය හෝ ඛණ්ඩනය වැනි යටින් පවතින මතක කළමනාකරණ ක්‍රම මත රඳා පවතී.

5) Dynamic Memory Allocation / ගතික මතක විභජනය කිරීම
Manages memory allocation at runtime, often using techniques like heap allocation. It doesn't directly track free and occupied blocks in the way you described, but it relies on memory management libraries or operating system services to allocate and deallocate memory.
බොහෝ විට heap විභජනය කිරීම වැනි තාක්ෂණික ක්‍රම භාවිතා කරමින් ධාවන වේලාවේදී මතක විභජනය කිරීම කළමනාකරණය කරයි. එය ඔබ විස්තර කළ ආකාරයෙන් නිදහස් සහ නිදහස් නොවන කොටස් සෘජුවම නිරීක්ෂණය නොකරයි, නමුත් එය මතකය විභජනය කිරීමට සහ බෙදා හැරීමට මතක කළමනාකරණ libraries හෝ මෙහෙයුම් පද්ධති සේවා මත රඳා පවතී.

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

6. Which of the following code snippet correctly converts a string input to an integer?
string ආදානය නිවැරදිව පූර්ණ සංඛ්‍යාවක් බවට පරිවර්තනය කරන්නේ පහත සඳහන් කුමන කේත කොටසින්ද?

1) `num = str (input ("Enter a number: "))`
2) `num = int (input ("Enter a number: "))`
3) `num = float (input ("Enter a number: "))`
4) `num = input ("Enter a number: ")`
5) `num = bool (input ("Enter a number: "))`

Answer : 2)

This code works by first using the `input()` function to take user input, which is always in string format. Then, the `int()` function is used to convert that string input into an integer, making it the correct way to convert a string input to an integer in Python.
මෙම කේතය ක්‍රියා කරන්නේ පරිශීලක ආදානය ලබා ගැනීම සඳහා ප්‍රථමයෙන් `input()` ශ්‍රිතය භාවිතා කිරීමෙනි, එය සැමවිටම string ආකෘතියේ ඇත. ඉන්පසුව, එම string ආදානය පූර්ණ සංඛ්‍යාවක් බවට පරිවර්තනය කිරීමට `int()` ශ්‍රිතය භාවිතා කරයි, එය Python හි string ආදානය පූර්ණ සංඛ්‍යාවක් බවට පරිවර්තනය කිරීමේ නිවැරදි ක්‍රමය බවට පත් කරයි.

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

7. Consider the following Python code
පහත සඳහන් පයිතන් කේතය සලකා බලන්න

```
my_string = "pythoniscool"
result = my_string[1:7:2]
print(result)
```

What will be the output of this code?
ඉහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

1) `ptn` 2) `yhn` 3) `ythn` 4) `yto` 5) `yhni`

Answer : 2)

- `my_string[1:7:2]` means,
  `my_string[1:7:2]` යනු,

Start at index 1 (second character).
index `1` (දෙවන අක්ෂරය) සිට ආරම්භ කරන්න.

Stop before index 7 (so it doesn't include index 7).
දර්ශකය 7 ට පෙර නවත්වන්න. (එබැවින් එය 7 වන index එක ඇතුළත් නොවේ).

Take every 2nd character in that range.
එම පරාසයේ සෑම 2 වන අක්ෂරයක්ම ගන්න.

For the string "`pythoniscool`", this picks the characters at index `1` (`y`), index `3` (`h`), and index `5` (`n`).
string සඳහා "`pythoniscool`", මෙය index `1` (`y`), index `3` (`h`) සහ index `5` (`n`) හි අක්ෂර තෝරා ගනී.

So, the result is "`yhn`", and that's what gets printed.
එබැවින්, ප්‍රතිඵලය වන්නේ "`yhn`" සහ එයයි මුද්‍රණය වන්නේ.

