

සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved]

Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc
 Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc
 Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc
 Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc
 Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc
 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 565 MARKING

2026/06/28
ictfromabc
Ravindu Bandaranayake

1. A digital clock uses BCD encoding to represent each digit of the current time. At one point, the clock displays the time 09:47. The microcontroller inside reads the digits separately and converts each digit into its 4-bit binary-coded decimal (BCD) representation.

ඩිජිටල් ඔරලෝසුවක් වත්මන් කාලයේ සෑම ඉලක්කමක්ම නිරූපණය කිරීම සඳහා BCD කේතනය භාවිතා කරයි. එක් අවස්ථාවකදී, ඔරලෝසුව 09:47 වේලාව පෙන්වයි. ඇතුළත ඇති ක්ෂුද්‍ර පාලකයා ඉලක්කම් වෙන වෙනම කියවා සෑම ඉලක්කමක්ම එහි බිටු 4 binary-coded decimal (BCD) නිරූපණය බවට පරිවර්තනය කරයි.

Which of the following correctly represents the full BCD encoding of the time 09:47?

(Note: Each digit is encoded separately using 4-bit BCD. Format is HHMM, i.e., H1 H2 M1 M2.) 09:47 වේලාවේ සම්පූර්ණ BCD කේතනය නිවැරදිව නියෝජනය කරන්නේ පහත සඳහන් ඒවායින් කුමක්ද?

(සටහන: සෑම ඉලක්කමක්ම බිටු 4 BCD භාවිතයෙන් වෙන වෙනම කේතනය කර ඇත. ආකෘතිය HHMM වනම්, H1 H2 M1 M2 වේ.)

- 1) 1001 0000 0100 0111 2) 1001 0000 0111 0100 3) 0000 1001 0111 0100
 4) 0000 1001 0100 0111 5) 1001 0000 0110 1000

Answer : 4)

Explanation / පැහැදිලි කිරීම

BCD is a method of representing decimal numbers (0–9) in binary form, where each decimal digit is represented by 4 binary bits.

BCD යනු ද්විමය ආකාරයෙන් දශම සංඛ්‍යා (0–9) නිරූපණය කිරීමේ ක්‍රමයකි, එහිදී සෑම දශම සංඛ්‍යාංකයක්ම ද්විමය බිටු 4 කින් නිරූපණය කෙරේ.

Digits / ඉලක්කම් → 0, 9, 4, 7

BCD encoding (4 bits per digit) / BCD කේතනය ඉලක්කමකට බිටු 4:

- 0 = 0000
- 9 = 1001
- 4 = 0100
- 7 = 0111

Correct BCD sequence / නිවැරදි BCD අනුපිළිවෙල: 0000 1001 0100 0111

The final answer is Option 4

අවසාන පිළිතුර වන්නේ විකල්පය 4

2. Which of the following is the SOP expression which can be obtained by the given K-map?
 දී ඇති කාන්ට් සිතියම මගින් ලබා ගත හැකි SOP ප්‍රකාශනය පහත සඳහන් ඒවායින් කුමක්ද?

AB \ D	00	01	11	10
0	1	1	1	1
1	1	1	1	0

- 1) $A+B+D$
- 2) $\bar{A}+B+\bar{D}$
- 3) $A+B$
- 4) $A+\bar{B}+\bar{D}$
- 5) $A+\bar{D}$

Answer: 2)

Let's obtained an expression from the above K-map. / ඉහත කාන්ට් සිතියමෙන් ප්‍රකාශනයක් ලබා ගනිමු.

AB \ D	00	01	11	10
0	1	1	1	1
1	1	1	1	0

$\bar{A}$ B $\bar{D}$

Expression : $\bar{A} + B + \bar{D}$

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

3. Consider the following features. / පහත ලක්ෂණ සලකා බලන්න.

- File size is significantly reduced.
ගොනු ප්‍රමාණය සැලකිය යුතු ලෙස අඩු වේ.
- Some data is permanently removed.
සමහර දත්ත ස්ථිරවම ඉවත් වේ.
- Useful for images, audio, and video files.
රූප, ශ්‍රව්‍ය සහ දෘශ්‍ය ගොනු සඳහා ප්‍රයෝජනවත් වේ.

Which file compression type matches the above features?
 ඉහත අනුරූප ලක්ෂණ ඇති ගොනු සම්පීඩන වර්ගය කුමක්ද?

- 1) Lossless compression / ගොනුවේ දත්ත වලට හානි නොවන සම්පීඩනය.
- 2) Lossy compression / ගොනුවේ දත්ත වලට හානි වන සම්පීඩනය.
- 3) Hybrid compression / දෙමුහුම් සම්පීඩනය.
- 4) Fractal Compression / ඛණ්ඩක සම්පීඩනය.
- 5) Dictionary-Based Compression / Dictionary පදනම් වූ සම්පීඩනය.

