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 528 MARKING

2026/05/22
ictfromabc
Ravindu Bandaranayake

1. Which of the following statements best describes the difference between volatile and non-volatile memory?
නශ්‍ය මතකය සහ නශ්‍ය නොවන මතකය අතර වෙනස වඩාත් හොඳින් විස්තර කරන්නේ පහත සඳහන් කුමන ප්‍රකාශයෙන්ද?

   1) Volatile memory retains data even when the power is turned off, while non-volatile memory loses all data when power is removed.
   නශ්‍ය මතකය විදුලි බලය ක්‍රියා විරහිත වූ විට පවා දත්ත රඳවා තබා ගන්නා අතර, නශ්‍ය නොවන මතකය විදුලි බලය ඉවත් කළ විට සියලු දත්ත නැති වේ.
   2) Volatile memory is slower and used for permanent storage, whereas non-volatile memory is faster and used as RAM.
   නශ්‍ය මතකය මන්දගාමී වන අතර ස්ථිර ගබඩා කිරීම සඳහා භාවිතා කරයි, නමුත් නශ්‍ය නොවන මතකය වේගවත් වන අතර RAM ලෙස භාවිතා කරයි.
   3) Both volatile and non-volatile memories lose their data once the system is shut down.
   පද්ධතිය වසා දැමීමෙන් පසු නශ්‍ය සහ නශ්‍ය නොවන මතකයන් දෙකටම ඔවුන්ගේ දත්ත අහිමි වේ.
   4) Volatile memory requires continuous power to retain data, while non-volatile memory retains data even without power.
   නශ්‍ය මතකයට දත්ත රඳවා ගැනීමට අඛණ්ඩ බලයක් අවශ්‍ය වන අතර නශ්‍ය නොවන මතකය බලය නොමැතිව වුවද දත්ත රඳවා ගනී.
   5) Non-volatile memory is used only in cache memory, while volatile memory is used in hard drives.
   නශ්‍ය නොවන මතකය භාවිතා වන්නේ වාරක මතකයේ පමණක් වන අතර නශ්‍ය මතකය දෘඪ තැටි වල භාවිතා වේ.

Answer: 4)

Explanation / පැහැදිලි කිරීම:

Volatile memory (like RAM) loses its data when the power is turned off. It is fast and mainly used for temporary storage during active processing.
නශ්‍ය මතකය (RAM වැනි) විදුලි බලය අක්‍රිය වූ විට එහි දත්ත නැති වේ. එය වේගවත් වන අතර සක්‍රිය සැකසුම් තුළ තාවකාලික ගබඩා කිරීම සඳහා ප්‍රධාන වශයෙන් භාවිතා වේ.

Non-volatile memory (like ROM, SSD, or HDD) retains data even when the device is powered down. It is used for permanent or long-term storage.
නශ්‍ය නොවන මතකය (ROM, SSD, හෝ HDD වැනි) උපාංගය ක්‍රියා විරහිත වූ විට පවා දත්ත රඳවා ගනී. එය ස්ථිර හෝ දිගු කාලීන ගබඩා කිරීම සඳහා භාවිතා වේ.

Let’s break down the incorrect options / වැරදි විකල්ප බිඳ දමමු:

Option / විකල්පය 1)
Incorrect. It reverses the roles of volatile and non-volatile memory.
වැරදියි. එය නශ්‍ය සහ නශ්‍ය නොවන මතකයේ භූමිකාවන් මාරු කර ඇත.

Option / විකල්පය 2)
Incorrect. Again, it misidentifies the speed and use; RAM (volatile) is faster than storage like SSDs or HDDs (non-volatile).
වැරදියි. නැවතත්, එය වේගය සහ භාවිතය වැරදි ලෙස හඳුනා ගනී; RAM (නශ්‍ය), SSD හෝ HDDs (නශ්‍ය නොවන) වැනි ගබඩා වලට වඩා වේගවත් වේ.

Option / විකල්පය 3)
Incorrect. Only volatile memory loses data on shutdown, not non-volatile.
වැරදියි. නශ්‍ය මතකය පමණක් ක්‍රියා විරහිත වූ විට දත්ත අහිමි වේ, නශ්‍ය නොවන නොවේ.

