සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved]

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

Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 332 MARKING

2025/11/07
ictfromabc
Ravindu Bandaranayake

1. Which of the following consists the correct sequence of Data Processing Life Cycle?
දත්ත සැකසීමේ ජීවන චක්‍රයේ පියවර නිවැරදි අනුපිළිවෙලින් සමන්විත වන්නේ පහත සඳහන් කුමන පිළිතුරේද?

   1) Data input, Data collecting, Data processing, Data storing, Data Output.
   දත්ත ආදානය, දත්ත රැස් කිරීම, දත්ත සැකසීම, දත්ත ගබඩා කිරීම, දත්ත ප්‍රතිදානය.

   2) Data collecting, Data processing, Data storing, Data input, Data Output.
   දත්ත එකතු කිරීම, දත්ත සැකසීම, දත්ත ගබඩා කිරීම, දත්ත ආදානය, දත්ත ප්‍රතිදානය.

   3) Data collecting, Data input, Data processing, Data storing, Data Output.
   දත්ත රැස් කිරීම, දත්ත ආදානය, දත්ත සැකසීම, දත්ත ගබඩා කිරීම, දත්ත ප්‍රතිදානය.

   4) Data input, Data collecting, Data storing, Data processing, Data Output.
   දත්ත ආදානය, දත්ත රැස් කිරීම, දත්ත ගබඩා කිරීම, දත්ත සැකසීම, දත්ත ප්‍රතිදානය.

   5) Data input, Data processing, Data collecting, Data storing, Data Output.
   දත්ත ආදානය, දත්ත සැකසීම, දත්ත රැස් කිරීම, දත්ත ගබඩා කිරීම, දත්ත ප්‍රතිදානය.

   Answer : 3)

   Let's consider each option one by one.
   එක් එක් විකල්පය එකින් එක සලකා බලමු.

   - **Statement / ප්‍රකාශය 1)**
   According to this option, it's false. This sequence is incorrect because "Data collecting" typically comes before "Data input."
   මෙම විකල්පයට අනුව, එය අසත්‍ය වේ. "දත්ත එකතු කිරීම" සාමාන්‍යයෙන් "දත්ත ආදානයට" පෙර එන නිසා මෙම අනුක්‍රමය වැරදිය.

   - **Statement / ප්‍රකාශය 2)**
   According to this option, it's false. This sequence is incorrect because "Data input" should occur before "Data processing".
   මෙම විකල්පයට අනුව, එය අසත්‍ය වේ. "දත්ත සැකසීමට" පෙර "දත්ත ආදානය" සිදුවිය යුතු නිසා මෙම අනුක්‍රමය වැරදිය.

   - **Statement / ප්‍රකාශය 3)**
   According to this option, it's true. The correct sequence in the Data Processing Life Cycle is to first collect data, then input it, process it, store it, and finally produce the output.
   මෙම විකල්පය අනුව, එය සත්‍ය වේ. දත්ත සැකසුම් ජීවන චක්‍රයේ නිවැරදි අනුපිළිවෙල වන්නේ පළමුව දත්ත රැස් කිරීම, පසුව එය ආදානය කිරීම, සැකසීම, ගබඩා කිරීම සහ අවසානයේ ප්‍රතිදානය නිෂ්පාදනය කිරීමයි.

   - **Statement / ප්‍රකාශය 4)**
   According to this option, it's false. This sequence is incorrect because "Data collecting" should come before "Data input."
   මෙම විකල්පයට අනුව, එය අසත්‍ය වේ."දත්ත එකතු කිරීම" "දත්ත ආදානයට" පෙර පැමිණිය යුතු නිසා මෙම අනුපිළිවෙල වැරදිය.

   - **Statement / ප්‍රකාශය 5)**
   According to this option, it's false. This sequence is incorrect because "Data collecting" should occur before "Data input."
   මෙම විකල්පයට අනුව, එය අසත්‍ය වේ."දත්ත ආදානයට" පෙර "දත්ත එකතු කිරීම" සිදුවිය යුතු නිසා මෙම අනුපිළිවෙල වැරදිය.