Answer: 2)

Review of the options / විකල්ප විශ්ලේෂණය:

Option / විකල්ප 1)

- No data is lost during compression.
සම්පීඩනය අතරතුර කිසිදු දත්තයක් නැති නොවේ.
- The original data can be perfectly reconstructed from the compressed file.
සම්පීඩිත ගොනුවෙන් මුල් දත්ත පරිපූර්ණ ලෙස ප්‍රතිනිර්මාණය කළ හැකිය.
Examples: ZIP, PNG
- Does not match the feature of "some data being permanently removed."
"සමහර දත්ත ස්ථිරවම ඉවත් කිරීම" යන විශේෂාංගයට නොගැලපේ.

Option / විකල්ප 2)

- Some data is permanently removed to reduce file size.
ගොනු ප්‍රමාණය අඩු කිරීම සඳහා සමහර දත්ත ස්ථිරවම ඉවත් කරනු ලැබේ.
- The exact original file cannot be fully restored.
නිශ්චිත මුල් ගොනුව සම්පූර්ණයෙන්ම ප්‍රතිසාධනය කළ නොහැක.
- Commonly used for images, audio, and video files.
රූප, ශ්‍රව්‍ය සහ දෘශ්‍ය ගොනු සඳහා බහුලව භාවිතා වේ.
Examples: JPEG (images), MP3 (audio), MP4 (video).
- This option matches all the features described in the question.
මෙම විකල්පය ප්‍රශ්නයේ විස්තර කර ඇති සියලුම විශේෂාංග සමඟ ගැලපේ.

Option / විකල්ප 3)

- A combination of both lossy and lossless methods.
ගොනුවේ දත්ත වලට හානි නොවන සහ ගොනුවේ දත්ත වලට හානි වන ක්‍රම දෙකෙහිම සංයෝජනයකි.
Used in certain complex algorithms but not a typical compression type associated directly with the features mentioned.
ඇතැම් සංකීර්ණ ඇල්ගොරිතමවල භාවිතා වේ, නමුත් සඳහන් කර ඇති විශේෂාංග සමඟ සෘජුවම සම්බන්ධ වන සාමාන්‍ය සම්පීඩන වර්ගයක් නොවේ.
- Does not fully match the given features.
ලබා දී ඇති විශේෂාංග සමඟ සම්පූර්ණයෙන්ම නොගැලපේ.

Option / විකල්ප 4)

- A specialized form of lossy compression using mathematical fractals.
ගණිතමය ඛණ්ඩක භාවිතා කරන දත්ත වලට හානි වන සම්පීඩනයේ විශේෂිත ආකාරයකි.
- Mostly theoretical or used in specific advanced image formats.
බොහෝ විට න්‍යායාත්මක හෝ විශේෂිත උසස් රූප ආකෘතිවල භාවිතා වේ.
- Not commonly associated with general use cases for images, audio, or video.
රූප, ශ්‍රව්‍ය හෝ විඩියෝ සඳහා සාමාන්‍ය භාවිත අවස්ථා සමඟ බහුලව සම්බන්ධ නොවේ.

Option / විකල්ප 5)

- A type of lossless compression that uses a dictionary of common strings for encoding.
කේතනය සඳහා පොදු strings dictionary එකක් භාවිතා කරන හානි නොවන සම්පීඩන වර්ගයකි.
- Used in formats like ZIP and LZW.
ZIP සහ LZW වැනි ආකෘතිවල භාවිතා වේ.
- Does not match the feature of data being permanently removed.
දත්ත ස්ථිරවම ඉවත් කිරීමේ විශේෂාංගයට නොගැලපේ.

So, the correct answer is 2).

එබැවින්, නිවැරදි පිළිතුර වනුයේ 2) යි.

It is the only type of compression where some data is permanently removed to achieve a smaller file size, making it suitable for media files like images, audio, and video.

රූප, ශ්‍රව්‍ය සහ දෘශ්‍ය වැනි මාධ්‍ය ගොනු සඳහා සුදුසු වන පරිදි කුඩා ගොනු ප්‍රමාණයක් ලබා ගැනීම සඳහා සමහර දත්ත ස්ථිරවම ඉවත් කරන එකම සම්පීඩන වර්ගය

4. What is the cardinality and degree of the following Customer-Order relationship?

පහත සඳහන් පාරිභෝගික-ඇණවුම් සම්බන්ධතාවයේ ගණනයනාවය සහ මට්ටම කුමක්ද?

CustomerID	Name
101	Akon
102	Bkon
103	Ckon

OrderID	OrderDate	Amount (Rs)	CustomerID
201	2025-01-10	2500.00	101
202	2025-01-12	300.00	101
203	2025-01-11	150.00	102
204	2025-01-15	2000.00	103
205	2025-01-13	400.00	103

- 1) 1:M, 3 2) 1:M, 2 3) M:N, 3 4) M:N, 2 5) 1:1, 2

Answer : 2)

Explanation / පැහැදිලි කිරීම

The correct answer is 2) 1:M, 2.

නිවැරදි පිළිතුර වන්නේ 2) 1:M, 2 වේ.