Option / විකල්පය 5)
Incorrect. Cache memory is a type of volatile memory, and hard drives are non-volatile.
වැරදියි. වාරක මතකය නශ්‍ය මතක වර්ගයක් වන අතර දෘඪ තැටි නශ්‍ය නොවන වේ.

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

2. 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 පාලකය 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 එනම්, 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 (0ජ්9) 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

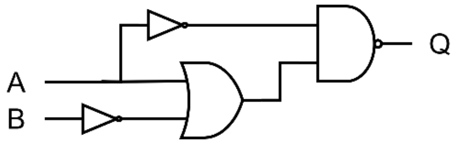
Correct BCD sequence / නිවැරදි BCD අනුපිළිවෙල: 0000 1001 0100 0111

The final answer is Option 4
අවසාන පිළිතුර වන්නේ විකල්පය 4

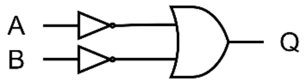
3. $Q = \overline{((A + \overline{B}).\overline{A})}$

Which of the following circuits implement this expression?
පහත සඳහන් පරිපථවලින් මෙම ප්‍රකාශනය ක්‍රියාත්මක කරන්නේ කුමක්ද?

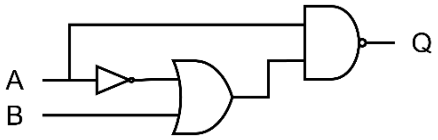
A.



B.



C.



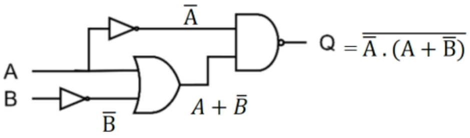
1) A only. / A පමණි.
2) B only. / B පමණි.
3) A and B only. / A හා B පමණි.
4) A and C only. / A හා C පමණි.
5) All of the above. / ඉහත සියල්ලම.

Answer: 1)

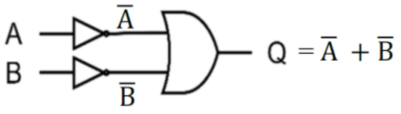
Explanation / පැහැදිලි කිරීම:

Let's get the output of each circuit given / ලබා දී ඇති එක් එක් පරිපථයේ ප්‍රතිදානය ලබා ගනිමු,

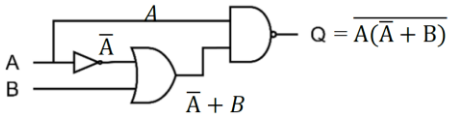
Circuit A



Circuit B



Circuit C



As the circuit A clearly outputs the expression $Q = \overline{((A + \overline{B}).\overline{A})}$, Option 1) is the answer.
A පරිපථය $Q = \overline{((A + \overline{B}).\overline{A})}$ ප්‍රකාශනය පැහැදිලිව ප්‍රතිදානය කරන බැවින්, විකල්පය 1) පිළිතුර වේ.

4. A system uses a 32-bit virtual address and has a page size of 4 KB. Each page table entry (PTE) takes 4 bytes. What is the size of the page table for a single process?
පද්ධතියක් 32-bit අතථ්‍ය ලිපිනයක් භාවිතා කරන අතර පිටු ප්‍රමාණය 4 KB වේ. සෑම පිටු වගු ඇතුළත් කිරීමකටම (PTE) බයිට් 4 ක් අවශ්‍ය වේ. තනි ක්‍රියාවලියක් සඳහා පිටු වගුවේ ප්‍රමාණය කොපමණද?

1) 1 MB 2) 8 MB 3) 2 MB 4) <u>4 MB</u> 5) 16 MB

Answer: 4)

Explanation / පැහැදිලි කිරීම:

A 32-bit virtual address means the process can access 2^{32} bytes (or 4 GB) of virtual memory. With a page size of 4 KB, which is 2^{12} bytes, the total number of virtual pages will be,
32-bit අථත්‍ය ලිපිනයක් යනු ක්‍රියාවලියට බයිට් 2^{32} හෝ 4 GB අථත්‍ය මතකයට ප්‍රවේශ විය හැකි බවයි. පිටු ප්‍රමාණය 4 KB එනම් බයිට් 2^{12} ක් සමඟ, මුළු අථත්‍ය පිටු ගණන,

