සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved]

Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 453 MARKING

2026/03/08
ictfromabc
Ravindu Bandaranayake

1. Which of the following(s) uses random access?
අහඹු ප්‍රවේශය භාවිතා කරන්නේ පහත සඳහන් කුමන උපාංග(ය) ද?

A. Magnetic tape (චුම්බක පටිය)
B. USB drive (සැනෙලි මතකය)
C. DVD (සංඛ්‍යාංක බහුවිධ තැටිය)

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

Answer : 3)

Random access / අහඹු ප්‍රවේශය
Random access allows data to be read from or written to any location directly and immediately. You don't need to follow specific sequence to access particular piece of data.This allows for quicker retrieval of data.
අහඹු ප්‍රවේශය ඕනෑම ස්ථානයක සිට සෘජුව සහ ක්ෂණිකව දත්ත කියවීමට හෝ ලිවීමට ඉඩ සලසයි. විශේෂිත දත්ත කොටසකට ප්‍රවේශ වීමට ඔබට නිශ්චිත අනුපිළිවෙළක් අනුගමනය කිරීමට අවශ්‍ය නොවේ. මෙය ඉක්මනින් දත්ත ලබා ගැනීමට ඉඩ සලසයි.

Sequential access / අනුක්‍රමික ප්‍රවේශය
Sequential access requires data to be accessed in a specific, ordered sequence. To reach a particular piece of data, all preceding data may need to be processed first.
අනුක්‍රමික ප්‍රවේශය සඳහා නිශ්චිත, සැකසූ අනුපිළිවෙලකට දත්ත ප්‍රවේශ වීම අවශ්‍යවේ. යම් දත්ත කොටසක් වෙත ළඟාවීමට සැකසූ දත්ත සියල්ල සැකසීමට අවශ්‍ය විය හැක.

Let's analyze the given statements / දී ඇති ප්‍රකාශ විශ්ලේෂණය කරමු.

A. **Magnetic tape / චුම්භකිත පටිය**
An example of a sequential access device. Data is stored in a linear sequence on the tape, so you need to fast-forward or rewind to reach a particular piece of data. This means accessing data involves reading through the tape in sequence, which is efficient for backup and archival purposes.
අනුක්‍රමික ප්‍රවේශ උපාංගයක උදාහරණයකි. දත්ත පටිය මත රේඛීය අනුපිළ ගබඩා කර ඇත, ඒබැවින් ඔබ යම් දත්ත කොටසකට ළඟා වීමට වේගයෙන් ඉදිරියට හෝ ආපසු හැරවීමට අවශ්‍ය වේ මෙයින් අදහස් කරන්නේ දත්ත වෙත ප්‍රවේශ වීම යනු අනුපිළිවෙලින් පටිය හරහා කියවීමයි. එය උපස්ථ සහ සංරක්ෂිත අරමුණු සඳහා කාර්යක්ෂම වේ.

B. **USB drive / සැනෙලි මතකය**
A USB drive allows for random access, meaning data can be read from or written to any location on the drive directly, without having to go through other data first
USB ධාවකයක් අහඹු ප්‍රවේශය සඳහා ඉඩ දෙයි, එයින් අදහස් කරන්නේ පළමුව වෙනත් දත්ත හරහා යාමකින් තොරව ධාවකයේ ඕනෑම ස්ථානයක සිට දත්ත කියවීමට හෝ ලිවීමට හැකි බවයි.

C. **DVD / සංඛ්‍යාංක බහුවිධ තැටිය**
DVDs also support random access, allowing the system to jump directly to any location on the disc to read or write data. This is essential for tasks like playing video from a specific point or loading software directly.
DVD අහඹු ප්‍රවේශයට ද සහාය වන අතර, පද්ධතියට දත්ත කියවීමට හෝ ලිවීමට තැටියේ ඕනෑම ස්ථානයකට කෙලින්ම පැනීමට ඉඩ සලසයි. නිශ්චිත ලක්ෂ්‍යයකින් වීඩියෝ වාදනය කිරීම හෝ මෘදුකාංග සෘජුවම පූරණය කිරීම වැනි කාර්යයන් සඳහා මෙය අත්‍යවශ්‍ය වේ.

As the USB drive and the DVD uses random access the correct answer is 3) B and C only.
USB ධාවකය සහ DVD අහඹු ප්‍රවේශය භාවිතා කරන බැවින් නිවැරදි පිළිතුර වනුයේ 3) B සහ C පමණි.

