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 432 MARKING

2026/02/15
ictfromabc
Ravindu Bandaranayake

1. LED monitors achieve greater energy efficiency because,
LED මොනිටර වැඩි බලශක්ති කාර්යක්ෂමතාවයක් ලබා ගන්නේ,

1) They do not require individual transistors for each pixel.
එක් එක් පික්සල් සඳහා තනි ට්‍රාන්සිස්ටර අවශ්‍ය නොවන නිසා.
2) They use light-emitting diodes as their primary light source.
ඔවුන් ඔවුන්ගේ ප්‍රධාන ආලෝක ප්‍රභවය ලෙස ආලෝක විමෝචක ඩයෝඩ භාවිතා කරන බැවිනි.
3) They rely on electron beam technology, which minimizes energy loss.
මක්නිසාදයත් ඒවා බලශක්ති අලාභය අවම කරන ඉලෙක්ට්‍රෝන කදම්භ තාක්ෂණය මත රඳා පවතින බැවිනි.
4) They utilize liquid crystals without a backlight.
ඔවුන් පසුතල ආලෝකයක් නොමැතිව දියර ස්ඵටික භාවිතා කරන බැවිනි.
5) They are made of entirely biodegradable materials.
ඒවා සම්පූර්ණයෙන්ම ජෛව හායනයට ලක්විය හැකි ද්‍රව්‍ය වලින් සාදා ඇති බැවිනි.

**Answer : 2 )**

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

**Option / විකල්පය 1)**
Incorrect. LED monitors require individual transistors as part of their operation, especially in pixel-based displays like OLED or active-matrix LED displays.
වැරදි වේ. තිර වලට ඒවායේ ක්‍රියාකාරිත්වයේ කොටසක් ලෙස තනි ට්‍රාන්සිස්ටර අවශ්‍ය වේ, විශේෂයෙන් OLED හෝ active-matrix LED තිර වැනි පික්සල් මත පදනම් වූ තිරවල ය.

**Option / විකල්පය 2)**
Correct. LED monitors are energy-efficient because light-emitting diodes (LEDs) are used as the primary light source. LEDs consume less power compared to older technologies.
නිවැරදි වේ. ප්‍රාථමික ආලෝක ප්‍රභවය ලෙස ආලෝක විමෝචක ඩයෝඩ (LED) භාවිතා කරන බැවින් LED තිර බලශක්ති කාර්යක්ෂම වේ. පැරණි තාක්ෂණයට සාපේක්ෂව LED අඩු බලයක් පරිභෝජනය කරයි.

**Option / විකල්පය 3)**
Incorrect. This is related to CRT (Cathode Ray Tube) monitors, not LED monitors. Electron beam technology is outdated and not relevant to LED displays.
වැරදි වේ. මෙය සම්බන්ධ වන්නේ CRT (Cathode Ray Tube) මොනිටරවලට මිස LED මොනිටරවලට නොවේ. ඉලෙක්ට්‍රෝන කදම්භ තාක්ෂණය යල්පැන ඇති අතර LED සංදර්ශක සඳහා අදාළ නොවේ.

**Option / විකල්පය 4)**
Incorrect. This statement refers to certain OLED displays that don't need a backlight, but traditional LED monitors require an LED backlight to illuminate the liquid crystals.
වැරදි වේ. මෙම ප්‍රකාශය පසුබිම් ආලෝකයක් අවශ්‍ය නොවන ඇතැම් OLED සංදර්ශක සඳහා වේ., නමුත් සාම්ප්‍රදායික LED මොනිටරවලට ද්‍රව ස්ඵටික ආලෝකමත් කිරීමට LED පසුබිම් ආලෝකයක් අවශ්‍ය වේ.

