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 328 MARKING

2025/11/03
ictfromabc
Ravindu Bandaranayake

1. Cloud computing offers storage space. What type of cloud computing is this?
   වළාකුළු පරිගණනය මගින් ආචයන ඉඩ පහසුකම් ලබා දෙයි. මෙය වළාකුළු පරිගණනයේ කුමන ආකාරයකට අයත් වේද?

   1) IaaS 2) PaaS 3) SaaS 4) FaaS 5) DaaS

   Answer : 1)

   Cloud computing offers a variety of options.
   වළාකුළු පරිගණනය මගින් විවිධ වූ ආකාර ලබා දෙයි.

   IaaS - Infrastructure as a Service
   This means providing various essential facilities like offering storage space to the user.
   පරිශීලකයාට ආචයන ඉඩ පහසුකම් වැනි අත්‍යවශ්‍ය පහසුකම් ලබා දීම මෙමගින් අදහස් කරයි.

   PaaS - Platform as a Service
   This means creating the necessary environment for the application to run.
   යෙදුම් ධාවනය සඳහා අවශ්‍ය වටපිටාවක් නිර්මාණය කර දීම මෙමගින් අදහස් වෙයි.

   SaaS - Software as a Service
   This means that various software can be used from within the respective cloud computing.
   විවිධ මෘදුකාංග අදාළ වළාකුළු පරිගණනය තුළ සිටම භාවිතයට අවස්ථාව ලබා දීම මෙමගින් අදහස් වේ.

2. Which of the following accurately represents the relationship between bits, bytes, and storage capacity?
   බිටු, බයිට සහ ගබඩා ධාරිතාව අතර සම්බන්ධය නිවැරදිව නියෝජනය කරන්නේ පහත සඳහන් කුමක්ද?

   1) 1 byte = 8 bits; 1 kilobyte (KB) = 1024 bytes
   2) 1 byte = 16 bits; 1 kilobyte (KB) = 1000 bytes
   3) 1 byte = 4 bits; 1 kilobyte (KB) = 1000 bytes
   4) 1 byte = 10 bits; 1 kilobyte (KB) = 1024 bytes
   5) 1 byte = 8 bits; 1 kilobyte (KB) = 1000 bytes

   Answer : 1)

   1 byte is made up of 8 bits. This is a fundamental relationship in computing. A bit can be either a 0 or a 1, and when you group 8 of them together, you get a byte.
   1 බයිටයක් සෑදී ඇත්තේ බිටු 8 කිනි. මෙය පරිගණනයේ මූලික සම්බන්ධතාවයකි. බිටු එකක් 0 හෝ 1 විය හැකි අතර, ඔබ ඒවායින් 8 ක් එකට එකතු කළ විට, ඔබට බයිට් එකක් ලැබේ.

   Now, when it comes to larger units like kilobytes (KB), there's a convention to use powers of 2. In this convention, 1 kilobyte is equal to 1024 bytes, not 1000 bytes. This is because in binary systems, which computers use, it's easier to work with powers of 2.
   දැන්, කිලෝබයිට් (KB) වැනි විශාල ඒකකයක් ගත් කල, 2 හි බල භාවිතා කිරීමේ සම්මුතියක් පවතී. මෙම සම්මුතියේදී, කිලෝබයිට් 1 බයිට් 1024 ට සමාන වේ, බයිට් 1000කට නොවේ. මන්දයත් පරිගණක වලට ද්විමය පද්ධතිවල 2හි බලයන් සමඟ වැඩ කිරීම පහසුය.

   So, the correct relationship is: 1 byte = 8 bits, and 1 kilobyte (KB) = 1024 bytes.
   එබැවින්, නිවැරදි පිළිතුර: 1 byte = 8 bits; 1 kilobyte (KB) = 1024 bytes.

3. If A is represented by 65 in ASCII code, what is the octal value representeing K?
ASCII කේත ක්‍රමයේදී A අක්ෂරය නිරූපණය වන්නේ 65 භාවිතයෙන් නම් K අක්ෂරය නිරූපණය කරන අෂ්ටමය සංඛ්‍යාව කුමක්ද?