Number of virtual pages / අතථ්‍ය පිටු ගණන = $2^{32} / 2^{12} = 2^{20}$

Each of these pages needs a page table entry (PTE) to keep track of where it is in physical memory. If each PTE takes 4 bytes, then the total size of the page table is,
මෙම සෑම පිටුවකටම භෞතික මතකයේ ඇති ස්ථානය නිරීක්ෂණය කිරීමට පිටු වගු ඇතුළත් කිරීමක් (PTE) අවශ්‍ය වේ. එක් එක් PTE එකකට බයිට් 4 ක් ගතවේ නම්, පිටු වගුවේ මුළු ප්‍රමාණය,

Total size of the page table / පිටු වගුවේ මුළු ප්‍රමාණය = $2^{20} \times 4 = 2^{22}$ bytes = 4 MB

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

5. Which of the following is not a SQL constraint?
පහත ඒවායින් SQL සම්බාධකයක් නොවන්නේ කුමක් ද?

1) `UNIQUE` 2) `FOREIGN KEY` 3) <u>`DELETE`</u> 4) `PRIMARY KEY` 5) `NOT NULL`

Answer: 3)

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

1) `UNIQUE`
The `UNIQUE` constraint ensures that all values in a column are different. It prevents duplicate values in the column, maintaining data integrity.
`UNIQUE` සීමාව තීරුවක ඇති සියලුම අගයන් වෙනස් බව සහතික කරයි. එය දත්ත අඛණ්ඩතාව පවත්වා ගනිමින් තීරුවේ අනුපිටපත් අගයන් වළක්වයි

2) `FOREIGN KEY`
The `FOREIGN KEY` constraint is used to link two tables together. It ensures that the value in a column (or group of columns) matches the value in another table's primary key column, maintaining referential integrity between the tables.
`FOREIGN KEY` බාධකය වගු දෙකක් එකට සම්බන්ධ කිරීමට භාවිතා කරයි. (එය තීරුවක හෝ තීරු සමූහයක) අගය වෙනත් වගුවක ප්‍රාථමික යතුරු තීරුවේ අගයට ගැළපෙන බව සහතික කරයි, වගු අතර යොමු අඛණ්ඩතාව පවත්වා ගනී.

3) `DELETE`
`DELETE` is not a constraint; it is a SQL command used to remove rows from a table. It specifies which records should be deleted from the database, but it does not impose any rules or constraints on the data.
`DELETE` යනු බාධාවක් නොවේත එය වගුවකින් පේළි ඉවත් කිරීමට භාවිතා කරන SQL විධානයකි. එය දත්ත සමුදායෙන් මකා දැමිය යුතු වාර්තා සඳහන් කරයි, නමුත් එය දත්ත මත කිසිදු රීතියක් හෝ සීමාවක් පනවා නැත.

4) `PRIMARY KEY`

The `PRIMARY KEY` constraint uniquely identifies each record in a table. A table can have only one primary key, which may consist of single or multiple columns. It ensures that no duplicate values exist for the primary key column(s) and that none of the values are null.

`PRIMARY KEY` සීමාව වගුවක ඇති සෑම වාර්තාවක්ම අනන්‍යව හඳුනා ගනී. වගුවකට තිබිය හැක්කේ එක් මූලික යතුරක් පමණක් වන අතර, එය තනි හෝ බහු තීරු වලින් සමන්විත විය හැක. එය ප්‍රාථමික යතුරු තීරු සඳහා අනුපිටපත් අගයක් නොපවතින බවත්, අගයන් කිසිවක් ශූන්‍ය නොවන බවත් සහතික කරයි.

5) `INSERT`

`INSERT` is not a constraint; it is a SQL command used to add new rows to a table. It specifies the table and the values to be inserted, but it does not impose any rules or constraints on the data.

`INSERT` යනු බාධාවක් නොවේත එය වගුවකට නව පේළි එකතු කිරීමට භාවිතා කරන SQL විධානයකි. එය වගුව සහ ඇතුළත් කළ යුතු අගයන් සඳහන් කරයි, නමුත් එය දත්ත මත කිසිදු රීතියක් හෝ සීමාවක් පනවන්නේ නැත.

So, the correct answer is 3).

එබැවින්, නිවැරදි පිළිතුර 3) වේ.