8. What is the output of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
x= [10, 20, 30, 40, 50]
x.append(60)
x.insert(2,5000)
x.insert(4,5000)
print (x)
```

1) [10, 20, 30, 40, 50]
2) [10, 20, 30, 40, 50, 60]
3) [10, 20, 5000, 30, 5000, 40, 50, 60]
4) [10, 20, 5000, 30, 5000, 50, 60]
5) [10, 20, 5000, 30, 40, 5000, 60]

Answer : 3)

The following steps will make it more clear what has happened in the code.
මෙහිදී කේතයෙන් සිදුවී ඇති දේ පහත පියවරයන් මගින් වඩාත් පැහැදිලි වේ.

1) **Step 1:** The list x is initialized with the given values.
පියවර 1: x ලැයිස්තුව ලබා දී ඇති අගයන් සමඟ ආරම්භ වේ.

2) **Step 2:** The value `60` is added to the end of the list using `append()`.
පියවර 2: `append()` භාවිතයෙන් ලැයිස්තුවේ අවසානයට `60` අගය එකතු කෙරේ.

3) **Step 3:** The value `5000` is inserted at index 2, shifting elements to the right.
පියවර 3: index - 2 හි අගය `5000` ඇතුළත් කර, elements දකුණට මාරු කරයි.

4) **Step 4:** Another `5000` is inserted at index 4, again shifting elements to the right.
පියවර 4: තවත් `5000` ක් index - 4 හි ඇතුළත් කර, නැවතත් elements දකුණට මාරු කරයි.

5) **Step 5:** The final contents of the list `x` are printed.
පියවර 5: `x` ලැයිස්තුවේ අවසාන අන්තර්ගතය මුද්‍රණය කර ඇත.

It is further explained by the table below.
එය පහත වගුව මගින් වැඩිදුර පැහැදිලි වේ.

Step	Statement	x
1	`x = [10, 20, 30, 40, 50]`	`[10, 20, 30, 40, 50]`
2	`x.append(60)`	`[10, 20, 30, 40, 50, 60]`
3	`x.insert(2, 5000)`	`[10, 20, 5000, 30, 40, 50, 60]`
4	`x.insert(4, 5000)`	`[10, 20, 5000, 30, 5000, 40, 50, 60]`
5	`print(x)`	`[10, 20, 5000, 30, 5000, 40, 50, 60]`

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

9. You create a link to download a PDF.
ඔබ PDF එකක් බාගත කිරීම සඳහා සබැඳියක් සාදයි.

```
<a href="report.pdf" download>Download Report</a>
```

The PDF opens in the browser instead of downloading. What's the hidden reason?
PDF එක බාගත කිරීම වෙනුවට අතිරික්සුවේ විවෘත වේ. සැඟවුණු හේතුව කුමක්ද?

1) The server doesn't set the Content-Disposition header.
   සේවාදායකය Content-Disposition ශීර්ෂය සකසා නැත.
2) Missing `target="_blank"` / `target="_blank"` යන්න යොදා නොමැත.
3) The download attribute only works with images.
   බාගත කිරීමේ ගුණාංගය ක්‍රියා කරන්නේ පින්තූර සමඟ පමණි.
4) Use download="true" instead / ඒ වෙනුවට download="true" භාවිතා කරන්න.
5) The browser blocks downloads from untrusted sites.
   අතිරික්සුව විශ්වාස කළ නොහැකි වෙබ් අඩවි වලින් බාගත කිරීම් අවහිර කරයි.

Answer : 1)

Explanation of the Issue. / ගැටලුව පිළිබඳ පැහැදිලි කිරීම.
The issue lies with the server configuration. When a link is clicked, the browser relies on the server's instructions for how to handle the linked file. If the Content-Disposition header is not set to attachment, the browser assumes the file should be opened in the browser itself, which is the default behavior for PDFs.
ගැටළුව පවතින්නේ සේවාදායක වින්‍යාසය සමඟ ය. සබැඳියක් ක්ලික් කළ විට, සබැඳි ගොනුව හැසිරවිය යුතු ආකාරය සඳහා අතිරික්සු සේවාදායකයේ උපදෙස් මත විශ්වාසය තබයි. Content-Disposition ශීර්ෂය ඇමුණුමට සකසා නොමැති නම්, ගොනුව අතිරික්සුව තුළම විවෘත කළ යුතු යැයි අතිරික්සුව උපකල්පනය කරයි, එය PDF සඳහා පෙරනිමි හැසිරීමයි.

Why Option 1 is Correct ? / විකල්ප 1 නිවැරදි වන්නේ ඇයි?

Option 1 is the correct answer because the Content-Disposition header determines whether a file is downloaded or displayed. If the server doesn't specify `Content-Disposition: attachment; filename="report.pdf"`, the browser will attempt to open the PDF instead of prompting the download.
විකල්ප 1 නිවැරදි පිළිතුර වන්නේ ගොනුවක් බාගත කර තිබේද නැතහොත් දර්ශනය වේද යන්න Content-Disposition ශීර්ෂය තීරණය කරන බැවිනි. සේවාදායකය `Content-Disposition: attachment; filename="report.pdf"` සඳහන් නොකරන්නේ නම්, අතිරික්සුව බාගත කිරීම සඳහා පෙළඹවීම වෙනුවට ජෘෆ විවෘත කිරීමට උත්සාහ කරනු ඇත.

Why Other Options Are Incorrect? / වෙනත් විකල්ප වැරදි වන්නේ ඇයි?

Option / විකල්පය 2:
Missing target="_blank" affects how the link opens (in a new tab), not whether the file is downloaded.
`target="_blank"` නොමැති වීම ගොනුව බාගත කර තිබේද යන්නට නොව සබැඳිය විවෘත වන ආකාරය කෙරෙහි බලපායි.

Option / විකල්පය 3:
The `download` attribute works for more than images; it can work for PDFs and other file types as well.
`download` කරගැනීමේ ගුණාංගය රූපවලට වඩා ක්‍රියා කරයි; එය PDF සහ අනෙකුත් ගොනු වර්ග සඳහාද ක්‍රියා කළ හැකිය.

Option / විකල්පය 4:
There is no valid `download="true"` attribute. The `download` attribute alone is sufficient.
වලංගු `download="true"`ගුණාංගයක් නොමැත. `download` කිරීමේ ගුණාංගය පමණක් ප්‍රමාණවත් වේ.

Option / විකල්පය 5:
Browser blocking downloads from untrusted sites is unrelated in this context, as the issue is due to server-side configuration, not security restrictions.
විශ්වාසදායක නොවන අඩවි වලින් බාගත කර ගැනීම් අතිරික්සුව අවහිර කිරීම මෙම සන්දර්භය තුළ අදාළ නොවේ, මන්ද ගැටළුව ආරක්ෂක සීමාවන් නිසා නොව සේවාදායක පාර්ශව වින්‍යාසය නිසා ඇති වූවකි.

10. What happens when the following <a> tag is used?
පහත <a> උසුලනය භාවිතා කළ විට කුමක් සිදුවේද?