1) 122_8 2) $\underline{113}_8$ 3) 102_8 4) 75_8 5) 221_8

Answer : 2)

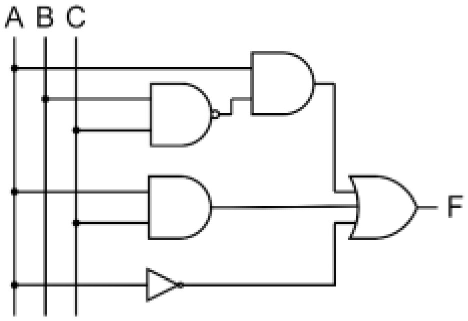
According to the linear procedure, 75 is obtained for K.
රේඛීය පටිපාටියට අනුව ක් සඳහා 75 ලැබේ.

Divide the decimal number by 8. Write down the remainder (in reverse order).
දශම සංඛ්‍යාව 8 න් බෙදන්න. ඉතිරිය (ප්‍රතිලෝම අනුපිළිවෙලින්) ලියන්න.

8	75	
8	9	-3 ↑
8	1	-1
	0	-1

So, the octal value representing the letter 'K' in the ASCII code is 113_8.
එමනිසා ASCII කේතයේ 'K' අක්ෂරය නියෝජනය කරන අෂ්ටක අගය 113_8 වේ.

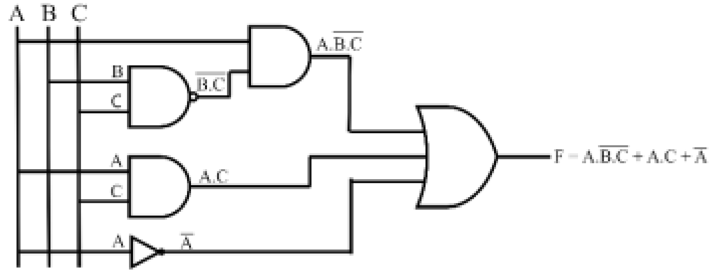
4. What is the output when the Boolean expression representing the logic circuit shown below is trivialized?
පහත දක්වා ඇති තාර්කික පරිපථය නියෝජනය කරන බූලියානු ප්‍රකාශනය සුළු කළ විට ප්‍රතිදානය කුමක්ද?



1) 0
2) $\underline{1}$
3) $\overline{A}$
4) $A.\overline{B.C}$
5) $A.(\overline{B.C}) + AC$

Answer : 2)

First we need to get the Boolean expression for this logic circuit.
පළමුව මෙම තාර්කික පරිපථයට අදාල බූලිය ප්‍රකාශනය ලබා ගත යුතුය.



$A\overline{BC} + AC + \overline{A}$	
$A(\overline{B} + \overline{C}) + AC + \overline{A}$	De morgan's law ද මුවාර් ප්‍රමේයය
$\overline{B} + \overline{C} + AC + \overline{A}$	Reduncl ancy law සමතිරික්ත න්‍යාය
$\overline{B} + \overline{C} + C + \overline{A}$	Reduncl ancy law සමතිරික්ත න්‍යාය
$\overline{B} + 1 + \overline{A}$	Inverse law ප්‍රතිලෝම න්‍යාය
$1 + \overline{A}$	Identity law සර්වසාම්‍ය න්‍යාය
1	Identity law සර්වසාම්‍ය න්‍යාය

The answer is 1 when the boolean expression is trivialized.
බූලිය ප්‍රකාශනය සුළු කළ විට පිළිතුර 1 වේ.

5. Which of the following Boolean expression/s give(s) the same output?
පහත දැක්වෙන බූලියන් ප්‍රකාශන අතරින් එකම ප්‍රතිදානය ලබා දෙන්නේ කුමක්ද?

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

B. $(\overline{BD} + BA)A + A1$

C. $A(\overline{DA} + 1A) + DC$

1) A only. / A පමණි.
2) A and C only. / A හා C පමණි.
3) All of the above. / ඉහත සියල්ලම.
4) A and B only. / A හා B පමණි.
5) None of the above. / ඉහත කිසිවක් නොවේ.

Answer : 4)

Let's trivialize the following boolean expressions.
පහත බූලිය ප්‍රකාශන සුළු කරමු.

