සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved]

Ravindu Bandaranayake - ictfromabc

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 382 MARKING

2025/12/27
ictfromabc
Ravindu Bandaranayake

1. What is the answer for devices in descending order as Considering the access speed?
ප්‍රවේශ වේගය සැලකූ විට අවරෝහණ පිළිවෙලට පවතින උපාංග සඳහන් පිළිතුර කුමක්ද?

    1) DVD, Flash drive, RAM, L3 cache, Registers
    සංඛ්‍යාංක බහුවිධ තැටි, සැනෙලි මතක, සසම්භාවී ප්‍රවේශ මතක, L3 වාරක මතක, රෙජිස්තර

    2) Registers, RAM, L3 cache, L2 cache, Hard disk
    රෙජිස්තර, සසම්භාවී ප්‍රවේශ මතක, L3 වාරක මතක, L2 වාරක මතක, දෘඪ තැටි

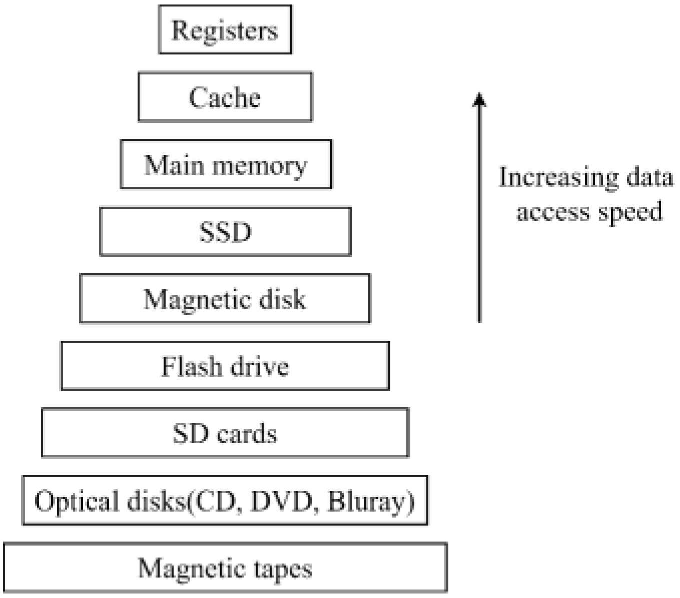
    3) L1 cache, NVMe drive, Registers, Flash drive, RAM
    L1 වාරක මතක, NVMe ධාවක, රෙජිස්තර, සැනෙලි මතක, සසම්භාවී ප්‍රවේශ මතක

    4) NVMe drive, L3 cache, Hard disk, Flash drive, Magnetic tape
    NVMe ධාවක, L3 වාරක මතක, දෘඪ තැටි, සැනෙලි මතක, චුම්බක පටි

    5) L2 Cache, RAM, Hard disk, Memory card, Magnetic tape
    L2 වාරක මතක, සසම්භාවී ප්‍රවේශ මතක, දෘඪ තැටි, මතක කාඩ්පත්, චුම්බක පටි

Answer : 5)

The computer memory hierarchy can be represented in a diagram as follows.
පරිගණක මතක ධුරාවලිය පහත ආකාරයට සටහනකින් දැක්විය හැකිය.



According to this diagram L2 Cache, RAM, Hard disk, Memory card, Magnetic tape can be taken as the correct answer.
මෙම සටහනට අනුව නිවැරදි පිළිතුර ලෙස L2 වාරක මතක, සසම්භාවී ප්‍රවේශ මතක, දෘඪ තැටි, මතක කාඩ්පත්, චුම්බක පටි ගත හැකිය.

So, the correct answer number is 5.
එමනිසා නිවැරදි පිළිතුරු අංකය 5 වේ.

2. What is the one's complement of the answer obtained by calculating $01001001_2+10010101_2-01100111_2$?
$01001001_2+10010101_2-01100111_2$ යන ගණනය කිරීමෙන් ලැබෙන පිළිතුරෙහි එකෙහි අනුපූරකය කුමක්වේද?