2. What is the stored-program concept?
ආචිත ක්‍රමලේඛ සංකල්පය යනු කුමක්ද?

1) The use of magnetic storage for data storage.
දත්ත ගබඩා කිරීම සඳහා චුම්බක ගබඩාව භාවිතා කිරීම.

2) Storing data and instructions in separate memory locations.
වෙනම මතක ස්ථානවල දත්ත සහ උපදෙස් ගබඩා කිරීම.

3) Storing both program instructions and data in the same memory space.
වැඩසටහන් උපදෙස් සහ දත්ත දෙකම එකම මතක අවකාශයක ගබඩා කිරීම.

4) Running a program without storing it in memory.
මතකයේ ගබඩා නොකර වැඩසටහනක් ක්‍රියාත්මක කිරීම.

5) Loading instructions directly from the CPU register.
CPU රෙජිස්තරවලින් සෘජුව උපදෙස් පූරණය කිරීම.

Answer : 3)

Option / විකල්පය 1
This refers to using magnetic media like hard drives or tapes for storing data, but it has no direct connection to the stored-program concept. The stored-program concept focuses on memory usage within the computer for instructions and data.
මෙය දත්ත ගබඩා කිරීම සඳහා දෘඪ තැටි හෝ චුම්බක පටි වැනි චුම්බක මාධ්‍ය භාවිතා කිරීම අදහස් කරයි, නමුත් එය ආචිත ක්‍රමලේඛ සංකල්පයට සෘජු සම්බන්ධයක් නොමැත. ආචිත ක්‍රමලේඛ සංකල්පය උපදෙස් සහ දත්ත සඳහා පරිගණකය තුළ මතක භාවිතය කෙරෙහි අවධානය යොමු කරයි.

Option / විකල්පය 2
This describes the Harvard architecture, where separate memory spaces are used for program instructions and data. The stored-program concept, however, is based on the von Neumann architecture, where they share the same memory space.
මෙය හාවර්ඩ් ආකෘතිය විස්තර කරයි, එහිදී වැඩසටහන් උපදෙස් සහ දත්ත සඳහා වෙනම මතක අවකාශයන් භාවිතා වේ. කෙසේ වෙතත්, ආචිත ක්‍රමලේඛ සංකල්පය, එකම මතක අවකාශය බෙදා ගන්නා වොන් නියුමාන් ආකෘතිය මත පදනම් වේ.

Option / විකල්පය 3
This is the correct answer of the stored-program concept. It allows a computer to fetch and execute instructions and manipulate data from the same memory location. This concept is fundamental to modern computing.
ආචිත ක්‍රමලේඛ සංකල්පයේ නිවැරදි පිළිතුර මෙයයි. එය පරිගණකයකට උපදෙස් ලබා ගැනීමට සහ ක්‍රියාත්මක කිරීමට සහ එකම මතක ස්ථානයේ සිට දත්ත හැසිරවීමට ඉඩ සලසයි. මෙම සංකල්පය නවීන පරිගණකකරණයට මූලික වේ.

Option / විකල්පය 4
This is not feasible in most computer systems. Programs must be stored in memory to be executed by the CPU. Running a program directly from external media, for instance, would still require loading instructions into memory.
බොහෝ පරිගණක පද්ධතිවල මෙය කළ නොහැක. CPU මඟින් ක්‍රියාත්මක කිරීමට වැඩසටහන් මතකයේ ගබඩා කළ යුතුය. උදාහරණයක් ලෙස, බාහිර මාධ්‍යයෙන් සෘජුවම වැඩසටහනක් ක්‍රියාත්මක කිරීම සඳහා, තවමත් මතකයට උපදෙස් පූරණය කිරීම අවශ්‍ය වේ.

Option / විකල්පය 5
Registers in the CPU are used for temporarily holding data during computation. However, program instructions are fetched from memory into the CPU registers for execution. This is not the essence of the stored-program concept.
CPU හි ඇති රෙජිස්තර ගණනය කිරීමේදී දත්ත තාවකාලිකව තබා ගැනීම සඳහා භාවිතා වේ. කෙසේ වෙතත්, ක්‍රමලේඛ උපදෙස් ක්‍රියාත්මක කිරීම සඳහා CPU රෙජිස්තර වෙත මතකයෙන් ලබා ගනී. ආචිත ක්‍රමලේඛ සංකල්පයේ හරය මෙය නොවේ.

So, the correct answer number here is 3.
එමනිසා, මෙහි නිවැරදි පිළිතුරු අංකය 3 වේ.