A. $\mathbf{AB(\overline{C} + C) + A\overline{B}}$

$AB(\overline{C} + C) + A\overline{B}$	
$AB1 + A\overline{B}$	Complement Law / ප්‍රතිලෝම න්‍යාය
$AB + A\overline{B}$	Identity Law / සර්වසාම්‍ය න්‍යාය
$A(B + \overline{B})$	Distributive Law / විඝටණ න්‍යාය
$A1$	Complement Law / ප්‍රතිලෝම න්‍යාය
A	Identity Law / සර්වසාම්‍ය න්‍යාය

B. $\mathbf{(\overline{BD} + BA)A + A1}$

$(\overline{BD} + BA)A + A1$	
$(\overline{B} + \overline{D} + BA)A + A1$	Demorgan Law / ද මුවාර් ප්‍රමේය
$(\overline{B} + \overline{D} + BA)A + A$	Identity Law / සර්වසාම්‍ය න්‍යාය
A	Redundancy Law / සමතිරික්ත න්‍යාය

C. $\mathbf{A(\overline{DA} + 1A) + DC}$

$A(\overline{DA} + 1A) + DC$	
$A(\overline{D} + \overline{A} + 1A) + DC$	Demorgan Law / විඝටණ න්‍යාය
$A(\overline{D} + \overline{A} + A) + DC$	Identity Law / සර්වසාම්‍ය න්‍යාය
$A(\overline{D} + 1) + DC$	Complement Law / ප්‍රතිලෝම න්‍යාය
$A1 + DC$	Identity Law / සර්වසාම්‍ය න්‍යාය
$A + DC$	Identity Law / සර්වසාම්‍ය න්‍යාය

Therefore the answer is 4, / එබැවින් පිළිතුර 4 වේ.

6. Which of the following ensures data integrity in a relational database?
සම්බන්ධක දත්ත සමුදායක දත්ත අඛණ්ඩතාව සහතික කරන්නේ පහත සඳහන් මොනවාද?

1) Using a primary key and foreign key relationship.
ප්‍රාථමික යතුරක් සහ ආගන්තුක යතුරු සම්බන්ධතාවයක් භාවිතා කිරීම.

2) Normalizing the database beyond the third normal form.
තෙවන ප්‍රමතකරණ ආකෘතියෙන් ඔබ්බට දත්ත සමුදාය ප්‍රමතකරණ කිරීම.

3) Using a single table for all records
සියලුම වාර්තා සඳහා තනි වගුවක් භාවිතා කිරීම.

4) Indexing every column in every table
සෑම වගුවකම සෑම තිරුවක්ම සුචිගත කිරීම.

5) Avoiding NULL values in any column
ඕනෑම තිරුවක NULL අගයන් වළක්වා ගැනීම.

Answer: 1)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
A primary key uniquely identifies each record in a table, and a foreign key establishes relationships between tables, ensuring referential integrity. This is a fundamental mechanism for maintaining data integrity in relational databases.
ප්‍රාථමික යතුරක් වගුවක ඇති සෑම වාර්තාවක්ම අනන්‍ය ලෙස හඳුනා ගන්නා අතර ආගන්තුක යතුරක් වගු අතර සම්බන්ධතා ඇති කරයි, යොමු අඛණ්ඩතාව සහතික කරයි. මෙය සම්බන්ධතා දත්ත සමුදායන්හි දත්ත අඛණ්ඩතාව පවත්වා ගැනීම සඳහා වන මූලික යාන්ත්‍රණයකි.

Option / විකල්ප 2)
While normalization helps reduce redundancy and maintain consistency, it doesn't directly ensure data integrity beyond third normal form.
ප්‍රමතකරණය අතිරික්තය අඩු කිරීමට සහ අනුකූලතාව පවත්වා ගැනීමට උපකාරී වන අතර, එය තෙවන ප්‍රමත අවස්ථාවෙන් ඔබ්බට දත්ත අඛණ්ඩතාව සෘජුවම සහතික නොකරයි.

Option / විකල්ප 3)
This approach contradicts the principles of relational databases and makes data integrity difficult to manage.
මෙම ප්‍රවේශය සම්බන්ධතා දත්ත සමුදායේ මූලධර්මවලට පටහැනි වන අතර දත්ත අඛණ්ඩතාව කළමනාකරණය කිරීමට අපහසු වේ.

Option / විකල්ප 4)
Indexes improve query performance but are unrelated to data integrity.
දර්ශක විමසුම් කාර්ය සාධනය වැඩි දියුණු කරන නමුත් දත්ත අඛණ්ඩතාවට සම්බන්ධ නොවේ.

