Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 273 MARKING

2025/09/09
ictfromabc
Ravindu Bandaranayake

1. A government wants to reduce the Digital Divide in rural areas. Which of the following strategies would be most effective?
ග්‍රාමීය ප්‍රදේශවල ඩිජිටල් බෙදීම අඩු කිරීමට රජයකට අවශ්‍යයි. පහත සඳහන් උපාය මාර්ගවලින් වඩාත්ම ඵලදායී වන්නේ කුමක්ද?

   1) Providing free or subsidized internet access and digital devices to underserved communities.
   අඩු පහසුකම් සහිත ප්‍රජාවන්ට නොමිලේ හෝ සහනාධාර සහිත අන්තර්ජාල ප්‍රවේශය සහ ඩිජිටල් උපාංග ලබා දීම.

   2) Encouraging people to buy the latest smartphones for social media use.
   සමාජ මාධ්‍ය භාවිතය සඳහා නවතම ස්මාර්ට්ෆෝන් මිලදී ගැනීමට ජනතාව දිරිමත් කිරීම.

   3) Limiting internet access in urban areas to balance usage.
   භාවිතය සමතුලිත කිරීම සඳහා නාගරික ප්‍රදේශවල අන්තර්ජාල ප්‍රවේශය සීමා කිරීම.

   4) Replacing all traditional schools with online-only education in cities.
   නගරවල සියලුම සාම්ප්‍රදායික පාසල් මාර්ගගත අධ්‍යාපනයෙන් ප්‍රතිස්ථාපනය කිරීම.

   5) Offering discounts on gaming consoles for teenagers.
   යෞවනයන් සඳහා ක්‍රීඩා කොන්සෝල සඳහා වට්ටම් ලබා දීම.

Answer : 1)

Digital Divide refers to the gap between those who have access to digital technologies and those who do not. ඩිජිටල් බෙදීම යනු ඩිජිටල් තාක්ෂණයන්ට ප්‍රවේශය ඇති අය සහ නොමැති අය අතර පරතරයයි.

Effective strategy: Ensure rural or underserved areas have internet connectivity and devices.
ඵලදායී උපාය මාර්ගය: ග්‍රාමීය හෝ ඌන සේවා ප්‍රදේශවල අන්තර්ජාල සම්බන්ධතාවය සහ උපාංග ඇති බව සහතික කිරීම.

Therefore the correct answer is option 1).
එබැවින් නිවැරදි පිළිතුර විකල්පය 1) වේ.

2. Which of the following best describes the primary function of the BIOS in a computer system?
පරිගණක පද්ධතියක BIOS හි ප්‍රධාන ක්‍රියාකාරිත්වය වඩාත් හොඳින් විස්තර කරන්නේ පහත සඳහන් ඒවායින් කුමක්ද?

   1) It permanently stores the operating system to ensure it loads every time the computer starts.
   පරිගණකය ආරම්භ වන සෑම අවස්ථාවකම එය පූරණය වන බව සහතික කිරීම සඳහා එය මෙහෙයුම් පද්ධතිය ස්ථිරවම ගබඩා කරයි.

   2) It initializes and tests hardware components before handing control to the operating system.
   මෙහෙයුම් පද්ධතියට පාලනය භාර දීමට පෙර එය දෘඩාංග සංරචක ආරම්භ කර පරීක්ෂා කරයි.

   3) It manages multitasking by allocating CPU time among running processes.
   ක්‍රියාත්මක වන ක්‍රියාවලීන් අතර CPU කාලය වෙන් කිරීමෙන් එය බහු කාර්යයන් කළමනාකරණය කරයි.

   4) It encrypts system data to protect it during the boot sequence.
   එය ඇරඹුම් අනුපිළිවෙල අතරතුර එය ආරක්ෂා කිරීම සඳහා පද්ධති දත්ත සංකේතනය කරයි.

   5) It provides cloud-based updates for drivers and firmware.
   එය ධාවක සහ ස්ථිරාංග සඳහා වලාකුළු මත පදනම් වූ යාවත්කාලීන කිරීම් සපයයි.