```
<a href="#section2">Go to Section 2</a>
```

1) The link redirects the user to another webpage called section2.html.
සබැඳිය පරිශීලකයා section2.html නමින් හැඳින්වෙන වෙනත් වෙබ් පිටුවකට හරවා යවයි.
2) The link opens a new tab displaying Section 2 of the current document.
සබැඳිය වත්මන් ලේඛනයේ 2 වන කොටස පෙන්වන නව පටිත්තක් (new tab) විවෘත කරයි.
3) The link navigates to an element with id="section2" within the same document.
සබැඳිය එකම ලේඛනය තුළ id="section2" සහිත මූලාංගයකට සංචාලනය කරයි.
4) The link does nothing because #section2 is not a valid URL.
#section2 වලංගු URL එකක් නොවන නිසා සබැඳිය කිසිවක් නොකරයි.
5) The link scrolls to the top of the document, regardless of section2.
section2 නොසලකා සබැඳිය ලේඛනයේ ඉහළට අනුචලනය වේ.

Answer: 3)

Explanation / පැහැදිලි කිරීම:

The href="#section2" attribute in the <a> tag refers to an anchor within the same webpage.
<a> උසුලනයේ ඇති href="#section2" ගුණාංගය එකම වෙබ් පිටුව තුළ ඇති anchor එකට යොමු කරයි.

The #section2 is an internal link, meaning it looks for an element in the same document that has the id="section2".
#section2 යනු අභ්‍යන්තර සබැඳියක් වන අතර, එයින් අදහස් වන්නේ එය id="section2" ඇති එම ලේඛනයේම මූලද්‍රව්‍යයක් සොයන බවයි.

When the user clicks the link, the page scrolls to the section with the matching id.
පරිශීලකයා සබැඳිය ක්ලික් කළ විට, පිටුව ගැලපෙන හැඳුනුම්පත සහිත කොටසට අනුචලනය වේ.

If there is no element with id="section2", clicking the link will do nothing noticeable except possibly causing a slight jump to the top of the page.
id="section2" සහිත මූලද්‍රව්‍යයක් නොමැති නම්, සබැඳිය ක්ලික් කිරීමෙන් පිටුවේ ඉහළට සුළු පැනීමක් ඇති කිරීම හැර වෙනත් සැලකිය යුතු දෙයක් සිදු නොවේ.

Correct answer / නිවැරදි පිළිතුර:
3) The link navigates to an element with id="section2" within the same document.
සබැඳිය එකම ලේඛනය තුළ id="section2", සහිත මූලාංගයකට සංචාලනය කරයි.