1) 1110110_2 2) $\underline{1110111}_2$ 3) 1111000_2 4) 0001000_2 5) 0001001_2

Answer : 2)

Let's first trivialize these three numbers.
පළමුව මෙම සංඛ්‍යා තුන සුළු කර ගනිමු.

$$
\begin{array}{r}
0\,1\,0\,0\,1\,0\,0\,1_2 \\
+\,1\,0\,0\,1\,0\,1\,0\,1_2 \\
\hline
1\,1\,0\,1\,1\,1\,1\,0_2 \\
\hline
\end{array}
$$

$$
\begin{array}{r}
1\,1\,0\,1\,1\,1\,1\,0_2 \\
-\;0\,1\,1\,0\,0\,1\,1\,1_2 \\
\hline
0\,1\,1\,1\,0\,1\,1\,1_2 \\
\hline
\end{array}
$$

The one's complement of a positive number is the binary value of that number. So the one's complement of the positive number 1110111_2 is 1110111_2 itself.
යම්කිසි ධන සංඛ්‍යාවක එකෙහි අනුපූරකය යනු එම සංඛ්‍යාවේ ද්විමය අගයම වේ. එනම් මෙහි ඇති 1110111_2 යන ධන සංඛ්‍යාවේ එකෙහි අනුපූරකය යනු 1110111_2 ම වේ.

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

3. Solve the boolean expression below.
පහත බූලියානු ප්‍රකාශනය සුළු කරන්න.

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

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

Answer : 3)

We simplify this expression using Boolean laws.
මෙම ප්‍රකාශනය බූලියානු නීති භාවිතයෙන් සුළු කරමු.

$C(\overline{1.\overline{B}}+D.B)+\overline{D}.D+\overline{A.B}.C$	
$C(\overline{1.\overline{B}}+D.B)+0+\overline{A.B}.C$	Inverse law ප්‍රතිලෝම න්‍යාය
$C(\overline{1.\overline{B}}+D.B)+\overline{A.B}.C$	Identity law සර්වසාම්‍ය න්‍යාය
$C(\overline{\overline{B}}+D.B)+\overline{A.B}.C$	Identity law සර්වසාම්‍ය න්‍යාය
$C(B+D.B)+\overline{A.B}.C$	Double inverse law ද්විත්ව ප්‍රතිලෝම න්‍යාය
$C.B+\overline{A.B}.C$	Redundancy law සමතිරික්ත න්‍යාය
$C.B+(\overline{A}+\overline{B}).C$	De Morgan’s law ද මුවාර් ප්‍රමේයය
$C(B+\overline{A}+\overline{B})$	Distributive law විඝටන න්‍යාය
$C(1+\overline{A})$	Inverse law ප්‍රතිලෝම න්‍යාය
$C(1)$	Identity law සර්වසාම්‍ය න්‍යාය
C	Identity law සර්වසාම්‍ය න්‍යාය

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

4. What is the primary goal of normalization in databases?
දත්ත සමුදායන්හි ප්‍රමතකරණයේ මූලික අරමුණ කුමක්ද?

1) Increase data redundancy / දත්ත අතිරික්තය වැඩි කිරීම.
2) Optimize query performance / විමසුම් කාර්ය සාධනය ප්‍රශස්ත කරන්න.
3) Minimize data anomalies and redundancy / දත්ත විෂමතා සහ අතිරික්තය අවම කිරීම.
4) Enhance security features / ආරක්ෂක විශේෂාංග වැඩි දියුණු කිරීම.
5) Ensure hardware compatibility / දෘඩාංග අනුකූලතාව සහතික කිරීම.

Answer : 3)

Explanation / පැහැදිලි කිරීම

If a student can enroll in multiple classes, and a class can have multiple students, the relationship between "Students" and "Classes" in a school database is Many-to-Many.
ශිෂ්‍යයෙකුට පන්ති කිහිපයකට ඇතුළත් විය හැකි නම් සහ පන්තියකට සිසුන් කිහිප දෙනෙකු සිටිය හැකි නම්, පාසල් දත්ත ගබඩාවක "Students" සහ "Classes" අතර සම්බන්ධතාවය බහු-බහු වේ.