   Therefore, the correct answer is, **3).** / එබැවින් නිවැරදි පිළිතුර නම්, **3).**

2. Which of the following is NOT a characteristic of information?
තොරතුරු වල ලක්ෂණයක් නොවන්නේ පහත ඒවායින් මොනවාද?

1) Accuracy / නිරවද්‍යතාව
2) Relevance / අදාළත්වය
3) Timeliness / කලානුරූපී බව
4) Completeness / සම්පූර්ණත්වය
5) Complexity / සංකීර්ණත්වය

Answer :5)

Complexity is not considered as a characteristic of information.
සංකීර්ණත්වය තොරතුරු වල ලක්ෂණයක් ලෙස නොසැලකේ.

Accuracy නිරවද්‍යතාව

- Accuracy ensures that information is correct and it has no mistakes. It affects the quality of information.
නිරවද්‍යතාව, තොරතුරු නිවැරදි බව සහ එහි වැරදි නොමැති බව සහතික කරයි. එය තොරතුරු වල ගුණාත්මකභාවයට බලපායි.

Relevance අදාළත්වය

- Data has to be relevant to the user and the purpose it serves. Relevance of the information varies from user to user.
දත්ත පරිශීලකයාට සහ එය සේවය කරන අරමුණට අදාළ විය යුතුය. තොරතුරු වල අදාළත්වය පරිශීලකයාගෙන් පරිශීලකයාට වෙනස් වේ.

Timeliness කලානුරූපී බව

- Validity of information depends on time therefore information must be on time for the purpose it required.
තොරතුරු වලංගුතාව කාලය මත රඳා පවතින බැවින් තොරතුරු අවශ්‍ය අරමුණ සඳහා නියමිත වේලාවට තිබිය යුතුය.

Completeness සම්පූර්ණත්වය

- To make a correct decision, information should contain all the necessary details. Mostly Information is incomplete and the user has to refer other sources before making the decision.
නිවැරදි තීරණයක් ගැනීම සඳහා, තොරතුරු සතුව අවශ්‍ය සියලු විස්තර අඩංගු විය යුතුය. බොහෝ දුරට තොරතුරු අසම්පූර්ණ වන අතර පරිශීලකයාට තීරණය ගැනීමට පෙර වෙනත් මූලාශ්‍ර වෙත යොමු විය යුතුය.

3. What is the 2's complement of -8_{10}?
-8_{10} හි 2 හි අනුපූරකය කුමක්ද?

1) 10001000_2 2) 11111001_2 3) 11111000_2 4) 11111011_2 5) 11011100_2

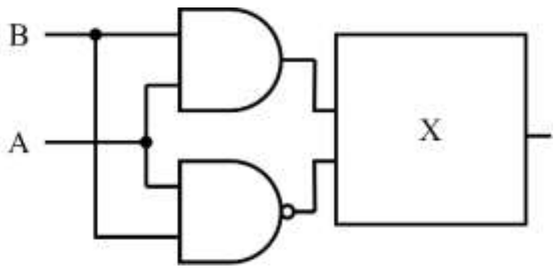
Answer : 3)

Let's follow the calculations / ගණනය කිරීම් අනුගමනය කරමු.

Conversion of 8_{10} to binary 8_{10} ද්විමය බවට පරිවර්තනය කිරීම						
64	32	16	8	4	2	1
0	0	0	1	0	0	0
1's Complement / 1 හි අනුපූරකය						
1	1	1	0	1	1	1
2's Complement / 2 හි අනුපූරකය						
1	1	1	0	1	1	1
+						1
1	1	1	1	0	0	0
1111000_2						

Therefore, the answer is, 3). / එබැවින්, පිළිතුර වන්නේ, 3).

4. What gate should be used in place of X to always get 1 as the output?
සැම විටම ප්‍රතිදානය ලෙස 1 ලබා ගැනීමට X වෙනුවට භාවිතා කළ යුතු ද්වාරය කුමක්ද?