Option / විකල්ප 5)
While avoiding NULLs might reduce certain issues, it's not a direct way to ensure overall data integrity.
NULL වලින් වැළකී සිටීමෙන් යම් යම් ගැටළු අඩු විය හැකි අතර, එය සමස්ත දත්ත අඛණ්ඩතාව සහතික කිරීමට සෘජු මාර්ගයක් නොවේ.

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

7. Which database model stores data in a single table without any relationships between records?
වාර්තා අතර කිසිදු සම්බන්ධතාවක් නොමැතිව තනි වගුවක දත්ත ගබඩා කරන දත්ත සමුදා ආකෘතිය කුමක්ද?

1) Relational model / සම්බන්ධතා ආකෘතිය
2) Hierarchical model / ධූරාවලි ආකෘතිය
3) <u>Flat file system / පැතලි ගොනු පද්ධතිය</u>
4) Network model / ජාල ආකෘතිය
5) Object-relational model / වස්තුව-සම්බන්ධතා ආකෘතිය

Answer : 3)

Explanation / පැහැදිලි කිරීම

The flat file system stores data in a single table without any relationships between records. In this model, data is organized in a simple, tabular structure where all information is stored in rows and columns, typically within a single file, like a spreadsheet. There are no links or connections between different rows, and each record is treated as an independent entity. This makes it straightforward and easy to use for smaller, less complex datasets, but it lacks the ability to efficiently handle relationships or enforce data integrity.
ඒක ගොනු පද්ධතිය වාර්තා අතර සබඳතාවයකින් තොරව තනි වගුවක දත්ත ගබඩා කරයි. මෙම ආකෘතිය තුළ, දත්ත සරල, වගු ව්‍යුහයක් තුළ සංවිධානය කර ඇති අතර එහිදී සියලුම තොරතුරු පේළි සහ තීරුවල ගබඩා කර ඇත, සාමාන්‍යයෙන් පැතුරුම්පතක් වැනි තනි ගොනුවක් තුළ. විවිධ පේළි අතර සබැඳි හෝ සම්බන්ධතා නොමැති අතර, සෑම වාර්තාවක්ම ස්වාධීන ආයතනයක් ලෙස සලකනු ලැබේ. මෙය කුඩා, අඩු සංකීර්ණ දත්ත කට්ටල සඳහා සරල සහ පහසු භාවිතා කරයි, නමුත් එයට සම්බන්ධතා කාර්යක්ෂමව හැසිරවීමට හෝ දත්ත අඛණ්ඩතාව බලාත්මක කිරීමට හැකියාවක් නොමැත.

This approach is suitable for scenarios where the data is not interrelated or when working with simple, standalone datasets. However, as the volume and complexity of data grow, flat file systems become inefficient and difficult to maintain, leading to redundancy and inconsistency. This is why modern database systems often adopt relational or other advanced models to better manage relationships between data.
මෙම ප්‍රවේශය දත්ත අන්තර් සම්බන්ධිත නොවන අවස්ථා සඳහා හෝ සරල, ස්වාධීන දත්ත කට්ටල සමඟ වැඩ කරන විට සුදුසු වේ. කෙසේ වෙතත්, දත්තවල පරිමාව සහ සංකීර්ණත්වය වර්ධනය වන විට, ඒක ගොනු පද්ධති අකාර්යක්ෂම සහ නඩත්තු කිරීමට අපහසු වන අතර, අතිරික්තය සහ නොගැලපීම සඳහා හේතු වේ. නූතන දත්ත සමුදා පද්ධති බොහෝ විට දත්ත අතර සම්බන්ධතා වඩා හොඳින් කළමනාකරණය කිරීම සඳහා සම්බන්ධතා හෝ වෙනත් උසස් මාදිලි භාවිතා කරන්නේ එබැවිනි.

Therefore the Correct answer is 3) / එබැවින් නිවැරදි පිළිතුර 3)

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

```
x="ictfromabc"
print(x[1:7])
```

1) `ictfro` 2) `ctfro` 3) `ctfroma` 4) `ictfrom` 5) `ctfrom`

Answer : 5)

The code extracts a substring from the string `x`, starting from index `1` and ending at index `6` (as the upper limit in slicing is non-inclusive). The string `"ictfromabc"` is indexed as follows:
කේතය `x` string එකෙන් උප string එකක් උපුටා ගනී, `1` වන දර්ශකයෙන් ආරම්භ වී `6` වන දර්ශකයෙන් අවසන් වේ (Slicing දී ඉහළ සීමාව ඇතුළත් නොවන බැවින්). `"ictfromabc"` string එක පහත පරිදි සුචිගත කර ඇත:

```
Index:  0 1 2 3 4 5 6 7 8 9
String: i c t f r o m a b c
```

So, `x[1:7]` refers to the characters from index `1` to `6`, which are:
එබැවින්, `x[1:7]` `1` සිට `6` දක්වා වූ අක්ෂර වෙත යොමු කරයි, ඒවා නම්:

`"ctfrom"`

The output of the code will be 5) `ctfrom`.
කේතයේ ප්‍රතිදානය වනුයේ 5) `ctfrom`.