**Option / විකල්පය 5)**
Incorrect. This is unrelated to energy efficiency. While the use of biodegradable materials can be environmentally friendly, it does not impact energy consumption.
වැරදි වේ. මෙය බලශක්ති කාර්යක්ෂමතාවයට සම්බන්ධ නැත. ජෛව හායනයට ලක්විය හැකි ද්‍රව්‍ය භාවිතය පරිසර හිතකාමී විය හැකි නමුත් එය බලශක්ති පරිභෝජනයට බලපාන්නේ නැත.

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

2. Consider the following statements.
 පහත ප්‍රකාශ සලකන්න.

 A. ASCII codes use an alphabetic scheme
 ASCII කේත වල අක්ෂර රේඛීය පටිපාටියක් භාවිත කරයි.

 B. 1110 in BCD is not an invalid combination
 BCD වල 1110 වලංගු නොවන සංයෝජනයක් නොවේ.

 C. 8 bits are used in EBCDIC coding system
 EBCDIC කේත ක්‍රමයේ දී බිටු 8 ක් භාවිතා වේ.

 Which of the above statements is true,
 ඉහත සඳහන් ප්‍රකාශ අතුරෙන් සත්‍ය වන්නේ

 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.

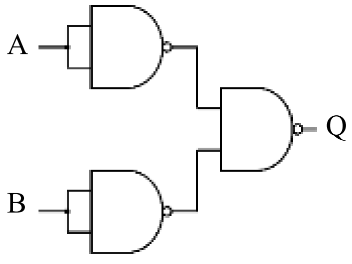
**Answer: 5)**

A - ASCII codes use an alphabetic scheme.
 ASCII කේත වල අක්ෂර රේඛීය පටිපාටියක් භාවිත කරයි.

B - Transitions 1010, 1011, 1100, 1101, 1110, 1111 are invalid combinations of BCD.
 1010, 1011, 1100, 1101, 1110, 1111 යන සංඛ්‍යා BCD වල වලංගු නොවන සංයෝජනයන් වේ.

C - 8 bits are used in EBCDIC coding system.
 EBCDIC කේත ක්‍රමයේ දී බිටු 8 ක් භාවිතා වේ.

3. Following universal gate equals to…….. ,
 පහත දැක්වෙන සාර්වත්‍ර ද්වාරය ........ට සමාන වේ,




 1) AND gate
 2) OR gate
 3) NOT gate
 4) XOR gate
 5) NAND gate

**Answer : 2)**

Universal gates are special types of logic gates that can be used to create any other type of logic gate, such as AND, OR, NOT, NAND, NOR, etc. These gates are considered "universal" because they can replicate the functions of other gates.
සාර්වත්‍ර ද්වාර යනු AND OR NOT යනාදී වෙනත් ඕනෑම තාර්කික ද්වාර වර්ගයක් නිර්මාණය කිරීමට භාවිතා කළ හැකි විශේෂ තාර්කික ද්වාර වර්ග වේ. මෙම ද්වාර සාර්වත්‍ර ලෙස සැලකෙන්නේ ඒවා මගින් අනෙක් ද්වාරද සාදා ගත හැකි බැවිනි. බොහෝ විට මේවා සෑදීමට NAND, NOR යන ද්වාර භාවිතා කරයි

4. Which encoding system does NTFS support that improves character representation?
අක්ෂර නිරූපණය වැඩි දියුණු කිරීමට NTFS සඳහා සහාය දක්වන කේතීකරණ පද්ධතිය කුමක්ද?