In the given relationship, each Customer can have multiple Orders (e.g., CustomerID 101 has two orders: 201 and 202), which represents a 1:M (One-to-Many) relationship. The degree of a relationship is determined by the number of entities involved. Here, the relationship involves two entities: Customer and Order, making the degree 2. The cardinality is 1:M, and the degree is 2.

ලබා දී ඇති සම්බන්ධතාවයේ දී, සැම පාරිභෝගිකයෙකුටම බහු ඇණවුම් තිබිය හැකිය (උදා: CustomerID 101 හි ඇණවුම් දෙකක් ඇත: 201 සහ 202) එය 1:M (One-to-Many) සම්බන්ධතාවයක් නියෝජනය කරයි. සම්බන්ධතාවයක මට්ටම තීරණය වන්නේ සම්බන්ධ වන භූතාර්ථ ගණන අනුව ය. මෙහිදී, සම්බන්ධතාවයට භූතාර්ථ දෙකක් ඇතුළත් වේ: පාරිභෝගිකයා සහ ඇණවුම, මට්ටම 2 බවට පත් කරයි. ගණනයනාවය 1:M වන අතර මට්ටම 2 වේ.

5. Consider the following Python program.

පහත පයිතන් කේතය සලකා බලන්න

```
def order_summary(item, quantity=1, price=100):
    print(f"{quantity} units of {item} cost {quantity * price}.")
order_summary("Laptop", price=500)
```

Which of the following is the output of this program?

මෙම වැඩසටහනේ ප්‍රතිදානය පහත සඳහන් ඒවායින් කුමක් ද?

- 1) 1 units of Laptop cost 100.
 2) 1 units of Laptop cost 500.
 3) 1 units of Laptop cost 1000.
 4) 500 units of Laptop cost 500.
 5) 1 units of Laptop cost Unknown.

Answer : 2)

The function `order_summary(item, quantity=1, price=100)` has two default arguments: `order_summary(item, quantity=1, price=100)` ශ්‍රිතයට පෙරනිමි තර්ක දෙකක් ඇත:

- quantity defaults to 1 if not provided / සපයා නොමැති නම් ප්‍රමාණය පෙරනිමි 1 වේ.
- price defaults to 100 if not provided / සපයා නොමැති නම් මිල පෙරනිමි 1 වේ.

In the order_summary("Laptop", price=500) call:
order_summary("Laptop", price=500) ඇමතීම:

- The item parameter is set to "Laptop" / item පරාමිතිය "Laptop" ලෙස සකසා ඇත.
- The price parameter is overridden to 500 / price පරාමිතිය 500 ලෙස ප්‍රතිස්ථාපනය කර ඇත.
- Since quantity is not explicitly provided, it retains its default value of 1.
quantity පැහැදිලිව සපයා නොමැති බැවින්, එය එහි පෙරනිමි අගය 1 රඳවා ගනී.

The function calculates quantity * price which is $1 * 500 = 500$.

මුත් quantity * price ගණනය කරයි, එය $1 * 500 = 500$ වේ.

It prints: "1 units of Laptop cost 500."

එය "1 units of Laptop cost 500." මුද්‍රණය කරයි.

Correct Answer / නිවැරදි පිළිතුර : 2)

6. What is the most popular development model of the following system development models?
පහත සඳහන් තොරතුරු පද්ධති සංවර්ධන ආකෘති වලින් ඉතාමත් ජනප්‍රිය සංවර්ධන ආකෘතිය කුමක් ද?
- 1) Spiral model / ස්පර්ලාකාර ආකෘතිය
 - 2) Object Oriented model / වස්තු නැඹුරු ආකෘතිය
 - 3) Waterfall model / දියඇලි ආකෘතිය
 - 4) Incremental Development model / වර්ධනාත්මක සංවර්ධන ආකෘතිය
 - 5) Rapid Development model / ශීඝ්‍ර යෙදුම් සංවර්ධන ආකෘතිය

Answer: 3)

Explanation for each option

එක් එක් විකල්පය සඳහා පැහැදිලි කිරීම

1) Spiral model

A risk-driven process model that combines elements of both iterative and waterfall models. It's used for large, complex projects but isn't as widely adopted as other models due to its complexity and cost. පුනරාවර්තන සහ දියඇලි ආකෘති දෙකෙහිම මූලද්‍රව්‍ය ඒකාබද්ධ කරන අවදානම් මත පදනම් වූ ක්‍රියාවලි ආකෘතියකි. එය විශාල, සංකීර්ණ ව්‍යාපෘති සඳහා භාවිතා කරන නමුත් එහි සංකීර්ණත්වය සහ පිරිවැය හේතුවෙන් අනෙකුත් මාදිලි තරම් පුළුල් ලෙස භාවිතා නොවේ.