3. What is the answer when the following Boolean expression is converted to a POS expression after simplifying?
පහත බූලියානු ප්‍රකාශනය සුළු කර POS ප්‍රකාශනයක් බවට පරිවර්තනය කළ විට පිළිතුර කුමක්ද?

$(AD + \overline{D1})D + \overline{D}\overline{C}$

1) $AD + \overline{D}\overline{C}$ 2) $(A + D)\overline{C}$ 3) $\underline{(A + \overline{D})(D + \overline{C})}$ 4) $(A + C)(D + C)$ 5) $(A + \overline{C})D$

Answer : 3)

First let's trivialize this boolean expression.
පළමුව මෙම බූලියානු ප්‍රකාශනය සුළු කර සිටිමු.

$(AD + \overline{D1})D + \overline{D}\overline{C}$	
$(AD + \overline{D})D + \overline{D}\overline{C}$	Identity law / සර්වසාම්‍ය න්‍යාය
$(A + \overline{D})D + \overline{D}\overline{C}$	Redundancy law / සමතිරික්ත න්‍යාය
$AD + \overline{D}D + \overline{D}\overline{C}$	Distributive law / විඝටන න්‍යාය
$AD + 0 + \overline{D}\overline{C}$	Inverse law / ප්‍රතිලෝම න්‍යාය
$AD + \overline{D}\overline{C}$	Identity law / සර්වසාම්‍ය න්‍යාය

Now let's take the inverse of this to convert this expression to POS.
දැන් මෙම ප්‍රකාශනය POS බවට පරිවර්තනය කිරීම සඳහා මෙහි ප්‍රතිලෝමය ලබාගනිමු.

$\overline{AD + \overline{D}\overline{C}}$	
$\overline{AD}\,\overline{\overline{D}\overline{C}}$	De Morgan’s law / ද මුවාර් ප්‍රමේයය
$(\overline{A} + \overline{D})\overline{\overline{D}\overline{C}}$	De Morgan’s law / ද මුවාර් ප්‍රමේයය
$(\overline{A} + \overline{D})(\overline{\overline{D}} + \overline{\overline{C}})$	De Morgan’s law / ද මුවාර් ප්‍රමේයය
$(\overline{A} + \overline{D})(D + C)$	Double inverse law / ද්විත්ව ප්‍රතිලෝම න්‍යාය
$\overline{A}D + \overline{A}C + \overline{D}D + \overline{D}C$	Distributive law / විඝටන න්‍යාය
$\overline{A}D + \overline{A}C + 0 + \overline{D}C$	Inverse law / ප්‍රතිලෝම න්‍යාය
$\overline{A}D + \overline{A}C + \overline{D}C$	Identity law / සර්වසාම්‍ය න්‍යාය
$\overline{A}D + \overline{A}C.1 + \overline{D}C$	Identity law / සර්වසාම්‍ය න්‍යාය
$\overline{A}D + \overline{A}C.(D + \overline{D}) + \overline{D}C$	Inverse law / ප්‍රතිලෝම න්‍යාය
$\overline{A}D + \overline{A}CD + \overline{A}C\overline{D} + \overline{D}C$	Distributive law / විඝටන න්‍යාය
$\overline{A}D(1 + C) + \overline{D}C(\overline{A} + 1)$	Distributive law / විඝටන න්‍යාය
$\overline{A}D(1) + \overline{D}C(1)$	Identity law / සර්වසාම්‍ය න්‍යාය
$\overline{A}D + \overline{D}C$	Identity law / සර්වසාම්‍ය න්‍යාය

Now let's remove the inversion applied first.
දැන් පළමුව යෙදූ ප්‍රතිලෝමය ඉවත් කරගනිමු.

$\overline{\overline{A}D + \overline{D}C}$	
$\overline{\overline{A}D}\,\overline{\overline{D}C}$	De Morgan’s law / ද මුවාර් ප්‍රමේයය
$(\overline{\overline{A}} + \overline{D})(\overline{\overline{D}} + \overline{C})$	De Morgan’s law / ද මුවාර් ප්‍රමේයය
$(A + \overline{D})(D + \overline{C})$	Double inverse law / ද්විත්ව ප්‍රතිලෝම න්‍යාය

So, the correct answer number here is 3.
එමනිසා, මෙහි නිවැරදි පිළිතුරු අංකය 3 වේ.