   1) ASCII  2) Unicode  3) ISO-8859-1  4) ANSI  5) EBCDIC

   **Answer : 2)**

   The correct answer is Unicode because NTFS supports this encoding system to enhance character representation. Unicode allows NTFS to handle a vast range of characters from various languages, symbols, and special characters, making it ideal for global compatibility and supporting internationalization. This broad character set helps NTFS store file names and metadata in almost any language, providing flexibility and preventing data corruption that might occur with limited encoding systems.
   නිවැරදි පිළිතුර Unicode වේ, මන්ද NTFS මෙම කේතීකරණ පද්ධතියට අක්ෂර නිරූපණය වැඩි දියුණු කිරීමට සහය දක්වයි. Unicode NTFS හට විවිධ භාෂා, සංකේත සහ විශේෂ අක්ෂර වලින් විශාල පරාසයක අක්ෂර හැසිරවීමට ඉඩ සලසයි, එය ගෝලීය ගැළපුම සහ ජාත්‍යන්තරකරණය සඳහා වඩාත් සුදුසු වේ. මෙම පුළුල් අක්ෂර කට්ටලය NTFS ඕනෑම භාෂාවකින් පාහේ ගොනු නාම සහ දත්ත පිළිබඳ දත්ත ගබඩා කිරීමට උපකාරී වේ, නම්‍යශීලී බව සහ සීමිත කේතීකරණ පද්ධති සමඟ සිදු විය හැකි දත්ත විනාශ වීම වළක්වයි.

   The other options are incorrect as they offer more restricted character representations. ASCII supports only basic English characters and symbols, not the full range needed for diverse languages. ISO-8859-1 extends ASCII but is still limited to Western European characters. ANSI varies between operating systems, offering some extended characters but lacking global language support. EBCDIC is primarily used in older IBM systems and does not support the variety of characters Unicode provides, making these options less effective for NTFS's global character requirements.
   අනෙකුත් විකල්ප වඩාත් සීමා කළ අක්ෂර නිරූපණයන් ලබා දෙන බැවින් ඒවා වැරදිය. ASCII සහාය දක්වන්නේ මූලික ඉංග්‍රීසි අක්ෂර සහ සංකේතවලට මිස විවිධ භාෂා සඳහා අවශ්‍ය සම්පූර්ණ පරාසයට නොවේ. ISO-8859-1 යනු ASCII හි දිගුවක් නමුත් තවමත් බටහිර යුරෝපීය අක්ෂරවලට සීමා වේ. ANSI මෙහෙයුම් පද්ධති අතර වෙනස් වන අතර, සමහර විස්තීරණ අක්ෂර ලබා දෙන නමුත් ගෝලීය භාෂා සහාය නොමැත. EBCDIC මූලික වශයෙන් පැරණි IBM පද්ධතිවල භාවිතා වන අතර Unicode සපයන විවිධ අක්ෂර සඳහා සහය නොදක්වයි, මෙම විකල්පයන් NTFS හි ගෝලීය අක්ෂර අවශ්‍යතා සඳහා අඩු ඵලදායී බවට පත් කරයි.

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

5. What SQL command would use to delete all rows where the salary column is less than 20000?
වැටුප් තීරුවේ 20000 ට වඩා අඩු සියලුම පේළි මකා දැමීමට භාවිතා කරන්නේ කුමන SQL විධානයද?

   1) `DELETE * FROM employees WHERE salary < 20000;`
   2) `DELETE age FROM employees WHERE salary < 20000;`
   3) `DROP employees WHERE salary < 20000;`
   4) `DELETE FROM employees WHERE salary < 20000;`
   5) `REMOVE FROM employees WHERE salary < 20000`

   **Answer : 4)**

   **Explanation/ පැහැදිලි කිරීම,**

   In SQL, the command that used to delete rows from tables are/ වගු වලින් පේළි මකා දැමීමට භාවිතා කරන විධානය වන්නේ,

   DELETE FROM tableName
   WHERE condition;

   The WHERE clause is used to filter records based on a condition. Only the records that satisfy the condition will be affected by the DELETE operation.
   කොන්දේසියක් මත පදනම්ව වාර්තා ලබා ගැනීමට WHERE භාවිතා කරයි. කොන්දේසිය සපුරාලන වාර්තා පමණක් DELETE මෙහෙයුමෙන් ඉවත් කරනු ලැබේ.