1) OR gate
2) NOT gate
3) NOR gate
4) AND gate
5) XNOR gate

Answer : 1)

Answer 1), This can be solved using the inverse law, NAND is the inverse or the compliment of AND gate so when these are given as inputs to X either the output of the AND gate or the output of the NAND gate will always be one as they are the compliments of each other so when there's an input as well the compliment of that input as another input to X the logical operation that should be done is an OR operation. This is according to the inverse law.
පිළිතුර 1) වේ. මෙය ප්‍රතිලෝම න්‍යාය භාවිතයෙන් විසඳිය හැක. NAND ද්වාරය යනු AND ද්වාරයෙහි ප්‍රතිලෝමය වේ. එබැවින් මේවා X වෙත ආදාන ලෙස ලබාදුන් විට AND ද්වාරයේ ප්‍රතිදානය හෝ NAND ද්වාරයෙහි ප්‍රතිදානය සැම විටම එකක් වනු ඇත. ඒවා එකිනෙකාගේ ප්‍රතිලෝමය වේ. එබැවින් එක් ආදානයක් ඇති විට X වෙත තවත් ආදානයක් ලෙස එම ආදානයේ ප්‍රතිලෝමය ලබාදෙන්නේ නම් සිදු කළ යුතු තාර්කික මෙහෙයුම OR මෙහෙයුමකි. මෙය ප්‍රතිලෝම න්‍යායට අනුරූප වේ.

5. A.(B+C) = (A.B)+(A.C) is an example for ?
A.(B+C) = (A.B)+(A.C) උදාහරණයක් වන්නේ කුමන න්‍යායටද?

1) Commutative law / න්‍යායදේශ න්‍යාය
2) Associative law / සංඝටන න්‍යාය
3) Distributive law / විඝටන න්‍යාය
4) Identity law / සර්වසාම්‍ය න්‍යාය
5) Redundancy law / සමතිරික්ත න්‍යාය

Answer : 3)

Distributive law විඝටන න්‍යාය
$A.(B + C) = (A.B) + (A.C)$
$A + (B.C) = (A + B).(A + C)$

Commutative law න්‍යායදේශ න්‍යාය
$A.B = B.A$
$A + B = B + A$

Associative law සංඝටන න්‍යාය
$(A + B) + C = A + (B + C) = A + B + C$
$(A.B).C = A.(B.C) = A.B.C$

Identity law සර්වසාම්‍ය න්‍යාය
$A + 0 = A$
$A + 1 = 1$
$A.0 = 0$
$A.1 = A$

Redundancy law සමතිරික්ත න්‍යාය
$A + A.B = A$
$A + \overline{A}.B = A + B$

6. What is the main benefit of partitioning a table in a database?
දත්ත සමුදායක වගුවක් කොටස් කිරීමේ ප්‍රධාන වාසිය කුමක්ද?

1) Reducing the need for indexes.
දර්ශක සඳහා අවශ්‍යතාවය අඩු කිරීම.

2) Minimizing the use of primary keys.
ප්‍රාථමික යතුරු භාවිතය අවම කිරීම.

3) Enhancing query performance and manageability of large datasets.
විමසුම් කාර්ය සාධනය සහ විශාල දත්ත කට්ටල කළමනාකරණය කිරීමේ හැකියාව වැඩි දියුණු කිරීම.

4) Simplifying table relationships.
වගු සම්බන්ධතා සරල කිරීම.

5) Allowing unlimited concurrent users.
අසීමිත සමගාමී පරිශීලකයින්ට ඉඩ දීම.

Answer : 3)

Table partitioning is a technique used to divide a large table into smaller, more manageable pieces called partitions.
වගු කොටස් කිරීම යනු විශාල වගුවක් කුඩා, වඩා කළමනාකරණය කළ හැකි කොටස් වලට කොටස් ලෙස බෙදීමට භාවිතා කරන තාක් ෂණයකි.

The main benefits include / ප්‍රධාන ප්‍රතිලාභ අතරට:

- Improved Query Performance / jeäÈhqKq l< úuiqï ld¾h idOkh(

By dividing the data, queries that access only specific partitions run faster, as the database system can skip irrelevant partitions.
දත්ත බෙදීමෙන්, නිශ්චිත කොටස් වලට පමණක් ප්‍රවේශ වන විමසුම් වේගයෙන් ක්‍රියාත්මක වේ, මන්ද දත්ත සමුදා පද්ධතියට අදාළ නොවන කොටස් මඟ හැරිය හැක.

