සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved ]

Ravindu Bandaranayake - ictfromabc

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

# 2026 QUIZ NO 420 MARKING

2026/02/03
ictfromabc
Ravindu Bandaranayake

1. Which of the following monitor has the best energy efficiency and color accuracy?
පහත දැක්වෙන මොනිටර අතුරින් හොඳම බලශක්ති කාර්යක්ෂමතාව සහ වර්ණ නිරවද්‍යතාවය ඇත්තේ කුමකටද?

1) CRT monitor 2) TFT monitor 3) LED monitor
4) Plasma monitor 5) LCD monitor

**Answer: 3)**

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

1) **CRT Monitor**
Consumes high power and outdated technology.
අධික බලය සහ යල් පැන ගිය තාක්ෂණය භාවිතා කරයි.

2) **TFT Monitor**
A type of LCD that is energy-efficient but not as much as LED monitors.
බලශක්ති කාර්යක්ෂම නමුත් LED තිර තරම් නොවන LCD වර්ගයකි.

3) **LED Monitor**
Combines the energy efficiency of LCDs with enhanced color accuracy and brightness using LED backlighting.
LED පසුබිම් ආලෝකය භාවිතයෙන් වැඩි දියුණු කළ වර්ණ නිරවද්‍යතාවය සහ දීප්තිය සමඟ LCD වල බලශක්ති කාර්යක්ෂමතාව ඒකාබද්ධ කරයි.

4) **Plasma Monitor**
Offers good color accuracy but consumes more energy than LEDs.
හොඳ වර්ණ නිරවද්‍යතාවයක් ලබා දෙන නමුත් LED වලට වඩා වැඩි ශක්තියක් පරිභෝජනය කරයි.

5) **LCD Monitor**
Energy-efficient but generally lacks the color accuracy and contrast of LED monitors.
බලශක්ති කාර්යක්ෂම නමුත් සාමාන්‍යයෙන් LED තිරවල වර්ණ නිරවද්‍යතාවය සහ වෙනස නොමැත.

Therefore, the correct answer is option 3), LED monitor.
එබැවින්, නිවැරදි පිළිතුර විකල්ප 3), LED Monitor වේ.

2. Which of the following represents the bitwise XNOR operation of three binary numbers $10101100_2$, $11101110_2$ and $10111010_2$?
පහත කවරක් ද්වීමය $10101100_2$, $11101110_2$ හා $10111010_2$ යන සංඛ්‍යා තුනෙහි බිටු අනුසාරීත XNOR (bitwise XNOR) මෙහෙයුම නිරූපණය කරයිද?

   1) $10101010_2$ 2) $\underline{00000111_2}$ 3) $10101000_2$ 4) $11111000_2$ 5) $01010000_2$

   **Answer : 2 )**

   First this binary number needs to be converted into 3-bit wise XOR.
   පලමුව මෙම ද්වීමය සංඛ්‍යා 3න බිටු අනුසාරිත XOR බවට පරිවර්ථනය කර ගත යුතුය.

   $1\ 0\ 1\ 0\ 1\ 1\ 0\ 0_2$
   $1\ 1\ 1\ 0\ 1\ 1\ 1\ 0_2$
   $\underline{1\ 0\ 1\ 1\ 1\ 0\ 1\ 0_2}$
   $1\ 1\ 1\ 1\ 1\ 0\ 0\ 0_2$

   Since bitwise XNOR is the opposite output of bitwise XOR, 1's must be flipped into 0's and 0's into 1's.
   බිටු අනුසාරිත XNOR යනු බිටු අනුසාරිත XOR හි ප්‍රතිවිරුද්ධ ප්‍රතිදානය බැවින් 1ක 0 බවටත්, 0, 1 බවටත් පෙරලිය යුතුය.

   $1\ 1\ 1\ 1\ 1\ 0\ 0\ 0_2 \rightarrow 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1_2$

   So the correct answer is, 2) $00000111_2$
   එබැවින් නිවැරදි පිළිතුර වනුයේ, 2) $00000111_2$

3. How many inputs does a full adder require to compute its SUM and CARRY outputs?
පූර්ණ ආකලයක එහි SUM සහ CARRY ප්‍රතිදානයන් ගණනය කිරීමට කොපමණ ආදාන අවශ්‍ය වේද?