Answer : 2)

BIOS (Basic Input/Output System) is firmware stored on a computer's motherboard.
BIOS යනු පරිගණකයක මවු පුවරුවේ ගබඩා කර ඇති ස්ථිරාංග වේ.

Its primary function is to perform POST (Power-On Self-Test), which checks that essential hardware components (like RAM, CPU, and storage devices) are functioning properly.
එහි මූලික කාර්යය වන්නේ POST (Power-On Self-Test) සිදු කිරීමයි, එය අත්‍යවශ්‍ය දෘඩාංග සංරචක (RAM, CPU සහ ගබඩා උපාංග වැනි) නිසි ලෙස ක්‍රියාත්මක වන බව පරීක්ෂා කරයි.

After testing, BIOS locates and loads the bootloader from the storage device, which then starts the operating system.
පරීක්ෂා කිරීමෙන් පසු, BIOS ගබඩා උපාංගයෙන් ඇරඹුම් කාරකය සොයාගෙන පූරණය කරයි, එය මෙහෙයුම් පද්ධතිය ආරම්භ කරයි.

Option 1)
BIOS does not permanently store the OS; the OS resides on storage drives.
BIOS ස්ථිරවම OS ගබඩා නොකරයි; OS ගබඩා ධාවක මත පවතී.

Option 3)
CPU multitasking and process allocation is handled by the operating system, not BIOS.
CPU බහුකාර්ය සහ ක්‍රියාවලි වෙන් කිරීම BIOS නොව මෙහෙයුම් පද්ධතිය විසින් හසුරුවනු ලැබේ.

Option 4)
BIOS may support security features, but its main purpose is not encrypting system data.
BIOS ආරක්ෂක විශේෂාංග සඳහා සහය දැක්විය හැකි නමුත් එහි ප්‍රධාන අරමුණ පද්ධති දත්ත සංකේතනය කිරීම නොවේ.

Option 5)
Cloud-based updates are not a BIOS function; they are handled through the OS or firmware utilities.
වලාකුළු මත පදනම් වූ යාවත්කාලීන කිරීම් BIOS කාර්යයක් නොවේ; ඒවා OS හෝ ස්ථිරාංග උපයෝගිතා හරහා හසුරුවනු ලැබේ.

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

3. Which of the following best explains the necessity of coding systems in computing for data representation and processing?
දත්ත නිරූපණය සහ සැකසීම සඳහා පරිගණනයේදී කේතිකරණ පද්ධතිවල අවශ්‍යතාවය පහත සඳහන් දේවලින් වඩාත් හොඳින් පැහැදිලි කරන්නේ කුමක්ද?

   1) Coding systems are only required for representing complex mathematical equations.
   කේතිකරණ පද්ධති අවශ්‍ය වන්නේ සංකීර්ණ ගණිතමය සමීකරණ නියෝජනය කිරීම සඳහා පමණි.

   2) They limit the use of standardized data types and structures for specific applications.
   ඒවා විශේෂිත යෙදුම් සඳහා ප්‍රමිතිගත දත්ත වර්ග සහ ව්‍යුහයන් භාවිතය සීමා කරයි.

   3) Coding systems ensure consistent and efficient interpretation of data across different platforms.
   කේතිකරණ පද්ධති මගින් විවිධ වේදිකා හරහා දත්තවල ස්ථාවර සහ කාර්යක්ෂම අර්ථකථනය සහතික කරයි.

   4) They allow for the unrestricted transmission of data without encryption or compression.
   ඒවා කේතනය හෝ සම්පීඩනයකින් තොරව දත්ත සීමා රහිතව සම්ප්‍රේෂණය කිරීමට ඉඩ සලසයි.

   5) Coding systems primarily serve as a means to obscure data from unauthorized users.
   කේතිකරණ පද්ධති මූලික වශයෙන් අනවසර පරිශීලකයින්ගෙන් දත්ත සගවා ගැනීමේ මාධ්‍යයක් ලෙස සේවය කරයි.