According to the above question, condition is salary < 20000. It means that the deletion will only apply to records where the salary column has a value less than 20,000.
ඉහත ප්‍රශ්නයට අනුව, කොන්දේසිය salary < 20000 වේ. එයින් අදහස් වන්නේ මකාදැමීම අදාළ වන්නේ වැටුප් තිරුවේ අගය 20000 ට වඩා අඩු වාර්තා සඳහා පමණක් බවයි.

So, any employee in the employees table with a salary below 20,000 will have their record deleted.
එබැවින්, සේවක වගුවේ සිටින රු.20000 ට අඩු වැටුපක් ඇති ඕනෑම සේවකයෙකුගේ වාර්තාව මකා දමනු ලැබේ.

Therefore the answer is 4) / නිවැරදි පිළිතුර 4) වේ.

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

```
for i in range(1, 8):
    if i == 5:
        break
    print(i, end=",")
```

1) `1 2 3` 2) `1,2,3,4,` 3) `1 2 3 4 5 6 7` 4) `1, 2, 3 4` 5) Error

**Answer : 2)**

The correct answer is, 2) 1,2,3,4,
නිවැරදි පිළිතුර වන්නේ, 2) 1,2,3,4,

Explanation / පැහැදිලි කිරීම

The Python **for** loop iterates through the numbers from 1 to 7 (inclusive). The **break** statement causes the loop to terminate prematurely when i = = 5. Here's a step-by-step breakdown:
පයිතන් for ලූපය 1 සිට 7 දක්වා සංඛ්‍යා හරහා (ඇතුළත්) පුනරාවර්තනය වේ. break ප්‍රකාශය i = = 5 වන විට ලූපය අවසන් වීමට හේතු වේ. පියවරෙන් පියවර බිඳවැටීමක් මෙන්න:

Initialization: The **for** loop starts with i = 1.
ආරම්භකකරණය: for ලූපය i = 1 සමඟ ආරම්භ වේ.

Condition Check and Iteration
තත්ව පරීක්ෂාව සහ පුනරාවර්තනය

i = 1: Not equal to 5, so it prints 1, and continues.
i = 1: 5 ට සමාන නොවන බැවින් එය 1 මුද්‍රණය කර දිගටම පවතී.

i = 2: Not equal to 5, so it prints 2, and continues.
i = 2: 5 ට සමාන නොවන බැවින් එය 2 මුද්‍රණය කර දිගටම පවතී.

i = 3: Not equal to 5, so it prints 3, and continues.
i = 3: 5 ට සමාන නොවන බැවින් එය 3 මුද්‍රණය කර දිගටම පවතී.

i = 4: Not equal to 5, so it prints 4, and continues.
i = 4: 5 ට සමාන නොවන බැවින් එය 4 මුද්‍රණය කර දිගටම පවතී.

i = 5: Equal to 5, so the break statement is executed, and the loop terminates immediately.
i = 5: 5 ට සමාන වේ, එබැවින් break ප්‍රකාශය ක්‍රියාත්මක වන අතර, ලූපය වහාම අවසන් වේ.

Output / ප්‍රතිදානය

The numbers 1,2,3,4, are printed in a single line because of end=",".
1,2,3,4, සංඛ්‍යා තනි පේළියකින් මුද්‍රණය කර ඇත්තේ end="," නිසාය.

The loop stops before printing 5. Final Output, 1,2,3,4,
5 මුද්‍රණය කිරීමට පෙර ලූපය නතර වේ. අවසාන ප්‍රතිදානය = 1,2,3,4,

7. What will happen if you open a non-existent file in "r" mode?
ඔබ නොපවතින ගොනුවක් "r" ආකාරයෙන් විවෘත කළහොත් කුමක් සිදුවේද?