- Better Manageability / වඩා හොඳ කළමනාකරණය කිරීමේ හැකියාව:

Easier to maintain and back up smaller partitions individually instead of handling a single massive table.
තනි දැවැන්ත වගුවක් හැසිරවීම වෙනුවට කුඩා කොටස් තනි තනිව නඩත්තු කිරීම සහ උපස්ථ කිරීම පහසුය.

- Efficient Resource Usage / කාර්යක්ෂම සම්පත් භාවිතය:

Indexing and data access are optimized, especially for large datasets.
සුචිගත කිරීම සහ දත්ත ප්‍රවේශය ප්‍රශස්ත කර ඇත, විශේෂයෙන් විශාල දත්ත කට්ටල සඳහා.

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
Incorrect, partitioning complements indexes but doesn't replace them.
වැරදියි, කොටස් කිරීම දර්ශක සම්පූර්ණ කරයි නමුත් ඒවා ප්‍රතිස්ථාපනය නොකරයි.

Option / විකල්ප 2)
Incorrect, primary keys remain essential. / වැරදියි, ප්‍රාථමික යතුරු අත්‍යවශ්‍ය වේ.

Option / විකල්ප 3)
Correct, partitioning improves performance and simplifies management of large tables.
නිවැරදියි, කොටස් කිරීම කාර්ය සාධනය වැඩි දියුණු කරන අතර විශාල වගු කළමනාකරණය සරල කරයි.

Option / විකල්ප 4)
Incorrect, partitioning doesn't affect relationships.
වැරදියි, කොටස් කිරීම සම්බන්ධතාවලට බලපාන්නේ නැත.

Option / විකල්ප 5)
Incorrect, unrelated to partitioning. / වැරදියි, කොටස් කිරීමට සම්බන්ධ නැත.

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

7. Which of the following scenarios best demonstrates a "Many-to-Many" relationship between two tables in a relational database?
සම්බන්ධක දත්ත ගබඩාවක වගු දෙකක් අතර "බහු-බහු" සම්බන්ධතාවයක් වඩාත් හොඳින් පෙන්නුම් කරන්නේ පහත දැක්වෙන කුමන අවස්ථාද?

    1) A customer can place many orders, and each order is placed by one customer.
    පාරිභෝගිකයෙකුට බොහෝ ඇණවුම් කළ හැකි අතර, සෑම ඇණවුමක්ම එක් පාරිභෝගිකයෙකු විසින් ලබා දෙනු ලැබේ.

    2) <u>An author can write many books, and a book can be written by many authors.</u>
    ලේඛකයෙකුට බොහෝ පොත් ලිවිය හැකි අතර, බොහෝ කතුවරුන් විසින් පොතක් ලිවිය හැකිය.

    3) An employee can be assigned to one department, and a department can have many employees.
    සේවකයෙකු එක් දෙපාර්තමේන්තුවකට පැවරිය හැකි අතර, දෙපාර්තමේන්තුවකට බොහෝ සේවකයින් සිටිය හැකිය.

    4) A student can have one mentor, and a mentor can have many students.
    ශිෂ්‍යයෙකුට එක් උපදේශකයෙකු සිටිය හැකි අතර උපදේශකයෙකුට බොහෝ සිසුන් සිටිය හැකිය.

    5) A company has one CEO, and the CEO manages many departments.
    සමාගමකට එක් ප්‍රධාන විධායක නිලධාරියෙකු සිටින අතර ප්‍රධාන විධායක නිලධාරියා බොහෝ දෙපාර්තමේන්තු කළමනාකරණය කරයි.

Answer: 2)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
This is a One-to-Many (1:M) relationship (one customer can place multiple orders, but each order belongs to only one customer).
මෙය ඒක-බහු (1:M) සම්බන්ධතාවයකි (එක් ගනුදෙනුකරුවෙකුට බහු ඇණවුම් කළ හැකි නමුත්, සෑම ඇණවුමක්ම එක් පාරිභෝගිකයෙකුට පමණක් අයත් වේ).