2) Object Oriented model

Refers to a programming paradigm rather than a development model. While object-oriented design is extremely popular in programming, it doesn't refer to a specific system development model. සංවර්ධන ආකෘතියකට වඩා ක්‍රමලේඛන ආදර්ශයට යොමු කරයි. වස්තු-නැඹුරු නිර්මාණය ක්‍රමලේඛනයේදී අභියෝගී ජනප්‍රිය වුවද, එය විශේෂිත පද්ධති සංවර්ධන ආකෘතියකට යොමු නොවේ.

3) Waterfall model

A linear sequential development approach, where progress flows downwards through phases like a waterfall. It is one of the earliest and most widely known development models. Despite some limitations, it remains popular due to its simplicity and structured approach.

දිය ඇල්ලක් වැනි අදියර හරහා ප්‍රගතිය පහළට ගලා යන රේඛීය අනුක්‍රමික සංවර්ධන ප්‍රවේශයකි. එය පැරණිතම සහ වඩාත් පුළුල් ලෙස දන්නා සංවර්ධන ආකෘතිවලින් එකකි. සමහර සීමාවන් තිබියදීත්, එහි සරල බව සහ ව්‍යුහාත්මක ප්‍රවේශය හේතුවෙන් එය ජනප්‍රියව පවතී.

4) Incremental Development model

A model where the system is developed and delivered in small, functional segments. It's gaining popularity, especially in Agile methodologies, but historically it hasn't been as dominant as the waterfall model. පද්ධතිය කුඩා, ක්‍රියාකාරී කොටස් වලින් සංවර්ධනය කර බෙදා හරින ලද ආකෘතියකි. එය ජනප්‍රිය වෙමින් පවති, විශේෂයෙන් Agile ක්‍රමවේදවල, නමුත් ඓතිහාසිකව එය දියඇලි ආකෘතිය තරම් ප්‍රමුඛ වී නැත.

5) Rapid Development model

Focuses on quick development and iteration, often through the use of prototypes. Models like Rapid Application Development (RAD) are becoming more popular with the rise of Agile methodologies. බොහෝ විට මූලාශ්‍ර කළ හැකි ඉක්මන් සංවර්ධනය සහ පුනරාවර්තනය කෙරෙහි අවධානය යොමු කරයි. වේගවත් යෙදුම් සංවර්ධනය (RAD) වැනි ආකෘති Agile ක්‍රමවේදවල නැගීමත් සමඟ වඩාත් ජනප්‍රිය වෙමින් පවතී.

While modern development trends are moving towards more iterative and incremental approaches like Agile, the Waterfall model remains historically the most popular and widely used development model. නවීන සංවර්ධන ප්‍රවණතා Agile වැනි වඩාත් පුනරාවර්තන සහ වර්ධන ප්‍රවේශයන් කරා ගමන් කරන අතර, දිය ඇල්ල ආකෘතිය ඓතිහාසික වශයෙන් වඩාත්ම ජනප්‍රිය සහ බහුලව භාවිතා වන සංවර්ධන ආකෘතිය ලෙස පවතී.

7. There are three main types of arrays in PHP. The types of those arrays that are given below are respectively,

PHP හි ප්‍රධාන ආරා වර්ග තුනක් ඇත. පහත දී ඇති ආරා වන් පිළිවෙලින් වේ

```
$arr1 = array("apple", "banana", "cherry");
$arr2 = array("name" => "John", "age" => 30, "city" => "New York");
$arr3 = array(
    array("apple", "banana", "cherry"),
    array("John", "Doe"),
    array("age" => 30, "city" => "New York")
);
```

- 1) IndexedArrays, MultiDimensionalArrays, AssociatedArrays
- 2) IndexedArrays, AssociatedArrays, MultidimensionalArrays
- 3) AssociatedArrays, IndexedArrays, MultiDimensionalArrays
- 4) AssociatedArrays, MultiDimensionalArrays, IndexedArrays
- 5) MultiDimensionalArrays, IndexedArrays, AssociatedArrays

Answer : 2)

In PHP, arrays can be categorized into three types: Indexed Arrays, Associative Arrays, and Multidimensional Arrays.

PHP හි, අරා වර්ග තුනකට වර්ග කළ හැක: ඒවා සුචිත අරා, ආශ්‍රිත අරා සහ බහුමාන අරා වේ.

Indexed Arrays

Indexed arrays are the simplest form of arrays where each element is accessed using a numerical index. The indices start at 0.

සුචිත අරා (Indexed arrays) යනු සංඛ්‍යාත්මක දර්ශක භාවිතයෙන් එක් එක් මූලාංග වෙත ප්‍රවේශ වන අරා වල සරලම ආකාරයයි. දර්ශක 0 න් ආරම්භ වේ.

ex:

```
$fruits = array("Apple", "Banana", "Cherry");
echo $fruits[0]; // Outputs: Apple
echo $fruits[1]; // Outputs: Banana
echo $fruits[2]; // Outputs: Cherry
```

Associative Arrays

Associative arrays use named keys that you assign to them. This allows for more descriptive data storage. ආශ්‍රිත අරා (Associative arrays) වලදී ඒවාට පවරන නම් කරන ලද යතුරු භාවිතා කරයි. මෙය වඩාත් විස්තරාත්මක දත්ත ගබඩා කිරීමට ඉඩ සලසයි.

ex:

```
$person = array("first_name" => "John", "last_name" => "Doe", "age" => 30);
echo $person["first_name"]; // Outputs: John
echo $person["last_name"]; // Outputs: Doe
echo $person["age"]; // Outputs: 30
```

Multidimensional Arrays

Multidimensional arrays are arrays that contain other arrays. These can be thought of as arrays of arrays and are useful for representing more complex data structures.