This type of relationship means that each student can be associated with multiple classes, and each class can involve multiple students. To model this relationship, a junction table (e.g., "Enrollment") is typically used to link the "Students" and "Classes" tables, containing foreign keys from both tables to maintain the many-to-many connection.
මෙම ආකාරයේ සබඳතාවයක් යනු සෑම සිසුවෙකුටම පන්ති කිහිපයක් සමඟ සම්බන්ධ විය හැකි අතර සෑම පන්තියකටම සිසුන් කිහිප දෙනෙකු සම්බන්ධ කර ගත හැකිය. මෙම සම්බන්ධතාව ආදර්ශනය කිරීමට, හන්දි වගුවක් (උදා: "Enrollment") සාමාන්‍යයෙන් "Students" සහ "Classes" වගු සම්බන්ධ කිරීමට භාවිතා කරයි, බහු-බහු සම්බන්ධතාව පවත්වා ගැනීම සඳහා වගු දෙකෙන්ම විදේශීය යතුරු අඩංගු වේ.

Other relationship types like One-to-One or One-to-Many would not apply because they limit the associations to a single entity on one or both sides, which does not fit the scenario described.
විස්තර කරන ලද අවස්ථාවට නොගැළපෙන එක් හෝ දෙපස තනි භූතාර්තයට ආශ්‍ර සීමා කරන නිසා One-to-One හෝ One-to-Many වැනි වෙනත් සම්බන්ධතා වර්ග අදාළ නොවේ.

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

5. Which SQL clause is used to filter the results of a query?
විමසුමක ප්‍රතිඵල පෙරීමට භාවිතා කරන්නේ කුමන SQL වගන්තියද?

1) `ORDER BY` 2) `WHERE` 3) `GROUP BY` 4) `HAVING` 5) `SELECT`

Answer : 2)

Explanation / පැහැදිලි කිරීම

The WHERE clause in SQL is used to filter the results of a query by specifying a condition that rows must satisfy to be included in the output. It is applied before any grouping of data occurs and is commonly used with comparison operators like =, <, > or logical operators like AND and OR.
SQL හි WHERE වගන්තිය ප්‍රතිදානයට ඇතුළත් කිරීමට පේළි තෘප්තිමත් විය යුතු කොන්දේසියක් සඳහන් කරමින් විමසුමක ප්‍රතිඵල පෙරීමට භාවිතා කරයි. එය කිසියම් දත්ත සමූහයක් සිදුවීමට පෙර යෙදෙන අතර =, <, > වැනි සංසන්දන ක්‍රියාකරුවන් හෝ AND සහ OR වැනි තාර්කික ක්‍රියාකරුවන් සමඟ බහුලව භාවිතා වේ.

For example / උදාහරණයක් ලෙස :

```
SELECT *
FROM Students
WHERE Age > 18;
```

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

6. Which constraint allows setting a rule that restricts the values entered in a column based on a logical condition?
තාර්කික කොන්දේසියක් මත තිරුවක ඇතුළත් කළ අගයන් සීමා කරන රීතියක් සැකසීමට ඉඩ දෙන්නේ කුමන සම්බාධකයද?

1) `PRIMARY KEY` 2) `UNIQUE` 3) `DEFAULT` 4) <u>`CHECK`</u> 5) `FOREIGN KEY`

Answer: 4)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
A PRIMARY KEY constraint ensures that each row in a table has a unique and non-null value for the specified column or set of columns. It does not enforce logical conditions beyond ensuring uniqueness and non-null constraints.
ප්‍රාථමික යතුරු සීමාවක් මඟින් වගුවක ඇති සෑම පේළියකටම නිශ්චිත තිරුව හෝ තිරු කුලක සඳහා අනන්‍ය සහ ශුන්‍ය නොවන අගයක් ඇති බව සහතික කරයි. එය අනන්‍යතාව සහ ශුන්‍ය නොවන සීමාවන් සහතික කිරීමට ඔබ්බට තාර්කික කොන්දේසි බලාත්මක නොකරයි.