1) It will create a new file.
එය නව ගොනුවක් නිර්මාණය කරයි.
2) It will raise a FileNotFoundError.
එය FileNotFoundError එකක් මතු කරයි.
3) It will open the file in read-only mode.
එය ගොනුව කියවීමට පමණක් ආකාරයෙන් විවෘත කරයි.
4) It will overwrite the existing file.
එය පවතින ගොනුව මත ලියයි.
5) It will open the file in binary mode.
එය ගොනුව ද්විමය ආකාරයෙන් විවෘත කරයි.

**Answer : 2)**

**Option / විකල්පය 1**
Incorrect. The "r" mode is for reading only and does not create new files.
වැරදියි. "r" ආකාරය කියවීම සඳහා පමණක් වන අතර නව ගොනු නිර්මාණය නොකරයි.

**Option / විකල්පය 2**
Correct. When you attempt to open a non-existent file in "r" mode (read mode), Python raises a `FileNotFoundError` because the file does not exist and cannot be opened for reading.
නිවැරදියි. ඔබ "r" ආකාරයේ (කියවීමේ ආකාරය) නොපවතින ගොනුවක් විවෘත කිරීමට උත්සාහ කරන විට, ගොනුව නොපවතින අතර කියවීම සඳහා විවෘත කළ නොහැකි බැවින් පයිතන් `FileNotFoundError` එකක් මතු කරයි.

**Option / විකල්පය 3**
Incorrect. While "r" mode is for reading, it does not open a new file; it raises an error for non-existent files.
වැරදියි. "r" ආකාරය කියවීම සඳහා වන අතර, එය නව ගොනුවක් විවෘත නොකරයි, එය නොපවතින ගොනු සඳහා දෝෂයක් මතු කරයි.

**Option / විකල්පය 4**
Incorrect. "r" mode does not overwrite files, and if the file exists, it is opened for reading, not writing.
වැරදියි. "r" ආකාරය ගොනු මත ලියන්නේ නැත, සහ ගොනුව පවතී නම්, එය ලිවීම සඳහා නොව කියවීම සඳහා විවෘත වේ.

**Option / විකල්පය 5**
Incorrect. "r" mode opens the file in text mode, not binary.
වැරදියි. "r" මාදිලිය ගොනුව ද්විමය ආකාරයෙන් නොව පාඨ ආකාරයෙන් විවෘත කරයි.

So, the correct answer number is 2.
එමනිසා, නිවැරදි පිළිතුරු අංකය 2 වේ.

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

```
def quicksort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quicksort(left) + middle + quicksort(right)
print(quicksort([3, 6, 8, 10, 1, 2, 1]))
```

1) [1, 2, 3, 6, 8, 10]
2) [1, 2, 3, 6, 8, 10, 1]
3) [10, 8, 6, 3, 1, 2]
4) <u>[1, 1, 2, 3, 6, 8, 10]</u>
5) Error

**Answer: 4)**

**Base Case / මූලික අවස්ථාව:**
If the input array `arr` has 1 or fewer elements, it is already sorted, so it is returned as is.
ආදාන අරාව වන `arr` හි මූලාංග 1ක් හෝ ඊට අඩු නම්, එය දැනටමත් වර්ග කර ඇති බැවින්, එය එලෙසම ආපසු ලබා දෙනු ලැබේ.

**Recursive Case / පුනරාවර්තන අවස්ථාව:**
The `pivot` is selected as the middle element of the array: `arr[len(arr) // 2]`
අරාවෙහි මැද මූලාංගය ලෙස `pivot` තෝරා ඇත: `arr[len(arr) // 2]`

Three subarrays are created / උප අරාවන් තුනක් නිර්මාණය වේ:

**`left`**: Contains elements smaller than the `pivot`. / `pivot` ට වඩා කුඩා මූලාංග අඩංගු වේ.
**`middle`**: Contains elements equal to the `pivot`. / `pivot` ට සමාන මූලාංග අඩංගු වේ.
**`right`**: Contains elements greater than the `pivot`. / `pivot` ට වඩා විශාල මූලාංග අඩංගු වේ.