බහුමාන අරා (Multidimensional arrays) යනු වෙනත් අරාවන් අඩංගු අරාවකි. මේවා arrays of arrays ලෙස සැලකිය හැකි අතර වඩාත් සංකීර්ණ දත්ත ව්‍යුහයන් නියෝජනය කිරීමට ප්‍රයෝජනවත් වේ.

ex:

```
$matrix = array(
    array(1, 2, 3),
    array(4, 5, 6),
    array(7, 8, 9)
);
echo $matrix[0][0]; // Outputs: 1
echo $matrix[1][2]; // Outputs: 6
echo $matrix[2][1]; // Outputs: 8
```

Therefore, for the given PHP code, the correct answer is 2).

එබැවින්, ලබා දී ඇති PHP කේතය සඳහා, නිවැරදි පිළිතුර 2) වේ.

8. Which HTML tag from following does not require an end tag?

පහත දැක්වෙන කුමන HTML උසුලනයද අවසන් උසුලනයක් අවශ්‍ය නොවේද?

- 1) 2) <title> 3) 4)
 5)

Answer : 4)

This question asks which HTML tag does not require an end tag. In HTML, most tags have opening and closing tags, but some are self-closing or void elements, meaning they do not require a separate closing tag.

මෙම ප්‍රශ්නය අසන්නේ කුමන HTML උසුලනයද අවසන් උසුලනයක් අවශ්‍ය නොවන බවයි. HTML හි, බොහෝ උසුලන වලට විවෘත සහ වසා දැමීමේ උසුලනයක් ඇත, නමුත් සමහරක් ස්වයං-වසා දැමීමේ හෝ හිස් මූලාංග වේ, එනම් ඒවාට වෙනම වසා දැමීමේ උසුලනයක් අවශ්‍ය නොවේ.

Option / විකල්ප 1)

 requires an end tag () to define the end of the font style.

 අකුරු විලාසයේ අවසානය අර්ථ දැක්වීමට අවසන් උසුලනයක් () අවශ්‍ය වේ.

Option / විකල්ප 2)

<title> requires an end tag (</title>) to indicate the title of the document.

<title> ලේඛනයේ මාතෘකාව දැක්වීමට අවසන් උසුලනයක් (</title>) අවශ්‍ය වේ.

Option / විකල්ප 3)

 requires an end tag () to close an ordered list.

 අනුපිළිවෙළක් සහිත ලැයිස්තුවක් වැසීමට අවසන් උසුලනයක් () අවශ්‍ය වේ.

Option / විකල්ප 4)

 is a line break tag and does not require a closing tag because it is a void element.

 යනු රේඛා බිඳීමේ උසුලනයක් වන අතර එය හිස් මූලාංගයක් බැවින් වසා දැමීමේ උසුලනයක් අවශ්‍ය නොවේ.

Option / විකල්ප 5)

 requires an end tag () to stop bold formatting.

 තද හැඩතල ගැන්වීම නැවැත්වීමට අවසන් උසුලනයක් () අවශ්‍ය වේ.

So, the correct answer is Option 4, as
 does not require an end tag.

මේ අනුව, නිවැරදි පිළිතුර වන්නේ විකල්ප 4 වන අතර,
 අවසන් උසුලනයක් අවශ්‍ය නොවන බැවිනි.

9. What are the core tasks are carried out in data management?
දත්ත කළමනාකරණයේදී සිදුවන වැදගත් කාර්යයන් වන්නේ?

- 1) Data validation / දත්ත වලංගුකරණය.
- 2) Storing data / දත්ත ගබඩා කිරීම.
- 3) Protecting processed data / සැකසූ දත්ත ආරක්ෂා කිරීම.
- 4) User Access / පරිශීලකයාට ප්‍රවේශ පහසුකම් සැපයීම.
- 5) All of the above / ඉහත සියල්ලම.

Answer: 5)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

- 1) Data validation / දත්ත වලංගුකරණය:**

Ensuring that the data collected or entered is accurate, complete, and consistent.

එකතු කරන ලද හෝ ඇතුළත් කළ දත්ත නිවැරදි, සම්පූර්ණ සහ ස්ථාවර බව සහතික කිරීම.

- 2) Storing data / දත්ත ගබඩාකිරීම:**

Organizing and storing data in a structured manner, typically within databases or data warehouses, for easy retrieval and analysis.