Option / විකල්ප 2)
A UNIQUE constraint ensures that all values in the specified column are distinct. It does not allow duplicate values but does not enforce logical conditions.
UNIQUE සීමාවක් නිශ්චිත තිරුවේ ඇති සියලුම අගයන් වෙනස් බව සහතික කරයි.එය අනුපිටපත් අගයන්ට ඉඩ නොදෙන නමුත් තාර්කික කොන්දේසි බලාත්මක නොකරයි.

Option / විකල්ප 3)
A DEFAULT constraint sets a default value for a column if no value is provided during insertion. It does not restrict the values entered but only provides a fallback value.
ඇතුළත් කිරීමේදී අගයක් සපයා නොමැති නම් DEFAULT සීමාවක් තිරුවක් සඳහා පෙරනිමි අගයක් සකසයි. එය ඇතුළත් කළ අගයන් සීමා නොකරන නමුත් ආපසු හැරවීමේ අගයක් පමණක් සපයයි.

Option / විකල්ප 4)
The CHECK constraint allows setting a rule that enforces logical conditions for values entered in a column. If the condition is violated, the database rejects the insertion or update. Therefore, this is the correct answer for restricting values based on logical conditions.
CHECK සීමාව මඟින් තිරුවක ඇතුළත් කළ අගයන් සඳහා තාර්කික කොන්දේසි බලාත්මක කරන රීතියක් සැකසීමට ඉඩ සලසයි. කොන්දේසිය උල්ලංඝනය වී ඇත්නම්, දත්ත සමුදාය ඇතුළත් කිරීම හෝ යාවත්කාලීන කිරීම ප්‍රතික්ෂේප කරයි. එබැවින් තාර්කික කොන්දේසි මත පදනම්ව අගයන් සීමා කිරීම සඳහා නිවැරදි පිළිතුර මෙයයි.

Option / විකල්ප 5)
A FOREIGN KEY constraint ensures that the values in a column correspond to values in another table's primary key or unique key. It enforces referential integrity but does not allow custom logical conditions.
FOREIGN KEY සීමාවක් මඟින් තිරුවක ඇති අගයන් වෙනත් වගුවක ප්‍රාථමික යතුරක හෝ අනන්‍ය යතුරක අගයන්ට අනුරූප වන බව සහතික කරයි. එය යොමු කිරීමේ අඛණ්ඩතාව බලාත්මක කරන නමුත් අභිරුචි තාර්කික කොන්දේසි වලට ඉඩ නොදේ.

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

7. What does the `len()` function do in Python?
පයිතන් හි `len()` ශ්‍රිතය සිදු කරන්නේ කුමක්ද?

1) Returns the length of a string, list, or tuple string / ලැයිස්තුවක හෝ tuple එකක දිග ආපසු ලබා දෙයි.
2) Removes the last element of a list / ලැයිස්තුවක අවසාන අංගය ඉවත් කරයි.
3) Calculates the size of an integer / පූර්ණ සංඛ්‍යාවක ප්‍රමාණය ගණනය කරයි.
4) Counts the number of variables / විචල්‍ය ගණන ගණනය කරයි.
5) Checks if a list is empty / ලැයිස්තුවක් හිස් දැයි පරීක්ෂා කරයි.

Answer : 1)

Explanation / පැහැදිලි කිරීම.

The `len()` function in Python is used to find the total number of elements or items contained in an object. This object can be a collection like a string, list, tuple, dictionary, or other iterable types. The function does not modify the object; it simply provides a count of the elements.
Pythonහි `len()` ශ්‍රිතය වස්තුවක අඩංගු මුළු මූලද්‍රව්‍ය හෝ අයිතම ගණන සොයා ගැනීමට භාවිතා කරයි. මෙම වස්තුව string, list, tuple, dictionary වෙනත් iterable වර්ග වැනි එකතුවක් විය හැකිය. ශ්‍රිතය වස්තුව වෙනස් නොකරයි එය හුදෙක් මූලද්‍රව්‍ය ගණන සපයයි.