6. How do you declare a global variable inside a function?
ශ්‍රිත යක් තුළ global විචල්‍යයක් ප්‍රකාශ කරන්නේ කෙසේද?

1) `global x = 5`
2) `global x; x = 5`
3) `x = global 5`
4) `x = 5; global x`
5) `global x; x := 5`

Answer : 2)

To declare a global variable inside a function, the correct way is to use the global keyword followed by the variable name, and then assign a value to it. In this case, `global x; x = 5` is the correct syntax. This tells Python that the variable `x` refers to the global variable, not a local one, and allows you to modify the global `x` from within the function. Without declaring it as global, any assignment to `x` inside the function would be treated as a local variable. So, this approach ensures that the global variable is used and updated as intended.

ශ්‍රිත යක් තුළ ගෝලීය විචල්‍යයක් ප්‍රකාශ කිරීමට, නිවැරදි ක්‍රමය නම් විචල්‍ය නාමයෙන් පසුව එන ගෝලීය මූල පදය භාවිතා කර එයට අගයක් පැවරීමයි. මෙම අවස්ථාවේදී, `global x; x = 5` යනු නිවැරදි කාරක රීතියයි. මෙය පයිතන්ට පවසන්නේ `x` විචල්‍යය ගෝලීය විචල්‍යයක් මිස ස්ථානීය එකක් නොවන බවත්, ශ්‍රිතය තුළ සිට ගෝලීය `x` වෙනස් කිරීමට ඔබට ඉඩ සලසයි. එය ගෝලීය ලෙස ප්‍රකාශ කිරීමකින් තොරව, ශ්‍රිතය තුළ `x` වෙත ඕනෑම පැවරුමක් ස්ථානීය විචල්‍යයක් ලෙස සලකනු ලැබේ. මේ අනුව, මෙම ප්‍රවේශය ගෝලීය විචල්‍යය භාවිතා කිරීම සහ අපේක්ෂිත පරිදි යාවත්කාලීන කිරීම සහතික කරයි.

7. Consider the following python statement
පහත පයිතන් ප්‍රකාශය සලකා බලන්න

```
DataList=[52,90,67,99]
for i in DataList:
    if i< 90:
    print(i)
    break
print('end')
```

Which of the following is the correct output?
පහත ඒවායින් නිවැරදි ප්‍රතිදානය කුමක්ද?

1) `52 end`
2) `52 90 67`
3) `52 90 67 end`
4) `52 90 67 90 end`
5) `52 end 90 end 99 end`

Answer: 1)

`DataList = [52, 90, 67, 99]`
This line creates a list called `DataList` containing the elements `[52, 90, 67, 99]`.
මෙම පේළිය මූලද්‍රව්‍ය `[52, 90, 67, 99]` අඩංගු `DataList` නමින් ලැයිස්තුවක් නිර්මාණය කරයි.

`for i in DataList`
This line starts a loop that will iterate through each element in the `DataList`. On each iteration, the variable `i` will hold the value of the current element in the list.
මෙම රේඛාව `DataList` හි එක් එක් මූලද්‍රව්‍ය හරහා පුනරාවර්තනය වන ලූපයක් ආරම්භ කරයි. එක් එක් පුනරාවර්තනයකදී, `i` විචල්‍යය ලැයිස්තුවේ වත්මන් මූලද්‍රව්‍යයේ අගය රඳවා ගනු ඇත.

`if i < 90`
Inside the loop, there's an `if` statement that checks whether the current value of `i` is less than `90`.
ලූපය ඇතුළත, `i` හි වත්මන් අගය `90` ට වඩා අඩු දැයි පරීක්ෂා කරන `if` ප්‍රකාශයක් ඇත.

`print(i)`
If the condition `i < 90` is true, the value of `i` is printed. / `i < 90` කොන්දේසිය සත්‍ය නම්, i අගය මුද්‍රණය කෙරේ.

`break`
After printing the value of `i`, the `break` statement immediately exits the loop, meaning no further elements in `DataList` will be checked or processed.
`i` අගය මුද්‍රණය කිරීමෙන් පසු, `break` ප්‍රකාශය වහාම ලූපයෙන් පිටවෙයි, එයින් අදහස් වන්නේ `DataList` හි තවත් මූලද්‍රව්‍ය පරීක්ෂා කිරීම හෝ සකසනු නොලැබේ.

`print('end')`
This line is outside the loop and will execute after the loop has finished, printing the word `'end'`.
මෙම පේළිය ලූපයෙන් පිටත වන අතර ලූපය අවසන් වූ පසු `'end'` යන වචනය මුද්‍රණය කිරීමෙන් ක්‍රියාත්මක වේ.