Option / විකල්ප 2)
This is a clear example of a Many-to-Many (M:N) relationship (one author can write multiple books, and one book can be co-authored by multiple authors).
මෙය බහු-බහු (M:N) සම්බන්ධතාවයකට පැහැදිලි උදාහරණයකි (එක් කතුවරයෙකුට පොත් කිහිපයක් ලිවිය හැකි අතර, එක් පොතක් බහු කතුවරුන් විසින් සම-කර්තෘත්වයට පත් කළ හැකිය).

Option / විකල්ප 3)
This is a One-to-Many (1:M) relationship (one department can have many employees, but each employee belongs to one department).
මෙය ඒක-බහු (1:M) සම්බන්ධතාවයකි (එක් දෙපාර්තමේන්තුවකට බොහෝ සේවකයින් සිටිය හැකි නමුත්, සෑම සේවකයෙකුටම එක් දෙපාර්තමේන්තුවකට අයත් වේ).

Option / විකල්ප 4)
This is another One-to-Many (1:M) relationship (one mentor can have multiple students, but each student has only one mentor).
මෙය තවත් ඒක-බහු (1:M) සම්බන්ධතාවයකි (එක් උපදේශකයෙකුට බහු සිසුන් සිටිය හැකි නමුත්, සෑම සිසුවෙකුටම එක් උපදේශකයෙකු පමණි).

Option / විකල්ප 5)
This is a One-to-Many (1:M) relationship (one CEO can manage multiple departments, but there is only one CEO).
මෙය ඒක-බහු (1:M) සම්බන්ධතාවයකි (එක් ප්‍රධාන විධායක නිලධාරියෙකුට බහු දෙපාර්තමේන්තු කළමනාකරණය කළ හැකි නමුත්, එක් ප්‍රධාන විධායක නිලධාරියෙකු පමණි).

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

8. What will be the output if `x = -2` and `y = 5` ?
`x = -2` සහ `y = 5` නම් ප්‍රතිදානය කුමක් වේවිද?

```
if x > 0:
  if y > 0:
    print("Both x and y are positive")
  else:
    print("x is positive, y is not")
else:
  print("x is not positive")
```

1) `Both x and y are positive`
2) `x is positive, y is not`
3) <u>`x is not positive`</u>
4) An error will occur. / දෝෂයක් ඇතිවේ.
5) The code will run indefinitely. / කේතය අවසානයක් නොමැතිව ක්‍රියාත්මක වේ.

Answer : 3)

If `x = -2` and `y = 5`, the code first checks if `x > 0`. Since `x` is `-2`, which is not greater than `0`, this condition is false. Therefore, the code skips the `nested if` statement and directly executes the **`else`** block, printing `"x is not positive"`.
`x = -2` සහ `y = 5` නම්, කේතය පළමුව පරීක්ෂා කරන්නේ `x > 0` ද යන්නයි. `x` යනු `-2` වේ. එය `0` ට වඩා වැඩි නොවන බැවින්, මෙම කොන්දේසිය අසත්‍ය වේ. එම නිසා, කේතය `nested if` ප්‍රකාශය මඟ හරින අතර සෘජුවම `else` ක්‍රියාත්මක කරයි, `"x is not positive"` මුද්‍රණය කරයි.

9. What is the output of the below python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
rows = 5
for i in range(1, rows + 1):
    for j in range(1, i + 1):
        print(j % 2, end=" ")
    print()
```

1)
```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

2)
```
1
1 0
1 0 1
1 0 1 0
1 0 1 0 1
```

3)
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

4)
```
2
2 4
2 4 6
2 4 6 8
2 4 6 8 10
```

5)
```
1
1 3
1 3 5
1 3 5 7
1 3 5 7 9
```

Answer : 2)

Let's consider the code. / ඉහත කේතය සලකා බලමු.

- **Outer loop**
  `for i in range(1, rows + 1)`
  This loop runs from 1 to rows (in this case, 5), determining the number of rows to print.
  මෙම ලූපය 1 සිට පේළි දක්වා දිව යයි (මෙම අවස්ථාවේදී, 5), මුද්‍රණය කිරීමට පේළි ගණන තීරණය කරයි.