Answer : 3)

Coding systems in computing, such as binary, ASCII, and Unicode, are essential for representing data in a standardized format that can be universally understood across different platforms and devices. They ensure that data can be consistently interpreted and processed, regardless of the system or application being used. This consistency and efficiency are crucial for seamless data exchange and integration in a diverse computing environment.
ද්විමය, ASCII සහ යුනිකේත වැනි පරිගණනයේ කේතිකරණ පද්ධති, විවිධ වේදිකා සහ උපාංග හරහා විශ්වීයව අවබෝධ කර ගත හැකි ප්‍රමිතිගත ආකෘතියකින් දත්ත නිරූපණය කිරීම සඳහා අත්‍යවශ්‍ය වේ. භාවිතා කරන පද්ධතිය හෝ යෙදුම කුමක් වුවත්, දත්ත අඛණ්ඩව අර්ථකථනය කර සැකසීමට හැකි බව ඒවා මගින් සහතික කරයි. විවිධ පරිගණක පරිසරයක් තුළ බාධාවකින් තොරව දත්ත හුවමාරු කිරීම සහ ඒකාබද්ධ කිරීම සඳහා මෙම අනුකූලතාව සහ කාර්යක්ෂමතාව ඉතා වැදගත් වේ.

Other Options: අනෙක් විකල්ප:

Statement / ප්‍රකාශය 1
Coding systems are not limited to representing complex mathematical equations. They are used for a broad range of data types and applications,
කේතිකරණ පද්ධති සංකීර්ණ ගණිතමය සමීකරණ නියෝජනය කිරීමට සීමා නොවේ. ඒවා පුළුල් පරාසයක දත්ත වර්ග සහ යෙදුම් සඳහා භාවිතා වේ,

Statement / ප්‍රකාශය 2
While coding systems do use standardized data types and structures, they do not limit their use. Instead, they facilitate interoperability and consistency across different applications, which is essential for data exchange.
කේතිකරණ පද්ධති ප්‍රමිතිගත දත්ත වර්ග සහ ව්‍යුහයන් භාවිතා කරන අතර, ඒවායේ භාවිතය සීමා නොකරයි. ඒ වෙනුවට, ඒවා දත්ත හුවමාරුව සඳහා අත්‍යවශ්‍ය වන විවිධ යෙදුම් හරහා අන්තර් ක්‍රියාකාරීත්වය සහ අනුකූලතාව සඳහා පහසුකම් සපයයි.

Statement / ප්‍රකාශය 4
Coding systems do not inherently allow for the unrestricted transmission of data without encryption or compression. These are separate concerns related to data security and efficiency, not the primary purpose of coding systems.
කේතිකරණ පද්ධති සංකේතනය හෝ සම්පීඩනයකින් තොරව දත්ත සීමා රහිතව සම්ප්‍රේෂණය කිරීමට සහජයෙන්ම ඉඩ නොදේ. මේවා දත්ත ආරක්ෂාව සහ කාර්යක්ෂමතාවයට සම්බන්ධ වෙනම සැලකිලි මිස කේතිකරණ පද්ධතිවල මූලික අරමුණ නොවේ.

Statement / ප්‍රකාශය 5
Coding systems do not inherently allow for the unrestricted transmission of data without encryption or compression. These are separate concerns related to data security and efficiency, not the primary purpose of coding systems.
කේතිකරණ පද්ධති සංකේතනය හෝ සම්පීඩනයකින් තොරව දත්ත සීමා රහිතව සම්ප්‍රේෂණය කිරීමට සහජයෙන්ම ඉඩ නොදේ. මේවා දත්ත ආරක්ෂාව සහ කාර්යක්ෂමතාවයට සම්බන්ධ වෙනම සැලකිලි මිස කේතිකරණ පද්ධතිවල මූලික අරමුණ නොවේ.

Therefore the correct answer is option 3).
එබැවින් නිවැරදි පිළිතුර විකල්පය 3) වේ.

4. "A" is represented by 1000001 in ASCII. What is the correct ASCII value for "CEDAB"
"A" හි ASCII අගය 1000001 මගින් නිරූපණය කෙරේ. "CEDAB" සඳහා නිවැරදි ASCII අගය කුමක්ද?