   1) 2 2) <u>3</u> 3) 4 4) 5 5) 6

   **Answer : 2)**

   **Explanation / පැහැදිලි කිරීම**

   A full adder is a digital circuit that computes the SUM and CARRY outputs using three inputs: A, B and Cin (carry-in). The inputs A and B represent the two binary digits to be added, while Cin represents the carry bit from a previous addition. These three inputs are essential for the full adder to handle binary addition involving carry propagation, which is a key requirement in multi-bit addition operations.
   පූර්ණ ආකලකය යනු ආදාන තුනක් භාවිතා කරමින් SUM සහ CARRY ප්‍රතිදාන ගණනය කරන ඩිජිටල් පරිපථයකි: A, B සහ Cin (රැගෙන යාම). A සහ B ආදාන එකතු කළ යුතු ද්විමය ඉලක්කම් දෙක නියෝජනය කරන අතර Cin පෙර එකතු කිරීමකින් රැගෙන යාමේ බිට් එක නියෝජනය කරයි. බහු-බිට් එකතු කිරීමේ මෙහෙයුම්වල ප්‍රධාන අවශ්‍යතාවයක් වන රැගෙන යාම ප්‍රචාරණය ඇතුළත් ද්විමය එකතු කිරීම හැසිරවීමට පූර්ණ ආකලකයට මෙම ආදාන තුන අත්‍යවශ්‍ය වේ.

   The full adder generates two outputs: SUM, which is the result of adding the three input bits, and CARRY, which indicates if a carry-out is generated when the sum exceeds the binary range (1). Without these three inputs, the full adder cannot produce accurate results for binary addition with carry. Hence, the full adder requires exactly three inputs, making Option 2: 3 inputs the correct answer.
   පූර්ණ ආකලකය ප්‍රතිදාන දෙකක් ජනනය කරයි: ආදාන බිටු තුන එකතු කිරීමේ ප්‍රතිඵලයක් වන SUM සහ එකතුව ද්විමය පරාසය 1 ඉක්මවන විට රැගෙන යාමක් ජනනය වන්නේද යන්න පෙන්නුම් කරන CARRY. මෙම ආදාන තුන නොමැතිව, පූර්ණ ආකලකයට රැගෙන යාම සමඟ ද්විමය එකතු කිරීම සඳහා නිවැරදි ප්‍රතිඵල නිපදවිය නොහැක. එබැවින්, පූර්ණ ආකලකයට හරියටම ආදාන තුනක් අවශ්‍ය වන අතර, විකල්ප 2) 3 ආදාන නිවැරදි පිළිතුර බවට පත් කරයි.

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

4. Which of the following is a good example of real-time processing?
තත්‍ය කාලීන සැකසුම් සඳහා පහත සඳහන් දේවලින් හොඳ උදාහරණයක් වන්නේ කුමක්ද?

1) Library book check-out system / පුස්තකාල පොත් check-out පද්ධතිය.
2) Automated stock market trading system / ස්වයංක්‍රීය කොටස් වෙළඳපොල වෙළඳ පද්ධතිය.
3) Daily website traffic analytics / දෛනික වෙබ් අඩවි තදබද විශ්ලේෂණ.
4) Monthly sales report generation / මාසික විකුණුම් වාර්තා උත්පාදනය.
5) Yearly tax report preparation / වාර්ෂික බදු වාර්තාව සකස් කිරීම.

**Answer : 2 )**

Real-time processing refers to systems that respond immediately to input and produce instant results.
තත්‍ය කාලීන සැකසුම් යනු ආදානයට ක්ෂණිකව ප්‍රතිචාර දක්වන සහ ක්ෂණික ප්‍රතිඵල ලබා දෙන පද්ධති වේ.

Let's review each option.
අපි එක් එක් විකල්පය සමාලෝචනය කරමු.