Execution / ක්‍රියාත්මක කිරීම:
On the first iteration, `i` is `52`. The condition `i < 90` is true (since `52` is less than `90`), so `52` is printed, and the break statement is triggered, causing the loop to exit. The loop stops after printing `52`, and no further elements (`90, 67, 99`) are checked or printed. After the loop exits, `'end'` is printed.
පළමු පුනරාවර්තනයේ දී, `i` `52` වේ. `i < 90` කොන්දේසිය සත්‍ය වේ (`52, 90` ට වඩා අඩු බැවින්), එබැවින් `52` මුද්‍රණය කර ඇති අතර, break ප්‍රකාශය ක්‍රියා විරහිත වේ, ලූපය පිටවීමට හේතු වේ. `52` මුද්‍රණය කිරීමෙන් පසු ලූපය නතර වේ, සහ තවත් මූලද්‍රව්‍ය (`90, 67, 99`) පරීක්ෂා කර හෝ මුද්‍රණය නොකෙරේ. ලූපය පිටවීමෙන් පසු, `'end'` මුද්‍රණය වේ.

So, the output will be 52 end. Therefore, the correct answer is 1).
එබැවින්, ප්‍රතිදානය 52 end. එබැවින්, නිවැරදි පිළිතුර 1) වේ.

8. What is represented by data flows in a data flow diagram?
දත්ත ගැලීම් සටහනක දත්ත ප්‍රවාහ මගින් නිරූපණය කරනු ලබන්නේ කුමක්ද?

1) Just a stream of data / දත්ත ගලනයක් පමණි.
2) Only the data going to the database / දත්ත ගබඩාවට යන දත්ත පමණි.
3) Data flows and their directions / දත්ත ගලනයක් හා ඒවායේ දිශාවන්ය.
4) Only data coming from an external entity / බාහිර භූතාර්ථයක් මගින් එන දත්ත පමණි.
5) Only the direction of data flow / දත්ත ගලන දිශාව පමණි.

Answer : 3)

The correct answer is 3) Data flows and their directions.
නිවැරදි පිළිතුර නම් 3) දත්ත ගලන හා ඒවායේ දිශාවන්ය.

In a Data Flow Diagram (DFD), data flows represent the movement of data between different components of a system, including processes, data stores, and external entities. The direction of the flow is important because it shows where the data is coming from and where it is going, helping to visualize how data moves through the system.
දත්ත ප්‍රවාහ රූප සටහනක (DFD) දත්ත ප්‍රවාහයන් මඟින් ක්‍රියාවලි, දත්ත ගබඩා සහ බාහිර ආයතන ඇතුළුව පද්ධතියක විවිධ සංරචක අතර දත්ත චලනය නියෝජනය කරයි. ප්‍රවාහයේ දිශාව වැදගත් වන්නේ එය දත්ත පැමිණෙන්නේ කොතැනින්ද සහ එය යන්නේ කොතැනටද යන්නත්, පද්ධතිය හරහා දත්ත චලනය වන ආකාරය දෘශ්‍යමාන කිරීමට උපකාරී වන නිසාත් ය.

Why Other Options Are Incorrect / වෙනත් විකල්ප වැරදි ඇයි,

Option / විකල්පය 1)
False, while data flows represent data, they also include the direction of movement, not just the data itself.
අසත්‍ය වේ, දත්ත ප්‍රවාහයන් දත්ත නියෝජනය කරන අතරම, ඒවාට දත්ත පමණක් නොව චලනය වන දිශාවද ඇතුළත් වේ.

Option / විකල්පය 2)
False, data flows represent all data movements, not just those going to a database. They can represent data moving between any part of the system, such as between processes or external entities.
අසත්‍ය වේ, දත්ත ප්‍රවාහයන් දත්ත සමුදායකට යන ඒවා පමණක් නොව සියලුම දත්ත චලනයන් නියෝජනය කරයි. ක්‍රියාවලි හෝ බාහිර භූතාර්ථ අතර වැනි පද්ධතියේ ඕනෑම කොටසක් අතර චලනය වන දත්ත ඒවාට නියෝජනය කළ හැකිය.