දත්ත සමූදායන් හෝ දත්ත ගබඩා තුළ, පහසුවෙන් ලබා ගැනීම සහ විශ්ලේෂණය සඳහා ව්‍යුහගත ආකාරයෙන් දත්ත සංවිධානය කිරීම සහ ගබඩා කිරීම

- 3) Protecting processed data / සැකසූ දත්ත ආරක්ෂා කිරීම:**

Implementing measures to safeguard data against unauthorized access, alteration, or loss, often through encryption, access controls, and backup systems.

බොහෝ විට ගුප්ත කේතනය, ප්‍රවේශ පාලන සහ උපස්ථ පද්ධති හරහා අනවසර ප්‍රවේශ, වෙනස් කිරීම් හෝ නැතිවීම් වලට එරෙහිව දත්ත ආරක්ෂා කිරීමට පියවර ක්‍රියාත්මක කිරීම.

- 4) User access / පරිශීලකයාට ප්‍රවේශ පහසුකම් සැපයීම:**

Managing permissions and controlling who has access to specific data based on their role or level of authorization, to maintain data security and privacy.

දත්ත ආරක්ෂාව සහ පෞද්ගලිකත්වය පවත්වා ගැනීම සඳහා අවසර කළමනාකරණය සහ ඔවුන්ගේ කාර්යභාරය හෝ අවසර මට්ටම මත පදනම්ව නිශ්චිත දත්තවලට ප්‍රවේශය ඇති අය පාලනය කිරීම.

So, the correct answer would be: 5) All above.

එබැවින්, නිවැරදි පිළිතුර වනුයේ: 5) ඉහත සියල්ල ය.

10. National Internet Service Provider (ISP) networks are connected to one another by private switching network is called,

ජාතික අන්තර්ජාල සේවා සපයන්නාගේ (ISP) ජාල පුද්ගලික මාරුකිරීමේ ජාලයක් ලෙසින් එකිනෙක සම්බන්ධ කර ඇත්තේ,

- 1) Peering Points
- 2) Network Access Points / ජාල ප්‍රවේශ ස්ථාන
- 3) National ISP / ජාතික ISP
- 4) Data Centers / දත්ත මධ්‍යස්ථාන
- 5) None of the above / ඉහත කිසිවක් නොවේ

Answer: 1)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම

1) Peering Points

Peering points are locations where different Internet Service Providers (ISPs) connect and exchange traffic directly with each other. This direct connection allows for the efficient and cost-effective transfer of data without having to pass through a third-party network. Peering can happen in both private and public settings. In a private setting, two ISPs set up a direct link between their networks, which is known as a private peering point. This option accurately describes the private switching network that connects national ISP networks.

Peering points යනු විවිධ අන්තර්ජාල සේවා සපයන්නන් (ISPs) එකිනෙකා සමඟ සෘජුවම සම්බන්ධ වී ගමනාගමනය හුවමාරු කරන ස්ථාන වේ. මෙම සෘජු සම්බන්ධතාවය තෙවන පාර්ශවය ජාලයක් හරහා ගමන් කිරීමකින් තොරව දත්ත කාර්යක්ෂමව සහ ලාභදායී ලෙස මාරු කිරීමට ඉඩ සලසයි. පුද්ගලික සහ පොදු සැකසුම් දෙකෙහිම පිරිසිදු සිදු විය හැක. පුද්ගලික සැකසුමකදී, අන්තර්ජාල සේවා සපයන්නන් දෙදෙනෙකු ඔවුන්ගේ ජාල අතර සෘජු සම්බන්ධයක් සකසයි, එය පුද්ගලික පිරිසිදු පොයින්ට් එකක් ලෙස හැඳින්වේ. මෙම විකල්පය ජාතික (ISP) ජාල සම්බන්ධ කරන පුද්ගලික මාරු කිරීමේ ජාලය නිවැරදිව විස්තර කරයි.

2) Network Access Points (NAPs) / ජාල ප්‍රවේශ ස්ථාන (NAPs)

Network Access Points are major public interconnection points where multiple ISPs come together to exchange traffic. They were crucial in the early days of the Internet to facilitate inter-ISP connectivity. NAPs are typically more public and open compared to private peering points, and they serve as a hub for traffic exchange among many ISPs. While NAPs are important for network connectivity, they do not specifically describe the private switching networks used for direct ISP connections.

ජාල ප්‍රවේශ ස්ථාන යනු ගමනාගමනය හුවමාරු කර ගැනීම සඳහා බහු අන්තර්ජාල සේවා සපයන්නන් එකතු වන ප්‍රධාන පොදු අන්තර් සම්බන්ධතා ස්ථාන වේ. inter-ISP සම්බන්ධතාව පහසු කිරීම සඳහා අන්තර්ජාලයේ මුල් දිනවල ඒවා තිරණාත්මක විය. පුද්ගලික සමානාත්මතා ස්ථාන හා සසඳන විට භ්ෂ සාමාන්‍යයෙන් වඩාත් පොදු සහ විවෘත වන අතර ඒවා බොහෝ ISPs අතර ගමනාගමන හුවමාරුව සඳහා කේන්ද්‍රස්ථානයක් ලෙස සේවය කරයි. NAPs ජාල සම්බන්ධතාව සඳහා වැදගත් වන අතර, ඒවා සෘජු ISP සම්බන්ධතා සඳහා භාවිතා කරන පුද්ගලික මාරු කිරීමේ ජාල නිශ්චිතව විස්තර නොකරයි.