- **Option / විකල්පය 1)**
  This system processes data instantly when a book is checked out. However, it doesn't require the strict timing that real-time processing demands.
  මෙම පද්ධතිය පොතක් පරීක්ෂා කරන විට ක්ෂණිකව දත්ත සකසයි. කෙසේ වෙතත්, එයට තත්‍ය කාලීන සැකසුම් ඉල්ලා සිටින දැඩි වේලාව අවශ්‍ය නොවේ.

- **Option / විකල්පය 2)**
  This system must respond instantly to market changes, as timing is critical. It is an example of real-time processing.
  කාලය තීරණාත්මක වන බැවින් මෙම පද්ධතිය වෙළඳපල වෙනස්කම් වලට ක්ෂණිකව ප්‍රතිචාර දැක්විය යුතුය. එය තත්‍ය කාලීන සැකසුම් සඳහා උදාහරණයකි.

- **Option / විකල්පය 3)**
  This system collects data over the day and then processes it, which isn't immediate. It's not real-time processing.
  මෙම පද්ධතිය දවස පුරා දත්ත රැස් කර පසුව එය සකසයි, එය ක්ෂණික නොවේ. එය තත්‍ය කාලීන සැකසුම් නොවේ.

- **Option / විකල්පය 4)**
  This process happens over time, not instantly. It's not a real-time system.
  මෙම ක්‍රියාවලිය ක්ෂණිකව නොව කාලයත් සමග සිදු වේ. එය තත්‍ය කාලීන පද්ධතියක් නොවේ.

- **Option / විකල්පය 5)**
  This process occurs once a year and doesn't involve instant processing.
  මෙම ක්‍රියාවලිය වසරකට වරක් සිදු වන අතර ක්ෂණික සැකසුම් ඇතුළත් නොවේ.

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

5. Which of the following SQL statements will change all occurrences of 'Colombo' to 'Kandy' in the 'City' attribute of the CUSTOMER table?
CUSTOMER වගුවේ 'City' උපලක්ෂණයේ 'Colombo' හි සියලුම අවස්ථා 'Kandy' ලෙස වෙනස් කරන්නේ පහත සඳහන් SQL ප්‍රකාශනවලින් කුමන එකද?

1) `UPDATE CUSTOMER SET City = 'Kandy' WHERE City = 'Colombo';`
2) `MODIFY CUSTOMER SET City = 'Kandy' WHERE City = 'Colombo';`
3) `UPDATE CUSTOMER SET City = 'Colombo' INTO City = 'Kandy';`
4) `UPDATE CUSTOMER CHANGE City = 'Colombo' TO 'Kandy';`
5) `UPDATE CUSTOMER SET City = 'Kandy' WHERE City != 'Colombo';`

**Answer : 1 )**

**Explanation**
**පැහැදිලි කිරීම**

This statement correctly uses the SQL `UPDATE` command to modify existing records in a table. The `SET` clause specifies the new value to be assigned to the City attribute, while the `WHERE` clause ensures that only records with City equal to 'Colombo' are updated. This prevents unintended changes to other records. The syntax follows standard SQL rules, making Option 1 the correct answer.
වගුවක පවතින වාර්තා වෙනස් කිරීම සඳහා මෙම ප්‍රකාශය SQL `UPDATE` විධානය නිවැරදිව භාවිතා කරයි. `SET` වගන්තිය මඟින් City උපලක්ෂණයට පැවරිය යුතු නව අගය නියම කරන අතර `WHERE` වගන්තිය මඟින් City 'Colombo' ට සමාන වන වාර්තා පමණක් යාවත්කාලීන වන බව සහතික කරයි. මෙය අනෙකුත් වාර්තා වලට අනපේක්ෂිත වෙනස්කම් වළක්වයි. වාක්‍ය ඛණ්ඩය සම්මත SQL නීති අනුගමනය කරන අතර, විකල්ප 1 නිවැරදි පිළිතුර බවට පත් කරයි.