Why the other options are incorrect / අනෙක් විකල්ප වැරදි ඇයි,

Removes the last element of a list / ලැයිස්තුවක අවසාන අංගය ඉවත් කරයි
`len()` does not remove anything; it only retrieves the count of elements.
`len()` කිසිවක් ඉවත් නොකරයි එය මූලද්‍රව්‍ය ගණන පමණක් ලබා ගනී.

Calculates the size of an integer / පූර්ණ සංඛ්‍යාවක ප්‍රමාණය ගණනය කරයි
It does not calculate the size of an integer; integers don't have a "length."
එය නිඛිලයක ප්‍රමාණය ගණනය නොකරයි නිඛිලවලට "length" නොමැත.

Counts the number of variables / විචල්‍ය ගණන ගණනය කරයි
It does not count the number of variables; it only works on objects with items or elements.
එය විචල්‍ය ගණන ගණන් නොකරයි එය අයිතම හෝ මූලද්‍රව්‍ය සහිත වස්තූන් මත පමණක් ක්‍රියා කරයි.

Checks if a list is empty / ලැයිස්තුවක් හිස් දැයි පරීක්ෂා කරයි
`len()` doesn't check if a list is empty, though if the length returned is 0, you can infer the list is empty.
`len()` ලැයිස්තුවක් හිස් දැයි පරීක්ෂා නොකරයි, නමුත් ආපසු ලබා දුන් දිග (0) නම්, ඔබට ලැයිස්තුව හිස් බව අනුමාන කළ හැකිය.

So, the correct answer is 1, as `len()` determines the length of iterable objects.
මේ අනුව, නිවැරදි පිළිතුර 1 වේ, මන්ද `len()` iterable වස්තූන්ගේ දිග තීරණය කරයි.

8. How does the Bubble Sort algorithm work?
Bubble Sort ඇල්ගොරිතම ක්‍රියා කරන්නේ කෙසේද?

1) It selects the smallest element and moves it to the front.
එය කුඩාම මූලාංග තෝරාගෙන එය ඉදිරිපසට ගෙන යයි.

2) It compares adjacent elements and swaps them if they are in the wrong order.
එය යාබද මූලාංග සංසන්දනය කර ඒවා වැරදි අනුපිළිවෙලක තිබේ නම් ඒවා මාරු කරයි.

3) It splits the list into smaller sublists and sorts each sublist recursively.
එය ලැයිස්තුව කුඩා උප ලැයිස්තු වලට බෙදන අතර සෑම උප ලැයිස්තුවක්ම පුනරාවර්තන ලෙස වර්ග කරයි.

4) It repeatedly finds the largest element and places it in the correct position.
එය නැවත නැවතත් විශාලතම මූලාංග සොයාගෙන එය නිවැරදි ස්ථානයේ තබයි.

5) None of the above.
ඉහත කිසිවක් නොවේ.

Answer: 2)

Bubble Sort Algorithm / Bubble Sort ඇල්ගොරිතම
In each pass, Bubble Sort "bubbles" the largest (or smallest, depending on the order) unsorted element to its correct position by repeatedly swapping adjacent elements if they are out of order.
සෑම ප්‍රවේශයකදීම, Bubble Sort විශාලතම (හෝ කුඩාම, අනුපිළිවෙල අනුව) මූලාංග පිළිවෙලට නොමැති නම් යාබද මූලාංගය නැවත නැවත මාරු කර "බුබුලු කිරීමෙන්" එහි නිවැරදි ස්ථානයේ තබයි.

The process repeats, reducing the number of elements to be checked, as the largest elements are sorted first and settle at the end of the list.
විශාලතම මූලාංග පළමුව වර්ග කර ලැයිස්තුවේ අවසානයේ තබන බැවින්, ක්‍රියාවලිය නැවත නැවතත් සිදු වේ, පරීක්ෂා කළ යුතු මූලාංග ගණන අඩු කරයි.

The algorithm continues until the list is fully sorted.
ලැයිස්තුව සම්පූර්ණයෙන්ම අනුපිළිවෙලට සකස් කරන තෙක් ඇල්ගොරිතම දිගටම පවතී.