1) 1000011 1000101 1000100 1000001 1000010
2) 1000010 1000101 1000100 1000001 1000010
3) 1000011 1000100 1000101 1000001 1000010
4) 1000010 1000100 1000100 1000000 1000110
5) 1000010 1000100 1000101 1000001 1000011

Answer : 1)

To determine the correct ASCII value for "CEDAB", we need to convert each letter to its ASCII binary equivalent.
"CEDAB" සඳහා නිවැරදි ASCII අගය තීරණය කිරීම සඳහා, සෑම අකුරක්ම එහි ASCII ද්විමය සමාන බවට පරිවර්තනය කළ යුතුය.

ASCII values for the letters / අකුරු සඳහා ASCII අගයන්
C = 67 → 1000011
E = 69 → 1000101
D = 68 → 1000100
A = 65 → 1000001
B = 66 → 1000010

Now, let's represent "CEDAB" in its correct ASCII binary values.
දැන්, "CEDAB" එහි නිවැරදි ASCII ද්විමය අගයන් නියෝජනය කරමු.

C → 1000011
E → 1000101
D → 1000100
A → 1000001
B → 1000010

Correct sequence / නිවැරදි අනුපිළිවෙල
1000011 1000101 1000100 1000001 1000010

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

5. What is the most simplified form of $A.(B+C)+\overline{A}.(B+C)+A.\overline{B}.C$
$A.(B+C)+\overline{A}.(B+C)+A.\overline{B}.C$ හි වඩාත් සරලතම ආකාරය කුමක්ද?

1) $A.B.C$ 2) $A+B+C$ 3) $\underline{B+C}$ 4) $A.(B+C)$ 5) None of the above
ඉහත කිසිවක් නොවේ

Answer : 3)

$=A.(B+C)+\overline{A}.(B+C)+A.\overline{B}.C$
$=A.B+A.C+\overline{A}.B+\overline{A}.C+A.\overline{B}.C$ Distributive law (විෂටන න්‍යාය)
$=A.B+A.C+\overline{A}.B+\overline{A}.C$ Redundancy law (සමතිරික්ත න්‍යාය)
$=B.(A+\overline{A})+C.(A+\overline{A})$ Distributive law (විෂටන න්‍යාය)
$=B.(1)+C.(1)$ Inverse law (ප්‍රතිලෝම න්‍යාය)
$=B+C$ Identity law (සර්වසාම්‍ය න්‍යාය)

Therefore, the answer is 3).
එබැවින්, පිළිතුර 3) වේ.

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

```
i = 0
while i < 3:
    j = 0
    while j < 2:
        print(i,j,end="_")
        j += 1
    i += 1
```

1) `0 0_1 0_2 0`
2) `0 0 0 1 1 0 1 1 2 0 2 1`
3) `(0, 0) (1, 1) (2, 2)`
4) Infinite loop. / අනන්ත ලූපයකි.
5) syntax error. / දෝශයක් ලබාදෙයි.

Answer : 2)

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

- **`i = 0`**

This initializes the outer loop variable `i` to `0`.
මෙය පිටත ලූපයේ විචල්‍යය `i`, `0` ලෙස අර්ථ දක්වයි.

- **`while i < 3:`**

The outer while loop will continue as long as `i < 3`. This loop runs three times because `i` starts at `0` and increments by `1` each time until it reaches 3.
පිටත while ලූපය `i < 3` වන තෙක් දිගටම පවතිනු ඇත. මෙම ලූපය තුන් වරක් ක්‍රියාත්මක වන්නේ `i`, `0` න් ආරම්භ වන අතර එය 3 ට ළඟා වන තෙක් සෑම අවස්ථාවකම `1` කින් වැඩි වන බැවිනි.

- **`j = 0`**

Inside the outer loop, `j = 0` initializes the inner loop variable `j` to `0`.
පිටත ලූපයෙන් ඇතුළත, `j = 0` මඟින් අභ්‍යන්තර ලූපයේ විචල්‍යය `j`, `0` ලෙස අර්ථ දක්වයි.

- **`while j < 2:`**

The inner loop runs two times because `j` starts at `0` and increments by `1` each time until it reaches 2.
`j`, `0` න් ආරම්භ වන අතර එය 2 ට ළඟා වන තෙක් සෑම අවස්ථාවකම `1` කින් වැඩි වන බැවින් අභ්‍යන්තර ලූපය දෙවරක් ක්‍රියාත්මක වේ.