The final answer is
අවසාන පිළිතුර වන්නේ

Option 1
විකල්පය 1

6. What is the output of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
def outer():
    def inner():
        return 10
    return inner

func = outer()
print(func())
```

1) `10` 2) `20` 3) `100` 4) `syntax error` 5) no output

**Answer : 1 )**

Let's consider the process of this python code step by step.
මෙම පයිතන් කේතයේ ක්‍රියාවලිය පියවරෙන් පියවර සලකා බලමු.

```
def outer():
```
Here, we define a function called `outer`. This function contains another function inside it, which is called `inner`.
මෙහිදී, අපි `outer` නම් ශ්‍රිතයක් නිර්වචනය කරමු. මෙම ශ්‍රිතය තුළ තවත් ශ්‍රිතයක් අඩංගු වන අතර එය `inner` ලෙස හැඳින්වේ.

```
def inner():
      return 10
```

Inside `outer`, there's another function defined called `inner`. The `inner` function simply returns the value 10 when called.
`outer` තුළ, `inner` ලෙස අර්ථ දක්වා ඇති තවත් ශ්‍රිතයක් ඇත. `inner` ශ්‍රිතය ඇමතූ විට 10 අගය ලබා දෙයි.

```
return inner
```

The `outer` function returns the `inner` function itself, not its result. So when `outer()` is called, it returns the `inner` function.
`outer` ශ්‍රිතය `inner` ශ්‍රිතයම ලබා දෙයි, එහි ප්‍රතිඵලය නොවේ. එබැවින් `outer()` ලෙස හැඳින්වූ විට, එය `inner` ශ්‍රිතය ආපසු ලබා දෙයි.

```
func = outer()
```

When `outer()` is called, it returns the `inner` function, and this function is assigned to the variable `func`. This means `func` now holds a reference to the `inner` function.
`outer()` කැඳවූ විට, එය `inner` ශ්‍රිතය ආපසු ලබා දෙන අතර, මෙම ශ්‍රිතය `func` විචල්‍ය වෙත පවරා ඇත. මෙයින් අදහස් කරන්නේ `func` දැන් `inner` ශ්‍රිතයට සඳහනක් දරයි.

```
print(func())
```

When `func()` is called, it's equivalent to calling `inner()`. Since `inner` returns 10, `func()` will return 10, which is then printed.
`func()` කැඳවූ විට, එය `inner()` ඇමතීමට සමාන වේ. `inner` මගින් 10 ලබාදෙන බැවින්, `func()` විසින් 10 ලබා දෙනු ඇත, එය ප්‍රතිදානය කෙරේ.

The output will be `10`. So, the correct answer number is 1.
ප්‍රතිදානය `10` වනු ඇත. එමනිසා, නිවැරදි පිළිතුරු අංකය 1 වේ.

7. What is the output of the following python program if the inputs were -2, -5 and 9?
ආදාන -2, -5 සහ 9 නම් පහත පයිතන් වැඩසටහනේ ප්‍රතිදානය කුමක්ද?

```
i=int(input("enter the num i:"))
j=int(input("enter the num j:"))
k=int(input("enter the num k:"))
if i<j:
     if j<k:
          i=j
     else:
          j=k
else:
     if j>k:
           j=i
     else:
           i=k
print(i,j,k)
```

1) <u>`9, -5 , 9`</u>  2) `-5, 9 , 9`  3) `9, -5 ,0`  4) `5, -7, 9`  5) `-8, 7 ,9`

**Answer : 1 )**

Let's analyze the Python code step-by-step with the given inputs.
දී ඇති ආදාන සමඟ පියවරෙන් පියවර පයිතන් කේතය විශ්ලේෂණය කරමු.

- First `if` Condition / පළමු `if` කොන්දේසිය
  `if i < j : → if -2 < -5`: is False.

Since this condition is false, we move to the `else` block.
මෙම කොන්දේසිය අසත්‍ය බැවින්, අපි `else` කොටස වෙත යමු.

- `else` Block
  `if j > k:` → `if -5 > 9` : is False.

Since this condition is false, we move to the `else` part inside the `else` block.
මෙම කොන්දේසිය අසත්‍ය බැවින්, අපි `else` බ්ලොක් එක ඇතුලේ `else` කොටසට යමු.

- Inner `else` Block
  `i = k → i = 9`

At this point, the values are updated as follows.
මෙම අවස්ථාවේදී, අගයන් පහත පරිදි යාවත්කාලීන වේ