The function then recursively sorts the `left` and `right` subarrays and combines them with the `middle` array.
ඉන්පසු ශ්‍රිතය `left` සහ `right` උප අරා පුනරාවර්තන ලෙස වර්ග කර `middle` අරාව සමඟ ඒකාබද්ධ කරයි.

**Output / ප්‍රතිදානය:**
The sorted array is returned by concatenating `quicksort(left)`, `middle` and `quicksort(right)`.
අනුපිළිවෙලට සැකසූ අරාව ආපසු ලබා දෙනු ලබන්නේ `quicksort(left)`, `middle` සහ `quicksort(right)` සම්බන්ධ කිරීමෙනි.

**Initial Call / මූලික කැඳවීම:**
Array: `[3, 6, 8, 10, 1, 2, 1]`
Pivot: `7 // 2 = 3` → `10` (middle element / මැද මූලාංගය)

`left = [3, 6, 1, 2, 1]` (elements less than 8 / 8 ට අඩු මූලාංග)
`middle = [8]` (elements equal to 8 / 8 ට සමාන මූලාංග)
`right = [10]` (elements greater than 8 / 8 ට වැඩි මූලාංග)

Recursive call / පුනරාවර්තන කැඳවීම:
`quicksort([3, 6, 1, 2, 1]) + [8] + quicksort([10])`

- **Recursive Call 1 (Left Subarray) / පුනරාවර්තන කැඳවීම 1 (වම් උපඅරාව):**

  Array: `[3, 6, 1, 2, 1]`
  Pivot: `5 // 2 = 2 → 1`

  `left = []` (no elements less than 1 / 1 ට අඩු මූලාංග නැත)
  `middle = [1, 1]` (elements equal to 1 / 1 ට සමාන මූලාංග)
  `right = [3, 6, 2]` (elements greater than 1 / 1 ට වැඩි මූලාංග)

  Recursive call / පුනරාවර්තන කැඳවීම: `quicksort([]) + [1, 1] + quicksort([3, 6, 2])`

- **Recursive Call 2 (Right Subarray of Recursive Call 1):**
  **පුනරාවර්තන කැඳවීම 2 (පුනරාවර්තන කැඳවීම 1 හි දකුණු උපඅරාව):**

  Array: `[3, 6, 2]`
  Pivot: `3 // 2 = 1 → 6`

  `left = [3, 2]`
  `middle = [6]`
  `right = []`

  Recursive call / පුනරාවර්තන කැඳවීම: `quicksort([3, 2]) + [6] + quicksort([])`

- **Recursive Call 3 (Left Subarray of Recursive Call 2):**
  **පුනරාවර්තන කැඳවීම 3 (පුනරාවර්තන කැඳවීම 2 හි වම් උපඅරාව):**

  Array: `[3, 2]`
  Pivot: `2 // 2 = 1 → 2`

  `left = []`
  `middle = [2]`
  `right = [3]`

  Recursive call / පුනරාවර්තන කැඳවීම: `quicksort([]) + [2] + quicksort([3])`

**Recursive Calls for Base Cases / මූලික අවස්ථා සඳහා පුනරාවර්තන කැඳවීම්::**
`quicksort([])` returns `[]` (base case).
`quicksort([3])` returns `[3]` (base case).
`quicksort([10])` returns `[10]` (base case).

**Combining Results / ප්‍රතිඵල ඒකාබද්ධ කිරීම:**
```
quicksort([3, 2]) → [] + [2] + [3] → [2, 3]
quicksort([3, 6, 2]) → [2, 3] + [6] + [] → [2, 3, 6]
quicksort([3, 6, 1, 2, 1]) → [] + [1, 1] + [2, 3, 6] → [1, 1, 2, 3, 6]
```

Final result / අවසාන ප්‍රතිඵලය:
`[1, 1, 2, 3, 6] + [8] + [10] → [1, 1, 2, 3, 6, 8, 10]`

So, the correct answer number is 4.
එමනිසා, නිවැරදි පිළිතුරු අංකය 4 වේ.