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
This describes the Selection Sort algorithm. In Selection Sort, the smallest element in the unsorted portion is identified and swapped with the first unsorted element.
මෙය Selection Sort ඇල්ගොරිතම විස්තර කරයි. තේරීමේ අනුපිළිවෙලෙහි, වර්ග නොකළ කොටසෙහි ඇති කුඩාම මූලාංගය හඳුනාගෙන පළමු වර්ග නොකළ මූලාංගය සමඟ මාරු කරනු ලැබේ.

Option / විකල්ප 2)
This correctly describes Bubble Sort algorithm. Bubble Sort works by comparing adjacent elements in a list and swapping them if they are in the wrong order. This process is repeated for every element in the list until the list is sorted.
මෙය Bubble Sort ඇල්ගොරිතම නිවැරදිව විස්තර කරයි. Bubble Sort ක්‍රියා කරන්නේ ලැයිස්තුවක යාබද මූලාංග සංසන්දනය කිරීමෙන් සහ ඒවා වැරදි පිළිවෙලට තිබේ නම් ඒවා මාරු කිරීමෙනි. ලැයිස්තුව අනුපිළිවෙලට සකස් කරන තෙක් ලැයිස්තුවේ ඇති සෑම අංගයක් සඳහාම මෙම ක්‍රියාවලිය නැවත නැවතත් සිදු කෙරේ.

Option / විකල්ප 3)
This describes the Merge Sort algorithm. Merge Sort is a divide-and-conquer algorithm that splits the list into smaller sublists, sorts them, and then merges the sorted sublists back together.
මෙය Merge Sort ඇල්ගොරිතම විස්තර කරයි. Merge Sort යනු divide-and-conquer ඇල්ගොරිතමයක් වන අතර එය ලැයිස්තුව කුඩා උප ලැයිස්තු වලට බෙදා, ඒවා වර්ග කර, පසුව අනුලේඛනගත උප ලැයිස්තු නැවත එකට ඒකාබද්ධ කරයි.

Option / විකල්ප 4)
This is not entirely wrong but better describes Selection Sort in reverse. While Bubble Sort does "bubble" the largest elements to their correct positions through swaps, the key process in Bubble Sort is the comparison and swapping of adjacent elements, which is missing from this description.
මෙය සම්පූර්ණයෙන්ම වැරදි නොවන නමුත් Selection Sort ප්‍රතිලෝම ලෙස විස්තර කරයි. Bubble Sort විසින් විශාලම මූලාංග ඒවායේ නිවැරදි ස්ථාන වලට "බුබුල කරන" අතර, Bubble Sort හි ප්‍රධාන ක්‍රියාවලිය වන්නේ යාබද මූලාංග සංසන්දනය කිරීම සහ හුවමාරු කිරීම, මෙම විස්තරයෙන් අතුරුදහන් වේ.

Option / විකල්ප 5)
This is incorrect because option 2 correctly describes Bubble Sort.
විකල්ප 2, Bubble Sort නිවැරදිව විස්තර කරන නිසා මෙය වැරදියි.

The correct answer is option 2) It compares adjacent elements and swaps them if they are in the wrong order.
නිවැරදි පිළිතුර විකල්ප 2) එය යාබද මූලාංග සංසන්දනය කර ඒවා වැරදි පිළිවෙලට තිබේ නම් ඒවා මාරු කරයි යන්න වේ.

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

```
a, b = 0, 1
count = 0
while count = 7:
    print(a)
    a, b = b, a + b
    count += 1
```

1) 0	2) 0	3) 0	4) 0	5) Error
1	1	1	1	
1	2	1	1	
2	3	2	2	
3	4	2	3	
5	5	3	5	
8	6	3	7	

Answer : 5)