```
i = 9
j = -5
k = 9
```

- Final Output
  `print(i, j, k)` will output: `9 -5 9`

Therefore the correct answer is **1)** `9 -5 9`
එබැවින් නිවැරදි පිළිතුර **1)** `9 -5 9`

8. Which of the following statements about file handling modes in Python is correct?
පයිතන්හි ගොනු හැසිරවීමේ ප්‍රකාරය පිළිබඳ පහත ප්‍රකාශ අතුරින් නිවැරදි වන්නේ කුමක්ද?

1) The mode `'r'` is used to open a file for both reading and writing.
කියවීම සහ ලිවීම යන දෙකටම ගොනුවක් විවෘත කිරීමට `'r'` ප්‍රකාරය භාවිතා කරයි.
2) The mode `'a'` is used to read from a file, and it raises an error if the file does not exist.
ගොනුවක් කියවීමට `'a'` ප්‍රකාරය භාවිතා කරන අතර, ගොනුව නොමැති නම් එය දෝෂයක් මතු කරයි.
3) <u>The mode `'w'` will create a new file if it does'nt exist, and if the file exists, it will overwrite its</u> content.
`'w'` ප්‍රකාරය ගොනුව නොමැති නම් නව ගොනුවක් සාදනු ඇත, සහ ගොනුව පවතී නම්, එය එහි අන්තර්ගතය උඩින් නැවත ලියයි.
4) The mode `'r+'` is used to open a file for writing only, and it creates a new file if it doesn't exist.
ලිවීම සඳහා පමණක් ගොනුවක් විවෘත කිරීමට `'r+'` ප්‍රකාරය භාවිතා කරන අතර එය නොමැති නම් එය නව ගොනුවක් සාදයි.
5) The mode `'x'` will append data to the end of the file if it already exists.
`'x'` ප්‍රකාරය ගොනුව දැනටමත් පවතී නම් ගොනුවේ අවසානයට දත්ත එකතු කරයි.

**Answer : 3 )**

Let's consider each file handling modes with each of their function.
එක් එක් ගොනු හැසිරවීමේ ප්‍රකාරයන් ඒවායේ එක් එක් කාර්යය සමග සලකා බලමු.

- `'r'`
  Opens a file for reading. Raises an error if the file doesn't exist.
  කියවීම සඳහා ගොනුවක් විවෘත කරයි. ගොනුව නොමැති නම් දෝෂයක් මතු කරයි.

- `'w'`
  Opens a file for writing. Creates a new file if it doesn't exist or overwrites if it does.
  ලිවීම සඳහා ගොනුවක් විවෘත කරයි. එය නොපවතියි නම් නව ගොනුවක් සාදන්න හෝ එය තිබේ නම් උඩින් (overwrite) ලියයි.

- **'a'**
  Opens a file for appending. Creates a new file if it doesn't exist.
  එකතු කිරීම සඳහා ගොනුවක් විවෘත කරයි. එය නොමැති නම් නව ගොනුවක් සාදයි.

- `'r+'`
  Opens a file for both reading and writing. Raises an error if the file doesn't exist.
  කියවීම සහ ලිවීම යන දෙකටම ගොනුවක් විවෘත කරයි. ගොනුව නොමැති නම් දෝෂයක් මතු කරයි.

- `'x'`
  Creates a new file. Raises an error if the file already exists.
  නව ගොනුවක් සාදයි. ගොනුව දැනටමත් තිබේ නම් දෝෂයක් මතු කරයි.

- **Option / විකල්ප 1)**
  Incorrect statement. The mode **'r'** is used only for reading, not both reading and writing.
  වැරදි ප්‍රකාශයකි. **'r'** ප්‍රකාරය කියවීම සඳහා පමණක් භාවිතා වේ, කියවීම සහ ලිවීම යන දෙකම නොවේ.