Option / විකල්පය 4)
False, data flows show movement between all parts of the system, including internal processes and data stores, not just from external entities.
අසත්‍ය වේ, දත්ත ප්‍රවාහයන් බාහිර භූතාර්ථවලින් පමණක් නොව අභ්‍යන්තර ක්‍රියාවලි සහ දත්ත ගබඩා ඇතුළුව පද්ධතියේ සියලුම කොටස් අතර චලනය පෙන්වයි.

Option / විකල්පය 5)
False, data flows represent both the data itself and the direction it travels in, not just the direction.
අසත්‍ය වේ, දත්ත ප්‍රවාහයන් දත්තම සහ දිශාව පමණක් නොව එය ගමන් කරන දිශාව යන දෙකම නියෝජනය කරයි.

So, the correct representation in a DFD includes both the data and its direction.
එබැවින්, DFD හි නිවැරදි නිරූපණයට දත්ත සහ එහි දිශාව යන දෙකම ඇතුළත් වේ.

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

```
<?php
$i = 0;
while ($i < 3) {
    echo $i;
    $i++;
}
?>
```

1) 012 2) 123 3) 0123 4) 1234 5) 01234

Answer: 1)

Explanation / පැහැදිලි කිරීම:

The given PHP code initializes a variable `$i` to 0 and then enters a `while loop` that continues as long as `$i` is less than 3. Inside the loop, it prints the current value of `$i` and then increments `$i` by `1`. The loop executes three times: first with `$i` as `0`, then `1`, and finally `2`. After the third iteration, `$i` becomes `3`, causing the loop condition `$i < 3` to be false, and the loop exits. Consequently, the output is `012`, showing the values of `$i` during each iteration.
ලබා දී ඇති PHP කේතයෙහි $i යනු `0` නමින් විචල්‍යයක් ආරම්භ කර, පසුව `$i` 3ට වඩා අඩු වන තාක් කල් පවතින `while` loop යක් ඇතුළු කරයි. ලූපය තුළ, එය වත්මන් අගය `$i` මුද්‍රණය කරයි සහ ඉන්පසුව `$i 1`කින් වැඩි කරයි. ලූපය තුන් වරක් ක්‍රියාත්මක වේ: පළමුව `$i 0` ලෙස, පසුව `1`, සහ අවසානයේ `2`. තුන්වන පුනරාවර්තනයෙන් පසු, `$i 3` බවට පත් වන අතර එමඟින් ලූප් තත්ත්වය `$i < 3` අසත්‍ය විය යුතු අතර, ලූපය පිටවෙයි. එහි ප්‍රතිඵලයක් ලෙස, ප්‍රතිදානය `012` වේ, එක් එක් පුනරාවර්තනය අතරතුර `$i` හි අගයන් පෙන්වයි.

So, the correct answer is 1.
එබැවින්, නිවැරදි පිළිතුර 1 වේ.

10. Which of the following attributes of the `<input>` element is used to specify the type of input control in an HTML form?
HTML පෝරමයක ආදාන පාලන වර්ගය සඳහන් කිරීමට `<input>` මූලාංගයේ පහත උපලක්ෂණ වලින් කුමන උපලක්ෂණ භාවිතා කරයිද?

1) name 2) action 3) method 4) value 5) type

Answer : 5)

The type attribute in the `<input>` element is used to define the `type` of input control to be displayed, such as `text`, `password`, `radio`, `checkbox`, etc. The `name` attribute identifies the form control, the action attribute specifies where the form data should be sent, the `method` attribute determines how the form data is sent (GET or POST), and the `value` attribute sets the default value of the input field.
`<input>` මූලාංගයේ ඇති උපලක්ෂණයේ `type` එක `text`, `password`, `radio`, `checkbox`, යනාදී ප්‍රදර්ශනය කළ යුතු ආදාන පාලන වර්ගය නිර්වචනය කිරීමට භාවිතා කරයි. `name` උපලක්ෂණය මඟින් ආකෘති පාලනය හඳුනා ගනී, `action` උපලක්ෂණය සඳහන් කරන්නේ පෝරම දත්ත යැවිය යුතු කොතැනද යන්නයි, `method` උපලක්ෂණය මගින් පෝරම දත්ත යවන ආකාරය තීරණය කරයි (GET හෝ POST), සහ උපලක්ෂණයේ `value` එක ආදාන ක්ෂේත්‍රයේ පෙරනිමි අගය සකසයි.

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