- **Inner loop**
  `for j in range(1, i + 1)`
  This loop runs for each row `i`, printing `j % 2` for each column. `j % 2` gives the remainder when `j` is divided by 2, alternating between 0 and 1.
  මෙම ලූපය සෑම පේළියක් සඳහාම ක්‍රියාත්මක වේ `i` එක් එක් තීරුව සඳහා `j % 2` මුද්‍රණය කරයි. `j % 2` සහ 1 අතර මාරුවෙන් මාරුවට `j`, 2 න් බෙදූ විට ඉතිරිය ලබා දෙයි.

- **`end=" "`**
  `print()` ensures that the values are printed on the same line with a space in between, and the `print()` with no arguments starts a new line after each row.
  `print()` මඟින් අගයන් අතර පරතරයක් ඇතිව එකම පේළියක මුද්‍රණය කර ඇති බව සහතික කරන අතර, තර්ක නොමැතිව `print()` එක් එක් පේළියට පසුව නව රේඛාවක් ආරම්භ කරයි.

For `i = 1`

- The inner loop runs from j = 1 to j = 1.
- අභ්‍යන්තර ලූපය j = 1 සිට j = 1 දක්වා දිව යයි.
- `j % 2 = 1 % 2 = 1.`
- Output for this row: `1`.
- මෙම පේළිය සඳහා ප්‍රතිදානය: `1`.

For `i = 2`

- The inner loop runs from j = 1 to j = 2.
  අභ්‍යන්තර ලූපය j = 1 සිට j = 2 දක්වා දිව යයි.
- `j = 1 → j % 2 = 1 % 2 = 1.`
- `j = 2 → j % 2 = 2 % 2 = 0.`
- Output for this row: `1 0`.
  මෙම පේළිය සඳහා ප්‍රතිදානය: `1 0`.

For `i = 3`

- The inner loop runs from j = 1 to j = 3.
  අභ්‍යන්තර ලූපය j = 1 සිට j = 3 දක්වා දිව යයි.
- `j = 1 → j % 2 = 1 % 2 = 1.`
- `j = 2 → j % 2 = 2 % 2 = 0.`
- `j = 3 → j % 2 = 3 % 2 = 1.`
- Output for this row: `1 0 1`.
  මෙම පේළිය සඳහා ප්‍රතිදානය: `1 0 1`.

For `i = 4`

- The inner loop runs from j = 1 to j = 4.
  අභ්‍යන්තර ලූපය j = 1 සිට j = 4 දක්වා දිව යයි.
- `j = 1 → j % 2 = 1 % 2 = 1.`
- `j = 2 → j % 2 = 2 % 2 = 0.`
- `j = 3 → j % 2 = 3 % 2 = 1.`
- `j = 4 → j % 2 = 4 % 2 = 0.`

- Output for this row: `1 0 1 0`.
  මෙම පේළිය සඳහා ප්‍රතිදානය: `1 0 1 0`.

For i = 5

- The inner loop runs from j = 1 to j = 5.
  අභ්‍යන්තර ලූපය j = 1 සිට j = 5 දක්වා දිව යයි.
- `j = 1 → j % 2 = 1 % 2 = 1.`
- `j = 2 → j % 2 = 2 % 2 = 0.`
- `j = 3 → j % 2 = 3 % 2 = 1.`
- `j = 4 → j % 2 = 4 % 2 = 0.`
- `j = 5 → j % 2 = 5 % 2 = 1.`
- Output for this row: `1 0 1 0 1`.
  මෙම පේළිය සඳහා ප්‍රතිදානය: `1 0 1 0 1`.

Final Output / අවසාන ප්‍රතිදානය

```
1
1 0
1 0 1
1 0 1 0
1 0 1 0 1
```

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

10. Which of the following statements about Python lists is true?
පයිතන් ලැයිස්තු පිළිබඳ පහත සඳහන් කුමන ප්‍රකාශය සත්‍ය ද?

1) Lists are immutable.
ලැයිස්තු වෙනස් කළ නොහැකි ය.