- **`print(i,j,end="_")`**

Inside the inner loop, `print(i, j, end="_")` prints the current values of `i` and `j` with an underscore (_) as the separator. The `end="_"` ensures that the printed output ends with an underscore instead of the default newline.
අභ්‍යන්තර ලූපය තුළ, `print(i, j, end="_")` මගින් `i` සහ `j` හි වත්මන් අගයන් වෙන්කිරීමක් ලෙස යටි ඉරි (_) සමඟ මුද්‍රණය කරයි. `end="_"` මගින් මුද්‍රිත ප්‍රතිදානය පෙරනිමි අගය වන නව රේඛාව වෙනුවට යටි ඉරකින් අවසන් වන බව සහතික කරයි.

- **`j += 1`**

After each `print` statement, `j += 1` increments `j` by `1`, which eventually ends the inner loop when `j` reaches 2.
එක් එක් `print` ප්‍රකාශයෙන් පසුව, `j += 1` මගින් `j` හි අගය `1` කින් වැඩි කරයි, `j` අවසානයේ 2ට ළඟා වූ විට අභ්‍යන්තර ලූපය අවසන් කරයි.

- `i += 1`

After the inner loop finishes, `i += 1` increments `i` by `1`, which eventually ends the outer loop when `i` reaches 3.
අභ්‍යන්තර ලූපය අවසන් වූ පසු, `i += 1` මගින් `i` හි අගය `1` කින් වැඩි කරයි, `i` අවසානයේ `3` ට ළඟා වූ විට පිටත ලූපය අවසන් කරයි.

So, the loop works as follows.
එවිට, පහත ආකාරයට ලූප ක්‍රියාකරයි.

For `i = 0`, `j` goes from `0` to `1`, producing `0 0_0 1_`.
`i = 0` සඳහා, `j`, `0` සිට `1` දක්වා ගොස්, `0 0_0 1_` නිපදවයි.

For `i = 1`, `j` goes from `0` to `1`, producing `1 0_1 1_`.
`i = 1` සඳහා, `j`, `0` සිට `1` දක්වා ගොස්, `1 0_1 1_` නිපදවයි.

For `i = 2`, `j` goes from `0` to `1`, producing `2 0_2 1_`.
`i = 2` සඳහා, `j`, `0` සිට `1` දක්වා ගොස්, `2 0_2 1_` නිපදවයි.

Then the final output will be as below.
එවිට අවසාන ප්‍රතිදානය පහත ආකාරයට වේ.

```
0 0_0 1_1 0_1 1_2 0_2 1_
```

So, the correct answer number is 2.
එබැවින්, නිවැරදි පිළිතුරු අංකය 2 වේ.

7. Which of the following is NOT a common use case for nested if statements?
පහත සඳහන් ඒවායින් nested if ප්‍රකාශ සඳහා පොදු භාවිත අවස්ථාවක් නොවන්නේ කුමක්ද?

1) Validating user input. / පරිශීලක ආදානය වලංගු කිරීම.
2) Implementing a decision tree. / decision tree එකක් ක්‍රියාත්මක කිරීම.
3) Replacing a simple if-else structure. / සරල if-else ව්‍යුහයක් ප්‍රතිස්ථාපනය කිරීම.
4) Handling complex logical conditions. / සංකීර්ණ තාර්කික කොන්දේසි හැසිරවීම.
5) Creating menu-driven applications. / මෙනු මත පදනම් වූ යෙදුම් නිර්මාණය කිරීම.

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. Nested if statements are commonly used to validate user input by checking multiple conditions in a structured manner.
මෙම විකල්පයට අනුව, එය අසත්‍ය වේ. Nested if ප්‍රකාශයන් ව්‍යුහගත ආකාරයෙන් බහු කොන්දේසි පරීක්ෂා කිරීමෙන් පරිශීලක ආදානය වලංගු කිරීමට සාමාන්‍යයෙන් භාවිතා වේ.