- **Option / විකල්ප 2)**
  Incorrect statement. The mode **'a'** is for appending to a file, not reading. It doesn't raise an error if the file doesn't exist but creates a new file.
  වැරදි ප්‍රකාශයකි. **'a'** ප්‍රකාරය ගොනුවකට එකතු කිරීම සඳහා මිස කියවීම නොවේ. ගොනුව නොමැති නමුත් නව ගොනුවක් නිර්මාණය කරන්නේ නම් එය දෝෂයක් මතු නොකරයි.

- **Option / විකල්ප 3)**
  Correct statement. The mode **'w'** will create a new file if it doesn't exist. If the file exists, it will overwrite its content.
  නිවැරදි ප්‍රකාශයකි. **'w'** ප්‍රකාරය එය නොමැති නම් නව ගොනුවක් සාදනු ඇත. ගොනුව පවතී නම්, එය එහි අන්තර්ගතය උඩින් ලියයි.

- **Option / විකල්ප 4)**
  Incorrect statement. The mode **'r+'** is for reading and writing, not writing only. It raises an error if the file doesn't exist.
  වැරදි ප්‍රකාශයකි. **'r+'** ප්‍රකාරය ලිවීමට පමණක් නොව කියවීමට සහ ලිවීම සඳහා වේ. ගොනුව නොමැති නම් එය දෝෂයක් මතු කරයි.

- **Option / විකල්ප 5)**
  Incorrect statement. The mode **'x'** creates a new file and raises an error if the file exists, but it doesn't append data to an existing file.
  වැරදි ප්‍රකාශයකි. **'x'** ප්‍රකාරය නව ගොනුවක් සාදන අතර ගොනුව තිබේ නම් දෝෂයක් ඇති කරයි, නමුත් එය පවතින ගොනුවකට දත්ත එකතු නොකරයි.

**Correct Answer is 3) The mode 'w' will create a new file if it does not exist, and if the file exists, it will overwrite its content.**
නිවැරදි පිළිතුර 3) `'w'` ප්‍රකාරය ගොනුව නොමැති නම් නව ගොනුවක් සාදනු ඇත, සහ ගොනුව පවතී නම්, එය එහි අන්තර්ගතය උඩින් නැවත ලියයි.

9. You have an HTML form, but the submit button is not working. Which of the following could be the reason?
ඔබට HTML ෆෝරමයක් ඇත, නමුත් submit කිරීමේ බොත්තම ක්‍රියා නොකරයි. පහත සඳහන් ඒවායින් කුමන හේතුවක් විය හැකිද?

1) The <form> tag is missing the action attribute.
<form> උසුලනයට action ගුණාංගය මග හැරී ඇත.
2) The <input> tag for the submit button is missing the type="submit" attribute.
submit කිරීමේ බොත්තම සඳහා <input> උසුලනයට type="submit" ගුණාංගය මග හැරී ඇත.
3) The <form> tag is nested inside a <div> / <form> උසුලනය <div> තුළ යොදා ඇත.
4) The <button> tag is used instead of <input> / <input> වෙනුවට <button> උසුලනය භාවිතා වේ.
5) The form is styled with CSS / ෆෝරමය CSS සමඟ හැඩගස්වා ඇත.

**Answer: 2)**

**Explanation / පැහැදිලි කිරීම:**

For an HTML form to be submitted, there must be a submit button. The most common way to do this is:
HTML ෆෝරමයක් ඉදිරිපත් කිරීමට නම්, submit බොත්තමක් තිබිය යුතුය. මෙය කිරීමට වඩාත් පොදු ක්‍රමය වන්නේ:

```
<input type="submit" value="Submit">
```

If the `type="submit"` attribute is missing, the input field will not trigger form submission.
`type="submit"` උපලක්ෂණ අස්ථානගත වී ඇත්නම්, ආදාන ක්ෂේත්‍රය ෆෝරමය submit කිරීම trigger වන්නේ නැත.

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

Why the other options are incorrect / වෙනත් විකල්ප වැරදි ඇයි?

1) **The `<form>` tag is missing the `action` attribute.**
The `action` attribute defines where the form data is sent, but even if it's missing, the form will still submit (to the current page by default).
`action` උපලක්ෂණයෙන් ෆෝරම දත්ත යවන ස්ථානය නිර්වචනය කරයි, නමුත් එය අතුරුදහන් වුවද, ෆෝරමය තවමත් ඉදිරිපත් කරනු ඇත (default පිටුවට).

3) **The `<form>` tag is nested inside a `<div>`.**
Wrapping a form inside a `<div>` does not prevent it from working.
`<div>` එකක් ඇතුලේ Wrap කිරීමෙන් එය ක්‍රියා කිරීම වලක්වන්නේ නැත.

4) **The `<button>` tag is used instead of `<input>`.**
The `<button>` element automatically acts as a submit button if it's inside a `<form>`.
`<button>` element එක `<form>` තුළ තිබේ නම් එය submit කිරීමේ බොත්තමක් ලෙස ස්වයංක්‍රියව ක්‍රියා කරයි.

Example / උදාහරණය: `<button type="submit">Submit</button>`

If `type="submit"` is missing, it may still submit by default.
`type="submit"` missing නම්, එය තවමත් default ලෙෂ submit කළ හැක.

5) **The form is styled with CSS.**
CSS only affects appearance; it does not prevent form submission.
CSS පෙනුමට පමණක් බලපායි; එය ෆෝරමය ඉදිරිපත් කිරීම වළක්වන්නේ නැත.

10. Which of the following tags is used to define a table in an HTML document?
HTML ලේඛනයක වගුවක් අර්ථ දැක්වීමට පහත සඳහන් උසුලන වලින් කුමන උසුලනය භාවිතා කරයිද?

1) <table></table>
2) <div></div>
3) <section></section>
4) <list></list>
5) <grid></grid>

**Answer : 1)**

**Explanation / පැහැදිලි කිරීම**

The `<table>` tag is used in an HTML document to define a table. It serves as the container for the entire table structure, including rows and columns. Inside the `<table>` tag, additional tags like `<tr>` (table row), `<td>` (table data), and `<th>` (table header) are used to specify the structure and content of the table.
HTML ලේඛනයක වගුවක් නිර්වචනය කිරීම සඳහා <table> උසුලනය භාවිතා කරයි. එය පේළි සහ තීරු ඇතුළුව සම්පූර්ණ වගු ව්‍යුහය සඳහා බහාලුම ලෙස ක්‍රියා කරයි. <table> උසුලනය තුළ, (<tr> වගු පේළිය), <td>( වගු දත්ත) සහ <th>
(වගු ශීර්ෂය) වැනි අමතර උසුලන වගුවේ ව්‍යුහය සහ අන්තර්ගතය නියම කිරීමට භාවිතා කරයි. උදාහරණයක් ලෙස:

For example:

```
<table border="1px solid black">
  <tr>
    <th>Header 1</th>
    <th>Header 2</th>
  </tr>
<td>Data 1</td>
    <td>Data 2</td>
  </tr>
</table>
```

Output :

| Header 1 | Header 2 |
|---|---|
| Data 1 | Data 2 |

This syntax ensures a well-structured representation of tabular data. Other tags such as <div> and <section> serve different purposes like layout and grouping content, while <list> and <grid> are not valid tags in HTML. Hence, <table> is the correct tag for defining tables මෙම වාක්‍ය ඛණ්ඩය වගු දත්තවල හොඳින් ව්‍යුහගත නිරූපණයක් සහතික කරයි. <div> සහ <section> වැනි අනෙකුත් උසුලනය පිරිසැලසුම සහ අන්තර්ගත කාණ්ඩ කිරීම වැනි විවිධ අරමුණු සඳහා සේවය කරන අතර, <list> සහ <grid> HTML හි වලංගු උසුලනය නොවේ. එබැවින්, <table> යනු වගු අර්ථ දැක්වීම සඳහා නිවැරදි උසුලනය වේ.