2) Lists can contain elements of different types.
ලැයිස්තුවල විවිධ වර්ගවල මූලාංග අඩංගු විය හැක.

3) Lists cannot be nested within other lists.
වෙනත් ලැයිස්තු තුළ ලැයිස්තු ලෙස පැවතිය නොහැක.

4) Lists are hashable. ලැයිස්තු hashable වේ.

5) Lists can only store integers.
ලැයිස්තු වලට ගබඩා කළ හැක්කේ පූර්ණ සංඛ්‍යා පමණි.

Answer : 2)

Let's consider each option one by one.
එක් එක් විකල්පය එකින් එක සලකා බලමු.

- **Option / විකල්පය 1)**
This statement is incorrect. Python lists are mutable, meaning you can change their contents (add, remove, or modify elements) after they are created. If lists were immutable, their contents would be fixed and unchangeable.
මෙම ප්‍රකාශය වැරදියි. පයිතන් ලැයිස්තු වෙනස් කළ හැකි ය, එයින් අදහස් වන්නේ ඒවා නිර්මාණය කිරීමෙන් පසුව ඔබට ඒවායේ අන්තර්ගතය වෙනස් කළ හැකිය (මූලාංග එකතු කිරීම, ඉවත් කිරීම හෝ වෙනස් කිරීම). ලැයිස්තු වෙනස් කළ නොහැකි නම්, ඒවායේ අන්තර්ගතය ස්ථාවර සහ වෙනස් කළ නොහැකි වනු ඇත.

- **Option / විකල්පය 2)**
This statement is true. Python lists are highly flexible and can contain elements of different types within the same list. For example, a list can have integers, strings, and other objects all together: [1, 'hello', 3.14].
මෙම ප්‍රකාශය සත්‍යයකි. පයිතන් ලැයිස්තු ඉතා නම්‍යශීලී වන අතර එකම ලැයිස්තුවේ විවිධ වර්ගවල මූලාංග අඩංගු විය හැක. උදාහරණයක් ලෙස, ලැයිස්තුවකට පූර්ණ සංඛ්‍යා, string සහ අනෙකුත් වස්තු සියල්ල එකට තිබිය හැක: [1, 'hello', 3.14].

- **Option / විකල්පය 3)**
This statement is incorrect. Python lists can be nested, meaning a list can contain other lists as its elements. For example: [[1, 2], [3, 4], [5]] is a valid list of lists.
මෙම ප්‍රකාශය වැරදියි. පයිතන් ලැයිස්තු කැදලි කළ හැක, එනම් ලැයිස්තුවක එහි මූලද්‍රව්‍ය ලෙස වෙනත් ලැයිස්තු අඩංගු විය හැක. උදාහරණයක් ලෙස: [[1, 2], [3, 4], [5]] යනු වලංගු ලැයිස්තු ලැයිස්තුවකි.

- **Option / විකල්පය 4)**
This statement is incorrect. Lists are not hashable because they are mutable. Hashable objects need to have a fixed hash value that does not change during their lifetime. Lists, being mutable, do not have this property.
මෙම ප්‍රකාශය වැරදියි. ලැයිස්තු වෙනස් කළ හැකි බැවින් ඒවා හැෂ් කළ නොහැක. හෂ් කළ හැකි වස්තූන්ට ඔවුන්ගේ ජීවිත කාලය තුළ වෙනස් නොවන ස්ථාවර හැෂ් අගයක් තිබිය යුතුය. ලැයිස්තු, විකෘති වීම,

- **Option / විකල්පය 5)**
This statement is incorrect. Python lists can store a wide range of data types, not just integers. They can contain strings, floats, other lists, and more.
මෙම ප්‍රකාශය වැරදියි. පයිතන් ලැයිස්තු වලට පූර්ණ සංඛ්‍යා පමණක් නොව පුළුල් පරාසයක දත්ත වර්ග ගබඩා කළ හැක. ඒවාට string, ඉපිලෙන, වෙනත් ලැයිස්තු සහ තවත් දේ අඩංගු විය හැක.

The correct answer is **2)**.
නිවැරදි පිළිතුර **2)** වේ.