3) National ISP / ජාතික ISP

A National ISP provides Internet services across a large geographical area, often the entire country. National ISPs may own extensive infrastructure and connect to other ISPs at various points, including both public NAPs and private peering points. However, a National ISP itself is not a private switching network; rather, it is an entity that uses these networks.

ජාතික ISP විශාල භූගෝලීය ප්රදේශයක් හරහා අන්තර්ජාල සේවා සපයයි, බොහෝ විට මුළු රටම. ජාතික අන්තර්ජාල සේවා සපයන්නන් හට පුළුල් යටිතල පහසුකම් හිමි විය හැකි අතර පොදු භ්ෂ සහ පුද්ගලික සමානාත්මතා ස්ථාන යන දෙකම ඇතුළුව විවිධ ස්ථානවල වෙනත් ISP වෙත සම්බන්ධ විය හැක. කෙසේ වෙතත්, ජාතික ISP යනු පුද්ගලික මාරු කිරීමේ ජාලයක් නොවේ. ඒ වෙනුවට, එය මෙම ජාල භාවිතා කරන ආයතනයකි.

4) Data Centers / දත්ත මධ්‍යස්ථාන

Data centers are facilities that house computer systems and associated components, such as telecommunications and storage systems. They provide the physical infrastructure for hosting servers, storing data, and running applications. While data centers are essential for various Internet services and can host interconnection points, they are not specifically the private switching networks that connect ISPs.

දත්ත මධ්‍යස්ථාන යනු විදුලි සංදේශ සහ ගබඩා පද්ධති වැනි පරිගණක පද්ධති සහ ඒ ආශ්‍රිත උපාංග තැන්පත් කර ඇති පහසුකම් වේ. ඔවුන් සන්නායක සේවාදායකයන්, දත්ත ගබඩා කිරීම සහ යෙදුම් ධාවනය සඳහා භෞතික යටිතල පහසුකම් සපයයි. විවිධ අන්තර්ජාල සේවා සඳහා දත්ත මධ්‍යස්ථාන අත්‍යවශ්‍ය වන අතර අන්තර් සම්බන්ධතා ලක්ෂ්‍යවලට සන්නායකත්වය සැපයිය හැකි අතර, ඒවා විශේෂයෙන්ම ISPs සම්බන්ධ කරන පුද්ගලික මාරුවීම් ජාලයන් නොවේ.

5) None of the Above / ඉහත කිසිවක් නොවේ

This option would imply that none of the given choices accurately describe the private switching network connecting national ISP networks. However, as explained, the correct term is "Peering Points".

මෙම විකල්පයෙන් ඇඟවෙන්නේ ලබා දී ඇති තේරීම් කිසිවක් ජාතික ISP ජාල සම්බන්ධ කරන පුද්ගලික මාරු කිරීමේ ජාලය නිවැරදිව විස්තර නොකරන බවයි. කෙසේ වෙතත්, පැහැදිලි කර ඇති පරිදි, නිවැරදි යෙදුම "Peering Points" වේ.

The correct answer is 1) Peering Points, as they refer to the private switching networks where national ISP networks interconnect directly to exchange traffic efficiently. This direct interconnection improves performance and reduces the cost of data transfer between different ISPs.

නිවැරදි පිළිතුර වන්නේ 1) Peering Points ඔවුන් ජාතික ISP ජාල කාර්යක්ෂමව ගමනාගමනය හුවමාරු කර ගැනීම සඳහා සෘජුවම අන්තර් සම්බන්ධිත පුද්ගලික මාරු කිරීමේ ජාල වෙත යොමු කරයි. මෙම සෘජු අන්තර් සම්බන්ධතාවය කාර්ය සාධනය වැඩි දියුණු කරන අතර විවිධ ISPs අතර දත්ත හුවමාරු කිරීමේ පිරිවැය අඩු කරයි.

1. (a) Consider the following optimized version of the Bubble Sort algorithm:

පහත දැක්වෙන පරිපූර්ණ කරන ලද ඛණ්ඩ කේටි ඇල්ගොරිතමයේ පයිතන් කේතය සලකා බලන්න:

```
def bubble_sort_modified(data):
    n = len(data)
    for i in range(n):
        is_sorted = True
        for j in range(0, n - i - 1):
            if data[j] > data[j + 1]:
                data[j], data[j + 1] = data[j + 1], data[j]
                is_sorted = False
        if is_sorted:
            break
    return data

nums = [1, 2, 5, 4, 8]
result = bubble_sort_modified(nums)
```

- i. In the first pass of the inner loop (when $i=0$), how many comparisons will be made if the input list has 5 elements?