- **Statement / ප්‍රකාශය 2)**

According to this option, it's false. Nested if statements are often used to implement decision trees, where conditions are checked in layers to make decisions.
මෙම විකල්පයට අනුව, එය අසත්‍ය වේ. Nested if ප්‍රකාශයන් බොහෝ විට decision tree එකක් ක්‍රියාත්මක කිරීමට භාවිතා කරයි, එහිදී තීරණ ගැනීම සඳහා ස්ථර වල කොන්දේසි පරීක්ෂා කරනු ලැබේ.

- **Statement / ප්‍රකාශය 3)**

According to this option, it's true. Nested if statements are not typically used to replace a simple if-else structure, which is already straightforward for basic conditional checks. Nested if statements are more suited for handling more complex conditions, rather than simplifying simple ones.

මෙම විකල්පය අනුව, එය සත්‍ය වේ. මූලික කොන්දේසි සහිත පරීක්ෂා කිරීම් සඳහා දැනටමත් සරල වන සරල if-else ව්‍යුහයක් ප්‍රතිස්ථාපනය කිරීමට Nested if ප්‍රකාශ සාමාන්‍යයෙන් භාවිතා නොවේ. Nested if ප්‍රකාශයන් සරල ඒවා සරල කරනවාට වඩා සංකීර්ණ තත්වයන් හැසිරවීමට වඩාත් සුදුසු වේ.

- **Statement / ප්‍රකාශය 4)**

According to this option, it's false. Nested if statements are useful for handling complex logical conditions by allowing multiple layers of checks.
මෙම විකල්පයට අනුව, එය අසත්‍ය වේ. බහුවිධ පරීක්ෂා කිරීම් වලට ඉඩ දීමෙන් සංකීර්ණ තාර්කික තත්ත්වයන් හැසිරවීමට Nested if ප්‍රකාශයන් ප්‍රයෝජනවත් වේ.

- **Statement / ප්‍රකාශය 5)**

According to this option, it's false. Nested if statements can be used in menu-driven applications to manage different choices and actions based on user selections.
මෙම විකල්පයට අනුව, එය අසත්‍ය වේ. පරිශීලක තේරීම් මත පදනම්ව විවිධ තේරීම් සහ ක්‍රියා කළමනාකරණය කිරීම සඳහා මෙනු මත පදනම් වූ යෙදුම්වල Nested if ප්‍රකාශ භාවිතා කළ හැක.

So, the correct answer number is 3.
එබැවින්, නිවැරදි පිළිතුරු අංකය 3 වේ.

8. What will be the output of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් වේවිද?

```
numbers = [1, 2, 3, 4, 5]
result = [num if num % 2 == 0 else num * 2 for num in numbers]
print(result)
```

1) `[2, 2, 6, 4, 10]` 2) `[1, 2, 3, 4, 5]` 3) `[2, 4, 6, 8, 10]`
4) `[1, 4, 3, 8, 5]` 5) `[2, 4, 6, 4, 10]`

. **Answer : 1)**

`num % 2 == 0:`
This checks if a number is even. If the number is even, it is kept as is (num). If the number is odd, it is multiplied by 2 (num * 2).
මෙය අංකයක් ඉරට්ටේ දැයි පරීක්ෂා කරයි. අංකය ඉරට්ටේ නම්, එය (num) එලෙසම ලෙස තබා ඇත. සංඛ්‍යාව ඔත්තේ නම්, එය 2 න් ගුණ කරනු ලැබේ (num * 2).

Let's go through the list / ලැයිස්තුව හරහා යමු:

1 (odd / ඔත්තේ) → 1 * 2 = 2
2 (even / ඉරට්ටේ) → remains 2 / ඉතිරි 2
3 (odd / ඔත්තේ) → 3 * 2 = 6
4 (even / ඉරට්ටේ) → remains 4 / ඉතිරි 4
5 (odd / ඔත්තේ) → 5 * 2 = 10

So, the final result list will be: `[2, 2, 6, 4, 10]`
එබැවින්, අවසාන ප්‍රතිඵල ලැයිස්තුව වනුයේ: `[2, 2, 6, 4, 10]`

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

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

```
games={"FarCry":8.4,"Dishonered":8.0,"Outlast":7.0,"Halo":4.8,"MassEffect":3
.0}
for name, grade in games.items():
    if grade>= 4.8:
    print(name)
```