9. What does `oct(10)` return in python?
පයිතන් වලදී `oct(10)` හි ප්‍රතිදානය කුමක් ද?

1) `'10'` 2) `'0o10'` 3) `'0o12'` 4) `'12'` 5) `'0o1010'`

Answer : 3)

In Python, the " oct() " function converts an integer to its octal representation as a string. The octal representation of a number is prefixed with "0o".
Python හි, "oct()" ශ්‍රිතය මඟින් පූර්ණ සංඛ්‍යාවක් එහි අෂ්ටක නිරූපණය string ලෙස පරිවර්තනය කරයි. අංකයක අෂ්ටක නිරූපණය සඳහා "0o" උපසර්ගය සඳහන් කර ඇත.

Correct Answer / නිවැරදි පිළිතුර : '0o12'

"oct (10)" in Python converts the decimal number 10 to its octal representation, which is 12 in octal notation. The result is returned as a string prefixed with "0o" to indicate it is in octal format. Hence, the correct result is '0o12'.
Python හි "oct(10)" දශම අංකය 10 එහි අෂ්ටක නිරූපණය බවට පරිවර්තනය කරයි, එය අෂ්ටක අංකනයෙහි 12 වේ. ප්‍රතිඵලය අෂ්ටක ආකෘතියෙන් බව දැක්වීමට "0o" සමඟ උපසර්ගයක් සහිත string ලෙස ආපසු එවනු ලැබේ. එබැවින්, නිවැරදි ප්‍රතිඵලය වන්නේ '0o12' ය.

10. What will be the final output of list1 after executing the following code?
පහත කේතය ක්‍රියාත්මක කිරීමෙන් පසු list1 හි අවසාන ප්‍රතිදානය කුමක් වේවිද?

```
list1 = [1, 2, 3]
list1.insert(2, 'a')
list1.append('b')
list1.extend(['c', 'd'])
print(list1)
```

1) `[1, 2, 3, 'a', 'b', 'c', 'd']`
2) `[1, 2, 'a', 3, 'b', 'c', 'd']`
3) `[1, 2, 'a', 3, 'b', ['c', 'd']]`
4) `[1, 2, 3, 'a', 'b', ['c', 'd']]`
5) `[1, 2, 3, ['a', 'b', 'c', 'd']]`

Answer : 2)

Initially, the list is `[1, 2, 3]`.
මුලදී, ලැයිස්තුව `[1, 2, 3]` වේ.

When 'a' is inserted at index `2`, the list becomes `[1, 2, 'a', 3]`.
දර්ශකය 2 හි 'a' ඇතුළත් කළ විට, ලැයිස්තුව `[1, 2, 'a', 3]` බවට පත් වේ.

Then, the append operation adds 'b' to the end, resulting in `[1, 2, 'a', 3, 'b']`.
ඉන්පසුව, ඇමුණුම් මෙහෙයුම අවසානයේ 'b' එකතු කරයි, එහි ප්‍රතිඵලයක් ලෙස `[1, 2, 'a', 3, 'b']`.

Finally, the extend function adds the elements 'c' and 'd' to the list, giving the final result `[1, 2, 'a', 3, 'b', 'c', 'd']`.
අවසාන වශයෙන්, දිගු ශ්‍රිතය ලැයිස්තුවට 'c' සහ 'd' මූලද්‍රව්‍ය එකතු කරයි, අවසාන ප්‍රතිඵලය `[1, 2, 'a', 3, 'b', 'c', 'd']` ලබා දෙයි.

Therefore, the correct answer is 2) `[1, 2, 'a', 3, 'b', 'c', 'd']`.
එබැවින්, නිවැරදි පිළිතුර වන්නේ 2) `[1, 2, 'a', 3, 'b', 'c', 'd']`.