9. Visited links stay the same color despite this CSS code.
මෙම CSS කේතය නොසලකා, පිවිසුණු (Visited) සබැඳි එකම වර්ණයෙන් පවතී.

```
a:visited { color: purple; }
```

What's the privacy-related reason?
පෞද්ගලිකත්වය හා සම්බන්ධ හේතුව කුමක්ද?

1) Modern browsers restrict `:visited` styles for security.
   නවීන අතිරික්සු ආරක්ෂාව සඳහා `:visited` විලාසයන් සීමා කරයි.
2) The CSS is in the wrong order / CSS වැරදි පිළිවෙලට ඇත.
3) Use `!important` / `!important` භාවිතා කළ යුතුයි.
4) The link lacks `href` / සබැඳියේ `href` නොමැත.
5) Cache needs clearing / නිහිතය හිස් කිරීම අවශ්‍යයි.

**Answer: 1)**

**Explanation / පැහැදිලි කිරීම:**
Modern browsers limit styling of `:visited` links to prevent history sniffing attacks. Attackers could use different styles to detect which links a user has visited.
නවීන අතිරික්සු **history sniffing attacks** වැලැක්වීම සඳහා `:visited` **links** විලාස සීමා කරයි.
පරිශීලකයෙකු පැමිණ ඇත්තේ කුමන සබැඳිද යන්න හඳුනා ගැනීමට ප්‍රහාරකයන්ට විවිධ styles භාවිතා කළ හැක.

- **Allowed properties for** `:visited` / `:visited` සඳහා අවසර ගුණාංග
  `color`
  `background-color`
  `border-color`
  `outline-color`
  `column-rule-color`
  `fill` & `stroke` (SVG)

- **Restricted properties for** `:visited` / `:visited` සඳහා අවසර නැති ගුණාංග
  `display`

  `Content`
  `Visibility`
  `Position`
  `box-shadow`
  `text-decoration`, etc.

Why the other options are incorrect / වෙනත් විකල්ප වැරදි ඇයි?

1) **The CSS is in the wrong order / CSS වැරදි පිළිවෙලට ඇත**
   Order doesn't affect `:visited.`
   `:visited` සඳහා පිළිවෙල බලපාන්නේ නැත.

2) **Use `!important` / `!important` භාවිතා කරන්න**
   Browser restrictions override it.
   බ්‍රවුසර් සීමා එය override කරයි.

3) **The link lacks `href` / සබැඳියේ href නොමැත**
   `:visited` only applies to links with a valid `href`, but this isn't the issue here.
   `:visited` වලංගු `href` එකක් සහිත සබැඳි සඳහා පමණක් අදාළ වේ, නමුත් මෙහි ගැටලුව මෙය නොවේ.

4) **Cache needs clearing / Cache හිස් කිරීම අවශ්‍යයි**
   Cache doesn't affect `:visited` behavior.
   `:visited` cache සඳහා බලපාන්නේ නැත.

10. A screen reader skips a <nav> with multiple links. What’s wrong?
screen reader බහු සබැඳි සහිත <nav> මඟ හරියි. මෙහි දෝෂය කුමක්ද?

```
<nav>
  <div>Home</div>
  <div>About</div>
</nav>
```

1) Use `<a>` tags instead of `<div>` / `<div>` වෙනුවට `<a>` උසුලන භාවිතා කිරීම.
2) Add `role="navigation"` / `role="navigation"` එක් කිරීම.
3) Missing `aria-label` on `<nav>` / `<nav>` හි `aria-label` අස්ථානගත වී ඇත.
4) Wrap the `<nav>` in `<header>` / `<header>` තුළ `<nav>` යෙදීම.
5) Use `<ul>` and `<li>` / `<ul>` සහ `<li>` භාවිතා කිරීම.