1)
```
FarCry
Dishonered
Outlast
Halo
```

2)
```
Outlast
FarCry
Dishonered
```

3)
```
FarCry
Dishonered
Halo
```

4) `FarCry Dishonered Outlast Halo`

5) error

Answer : 5)

The code has an `indentation error` because the `print(name)` statement is not properly indented under the `if` condition. Now, let's consider the above python code step by step.
`print(name)` ප්‍රකාශය `if` කොන්දේසිය යටතේ නිසියාකාරව indentation වී නැති නිසා කේතයට `indentation error` එකක් ඇත. දැන් අපි ඉහත පයිතන් කේතය පියවරෙන් පියවර සලකා බලමු.

- **`games={"FarCry":8.4,"Dishonered":8.0,"Outlast":7.0,"Halo":4.8,"MassEffect":3.0}`**

A dictionary `games` is created with game names as keys and their respective grades as values.
Dictionary `games` නිර්මාණය කර ඇත්තේ ක්‍රීඩා නාම (game names) යතුරු ලෙසින් සහ ඒවායේ ශ්‍රේණි (grades) අගයන් ලෙසිනි.

- **`for name, grade in games.items():`**

This line iterates over each key-value pair (game name and grade) in the games dictionary.
මෙම පාඨයේ `games` dictionary එක් එක් යතුරු-අගය යුගල (ක්‍රීඩා නාමය සහ ශ්‍රේණි) මත පුනරාවර්තනය වේ.

- **`if grade>= 4.8:`**

This line checks if the grade of the current game is greater than or equal to 4.8. However, the next line should be indented to be part of this `if` block, but it is not.
මෙම පාඨයේ වත්මන් ක්‍රීඩාවේ ශ්‍රේණිය(grade) 4.8 ට වඩා වැඩි හෝ සමාන නම් පරීක්ෂා කරයි. කෙසේ වෙතත්, if block හි මිළඟ පේළිය ඉන්ඩෙන්ට් කළ යුතුය, නමුත් එය එසේ නොවේ.

- **`print(name)`**

Due to the lack of indentation, Python will throw an IndentationError because `print(name)` is expected to be indented as part of the `if` block.
indentation නොමැතිකම හේතුවෙන් පයිතන් IndentationError එකක් දමනු ඇත, මන්ද `print(name)`if block කොටස තුළට indent කිරීමට බලාපොරොත්තු වේ.

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

10. Which of the following is the correct way of implementing the if condition in python?
පයිතන් හි if කොන්දේසිය ක්‍රියාත්මක කිරීමේ නිවැරදි ක්‍රමය පහත ඒවායින් කවරක්ද?

A.
```
If a>b:
   print('a is greater')
```

B.
```
if a>b:
   print('a is greater')
```

C.
```
if a>b:
   print('a is greater')
```

1) A only. A පමණි.
2) B only. B පමණි.
3) C only. C පමණි.
4) A and B only. A හා B පමණි.
5) A and C only. A හා C පමණි.

Answer : 3)

The correct answer is 3).
නිවැරදි පිළිතුර 3) වේ.

Python is sensitive to indentation, and blocks of code must be indented correctly. In this case, the statement under the if condition, `print('a is greater')`, must be indented to indicate it is part of the if block.
Python indentation වලට සංවේදී වන අතර, කේත කොටස් නිවැරදිව indent කළ යුතුය. මෙම අවස්ථාවෙහිදී, if කොන්දේසිය යටතේ ඇති ප්‍රකාශය, `print('a is greater')`, if කොටසෙහි කොටසක් බව දැක්වීමට indent කළ යුතුය.

Option A is incorrect because the if should be written in lowercase (`if`, not `If`).
A විකල්පය වැරදියි මන්ද if යන්න කුඩා අකුරින් ලිවිය යුතු නිසා (`if`, `If` නොවේ).

Option B is incorrect because the `print` statement is not properly indented.
`print` ප්‍රකාශය නිසියාකාරව indent කර නොමැති නිසා B විකල්පය වැරදියි.

So, option C is the correct syntax.
එබැවින්, C විකල්පය නිවැරදි වාක්‍ය ඛණ්ඩයයි.