Explanation / පැහැදිලි කිරීම

The code will produce a syntax error due to an incorrect assignment in the `while loop` condition. Specifically, `while count = 7:` uses a single equals sign (`=`), which is an assignment operator rather than a comparison operator. In Python, `=` is used to assign values, whereas `==` is used to compare values. Since `while count = 7:` attempts to assign `7` to `count` rather than check if `count` is equal to `7`, Python will raise a syntax error. To correct this, the line should be changed to `while count == 7:` if the goal is to create a loop that only executes when `count` is exactly `7`, or modify it as needed to achieve the intended behavior.
`while loop` තත්වයේ වැරදි පැවරුමක් හේතුවෙන් කේතය කාරක රීති දෝෂයක් ඇති කරයි. නිශ්චිතවම, `while count = 7:` තනි සමාන ලකුණක් (`=`) භාවිතා කරයි, එය සංසන්දන ක්‍රියාකරුවෙකුට වඩා පැවරුම් ක්‍රියාකරුවක් වේ. Python හි, අගයන් පැවරීමට `=` භාවිතා කරන අතර, අගයන් සංසන්දනය කිරීමට `==` භාවිතා කරයි. අයසකෑ `while count = 7: countz 7` ට සමාන දැයි පරීක්ෂා කරනවාට වඩා `7 count` වෙත පැවරීමට උත්සාහ කරන බැවින්, පයිතන් කාරක රීති දෝෂයක් මතු කරයි. මෙය නිවැරදි කිරීම සඳහා, ඉලක්කය වන්නේ `count` හරියටම `7` වන විට පමණක් ක්‍රියාත්මක වන ලූපයක් සෑදීම හෝ අපේක්ෂිත හැසිරීම සාක්ෂාත් කර ගැනීමට අවශ්‍ය පරිදි එය වෙනස් කිරීම නම්, කේත පේළිය `while count == 7:` ලෙස වෙනස් කළ යුතු වේ.

Therefore the correct answer is, 5).
එබැවින් නිවැරදි පිළිතුර වන්නේ, 5) වේ.

10. What is a tag in HTML that does not have a closing tag?
HTML හි අවසන් කරන උසුලනයක් නොමැති උසුලනය වනුයේ කුමක්ද?

1) <p> 2) 3) <a> 4) <hr> 5) <head>

Answer: 4)

Explanation / පැහැදිලි කිරීම:

A tag in HTML that does not have a closing tag is called a self-closing tag or void element. These tags do not require an explicit closing tag because they do not enclose any content.
HTML හි වසා දැමීමේ උසුලනයක් නොමැති උසුලනයක් ස්වයං-වසා දැමීමේ උසුලනයක් හෝ හිස් අංගයක් ලෙස හැඳින්වේ. මෙම උසුලනවලට කිසිදු අන්තර්ගතයක් ආවරණය නොකරන බැවින් පැහැදිලි වසා දැමීමේ උසුලනයක් අවශ්‍ය නොවේ.

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

1) <p>
This is a paragraph tag and requires a closing tag (</p>).
මෙය ඡේද උසුලනයක් වන අතර වසා දැමීමේ උසුලනයක් අවශ්‍ය වේ (</p>).

**2) **
This is a bold tag and requires a closing tag ().
මෙය තද පැහැති උසුලනයක් වන අතර වසා දැමීමේ උසුලනයක් අවශ්‍ය වේ ().

3) <a>
This is an anchor tag and requires a closing tag ().
මෙය අධිසම්බන්ධක උසුලනයක් වන අතර වසා දැමීමේ උසුලනයක් අවශ්‍ය වේ ().

4) <hr>
This is a horizontal rule tag that does not require a closing tag.
මෙය වසා දැමීමේ උසුලනයක් අවශ්‍ය නොවන තිරස් රීති උසුලනයකි.

5) <head>
This is a sectioning tag that requires a closing tag (</head>).
මෙය වසා දැමීමේ උසුලනයක් අවශ්‍ය වන කොටස් කිරීමේ උසුලනයකි (</head>).

The <hr> tag is used to create a horizontal line and does not need a corresponding closing tag.
<hr> උසුලනය තිරස් රේඛාවක් නිර්මාණය කිරීමට භාවිතා කරන අතර අනුරූප වසා දැමීමේ උසුලනයක් අවශ්‍ය නොවේ.

The correct answer is / නිවැරදි පිළිතුර වන්නේ,
4) <hr>