**Answer: 1)**

**Explanation / පැහැදිලි කිරීම:**

The `<nav>` element is designed to contain navigation links, and for screen readers to properly interpret the links, they should be wrapped in `<a>` tags (anchor tags), which are designed for navigation purposes. Using `<div>` elements won’t provide the proper semantics for screen readers.
`<nav>` මූලාංගය සැලසුම් කර ඇත්තේ සංචාලන සබැඳි අඩංගු වන පරිදි වන අතර, තිර කියවන්නන්ට සබැඳි නිවැරදිව අර්ථ දැක්වීම සඳහා, ඒවා සංචාලන අරමුණු සඳහා නිර්මාණය කර ඇති `<a>` උසුලන (anchor උසුලන) වලින් wrap කර යුතුය. `<div>` මූලාංග භාවිතා කිරීම තිර කියවනය සඳහා නිසි අර්ථකථන ලබා නොදේ.

**Correct answer / නිවැරදි පිළිතුර:**
1) Use <a> tags instead of <div> / <div> වෙනුවට <a> උසුලන භාවිතා කරන්න

Why other options are incorrect / අනෙක් විකල්ප වැරදි ඇයි?

**Option / විකල්පය 2)**
The `<nav>` element already implicitly has the "navigation" role in HTML5. There's no need to manually add `role="navigation"`, as it's already understood by modern screen readers.
`<nav>` මූලාංගයට දැනටමත් HTML5 හි "navigation" භූමිකාව ව්‍යංගයෙන් ඇත. එය දැනටමත් නවීන තිර කියවන්නන් (screen readers) විසින් තේරුම් ගෙන ඇති බැවින්, `role="navigation"` අතින් එකතු කිරීමට අවශ්‍ය නැත.

**Option / විකල්පය 3)**
While adding an `aria-label` to the `<nav>` element could improve accessibility, it’s not the primary issue here. The issue is that the `<div>` elements inside the `<nav>` don't provide the correct navigation semantics that screen readers expect.
`<nav>` මූලාංගයට `aria-label` එකක් එක් කිරීම ප්‍රවේශ්‍යතාව වැඩිදියුණු කළ හැකි අතර, එය මෙහි මූලික ගැටලුව නොවේ. ගැටළුව වන්නේ, `<div>` තුළ ඇති `<nav>` මූලාංග තිර කියවන්නන් (screen readers) අපේක්ෂා කරන නිවැරදි සංචාලන අර්ථ ශාස්ත්‍රය ලබා නොදීමයි.

**Option / විකල්පය 4)**
This is not necessary. While `<nav>` can be placed inside a `<header>` element, it’s not the cause of the screen reader skipping the navigation. The problem is with the use of `<div>` instead of `<a>` tags.
මෙය අවශ්‍ය නොවේ. `<nav>` හි `<header>` මූලාංගයක් තුළ තැබිය හැකි වුවද, තිර කියවන (screen reader) සංචලනය මඟ හැරීමට හේතුව එය නොවේ. ගැටළුව වන්නේ `<a>` උසුලන වෙනුවට `<div>` භාවිතයයි.

**Option / විකල්පය 5)**
Using an unordered list (`<ul>`) with list items (`<li>`) is a common convention for organizing links in navigation menus, but it’s not required. The key issue here is the absence of anchor tags (`<a>`), not the structure of the links.
List items (`<li>`) සමඟ unordered list (`<ul>`) භාවිතා කිරීම සංචාලන මෙනුවල සබැඳි සංවිධානය කිරීම සඳහා පොදු සම්මුතියකි, නමුත් එය අවශ්‍ය නොවේ. මෙහි ඇති ප්‍රධාන ගැටළුව වන්නේ anchor උසුලන (`<a>`) නොමැතිකමයි, සබැඳිවල ව්‍යුහය නොවේ.