ආදාන ලැයිස්තුවේ අයිතම 5 ක් ඇති විට, අභ්‍යන්තර උපයේ පළමු වටයේදී (එනම් $i=0$ වන විට) සංසන්දනය කිරීම් කීයක් සිදු වේද?

4

- ii. Explain the mathematical reason behind using $n - i - 1$ in the inner `range()` instead of just $n - 1$. What is the benefit of this?

අභ්‍යන්තර `range()` එක තුළ $n - 1$ වෙනුවට $n - i - 1$ භාවිතා කිරීම පිටුපස ඇති ගණිතමය හේතුව පැහැදිලි කරන්න. මෙහි වාසිය කුමක්ද?

In Bubble Sort, after each pass i , the i -th largest element is guaranteed to be at its correct final position at the end of the list. Therefore, we don't need to check the last i elements in subsequent passes. This reduces the number of comparisons, improving efficiency.

ඛණ්ඩ කේටි වලදී, සෑම වටයකටම (i) පසුව, ලැයිස්තුවේ ඇති විශාලතම අගයන් ලැයිස්තුවේ අගට ස්ථිරවම ගමන් කරයි. එමනිසා, ඊළඟ වට වලදී අවසානයේ ඇති එම අයිතම නැවත පරීක්ෂා කිරීම අවශ්‍ය නොවේ. මෙය සංසන්දනය කිරීම් ගණන අඩු කර කාර්යක්ෂමතාව වැඩි කරයි.

- iii. If the input list is `[1, 2, 3, 4, 5]`, how many times will the outer loop run before the function returns the result? Justify your answer using the code.

ආදාන ලැයිස්තුව `[1, 2, 3, 4, 5]` නම්, ශ්‍රිතය ප්‍රතිඵලය ලබා දීමට පෙර ඛණ්ඩ උපය වාර කීයක් ක්‍රියාත්මක වේද? කේතය ඇසුරින් ඔබේ පිළිතුර සාධාරණීකරණය කරන්න.

The outer loop will run once (one full pass of the inner loop).
ඛණ්ඩ උපය එක් වරක් පමණක් ක්‍රියාත්මක වේ.

Reason: Since the list is already sorted, the `if data[j] > data[j + 1]` condition is never met, so `is_sorted` remains True. The `if is_sorted: break` statement then terminates the outer loop immediately.

හේතුව: ලැයිස්තුව දැනටමත් සකස් කර ඇති බැවින් `if data[j] > data[j + 1]` කොන්දේසිය කිසිවිටෙකත් සපුරාලන්නේ නැත. එවිට `is_sorted` විචල්‍යය True ලෙසම පවතී. `if is_sorted: break` විධානය මගින් ඛණ්ඩ උපය වහාම නවත්වනු ලබයි.

(b) A sports club has a list of players and their scores stored as tuples:

ක්‍රීඩා සමාජයක ක්‍රීඩකයින් සහ ඔවුන් ලබාගත් ලකුණු tuples ලෙස ලැයිස්තුගත ගබඩා කර ඇත:

```
players = [("Amal", 45), ("Nimal", 78), ("Bimal", 62), ("Kamal", 78)]
```

Write a Python function named `sort_players(player_data)` that uses the Bubble Sort logic to sort this list in descending order (highest score first) based on the scores. If two players have the same score, their original relative order should remain unchanged (Stable Sort). Do not use the built-in `.sort()` or `sorted()` functions.

ලකුණු මත පදනම්ව මෙම ලැයිස්තුව අවරෝහණ පිළිවෙලට (වැඩිම ලකුණු පළමුව එන සේ) සකස් කිරීම සඳහා බබල් සෝට් (Bubble Sort) තර්කය භාවිතා කරමින් `sort_players(player_data)` නමින් පයිතන් ශ්‍රිතයක් ලියන්න. ක්‍රීඩකයින් දෙදෙනෙකුට සමාන ලකුණු තිබේ නම්, ඔවුන්ගේ මුල් අනුපිළිවෙල වෙනස් නොවිය යුතුය (Stable Sort). පයිතන් හි ඇති `.sort()` හෝ `sorted()` ශ්‍රිත භාවිතා නොකරන්න.

```
def sort_players(player_data):
    n = len(player_data)
    # Outer loop for each pass
    for i in range(n):
        # Inner loop for comparisons
        for j in range(0, n - i - 1):
            # Compare the scores (index 1 of the tuple)
            # Use '<' for descending order
            if player_data[j][1] < player_data[j+1][1]:
                # Swap the entire tuples
                player_data[j], player_data[j+1] = player_data[j+1],
                player_data[j]

    return player_data

# Example usage
# players = [("Amal", 45), ("Nimal", 78), ("Bimal", 62), ("Kamal", 78)]
# print(sort_players(players))
```