4. Which utility software is primarily used for low-level management and optimization of computer hardware components?
පරිගණක දෘඩාංග සංරචක පහළ මට්ටමේ කළමනාකරණය සහ ප්‍රශස්තකරණය සඳහා මූලික වශයෙන් භාවිතා කරන උපයෝගිතා මෘදුකාංග මොනවාද?

   1) Disk Defragmenter / තැටි ප්‍රතිභාගිකරණ මෘදුකාංගය
   2) Device Driver / උපාංග ධාවකය
   3) Antivirus Software / ප්‍රතිවෛරස මෘදුකාංගය
   4) File Compression Utility / ගොනු සම්පීඩන මෘදුකාංග
   5) Registry Cleaner / රෙජිස්තර පිරිසිදු කරන මෘදුකාංග

Answer : 2)

1) **Disk Defragmenter:** තැටි ප්‍රතිභාගිකරණ මෘදුකාංගය:
This utility helps to rearrange fragmented data on a disk to improve performance. While it is related to disk optimization, it does not manage or optimize hardware components at a low level.
මෙම උපයෝගිතාව කාර්ය සාධනය වැඩි දියුණු කිරීම සඳහා තැටියක ඛණ්ඩනය වූ දත්ත නැවත සකස් කිරීමට උපකාරී වේ. එය තැටි ප්‍රශස්තිකරණයට සම්බන්ධ වන අතර, එය දෘඩාංග සංරචක අඩු මට්ටමක කළමනාකරණය හෝ ප්‍රශස්ත කිරීම සිදු නොකරයි.

2) **Device Driver:** උපාංග ධාවකය:
Device drivers are software programs that allow the operating system and other software to communicate with hardware components. They are essential for the proper functioning of hardware such as printers, graphics cards, and disk drives. They provide low-level management and can also optimize hardware performance by ensuring that hardware components are used effectively.
උපාංග ධාවකයන් යනු මෙහෙයුම් පද්ධතියට සහ අනෙකුත් මෘදුකාංගවලට දෘඩාංග සංරචක සමඟ සන්නිවේදනය කිරීමට ඉඩ සලසන මෘදුකාංග වැඩසටහන් වේ. මුද්‍රණ යන්ත්‍ර, ග්‍රැෆික් කාඩ්පත් සහ තැටි ධාවක වැනි දෘඩාංගවල නිසි ක්‍රියාකාරිත්වය සඳහා ඒවා අත්‍යවශ්‍ය වේ. ඔවුන් පහත මට්ටමේ කළමනාකරණයක් සපයන අතර දෘඩාංග සංරචක ඵලදායී ලෙස භාවිතා කරන බව සහතික කිරීම මගින් දෘඩාංග කාර්ය සාධනය ප්‍රශස්ත කළ හැක.

3) **Antivirus Software:** ප්‍රතිවෛරස මෘදුකාංගය:
This software is used to detect and remove malware from a computer. It does not handle low-level hardware management.
මෙම මෘදුකාංගය පරිගණකයකින් අනිෂ්ට මෘදුකාංග හඳුනා ගැනීමට සහ ඉවත් කිරීමට භාවිතා කරයි. එය පහත් මට්ටමේ දෘඩාංග කළමනාකරණය හසුරුවන්නේ නැත.

4) **File Compression Utility: ගොනු සම්පීඩන මෘදුකාංග:**
This software compresses files to save space or make them easier to transfer. It does not deal with hardware management.
මෙම මෘදුකාංගය ඉඩ ඉතිරි කර ගැනීමට හෝ ඒවා මාරු කිරීම පහසු කිරීමට ගොනු සම්පීඩනය කරයි. එය දෘඩාංග කළමනාකරණය සමඟ කටයුතු නොකරයි.

5) **Registry Cleaner: රෙජිස්තර පිරිසිදු කරන මෘදුකාංගය:**
This utility cleans up and fixes issues in the Windows registry. It is related to system maintenance but not specifically to low-level hardware management.
මෙම උපයෝගිතාව වින්ඩෝස් රෙජිස්තරයේ ඇති ගැටළු පිරිසිදු කර නිවැරදි කරයි. එය පද්ධති නඩත්තුවට සම්බන්ධ නමුත් විශේෂයෙන් අඩු මට්ටමේ දෘඩාංග කළමනාකරණයට නොවේ.

So, the correct answer is 2) Device Driver.
මේ අනුව, නිවැරදි පිළිතුර 2) උපාංග ධාවකයයි.

5. What is the SQL statement to delete all attributes in the BOOK table as author's data Edward?
කර්තෘගේ දත්ත ලෙස එඩ්වඩ් පොත් වගුවේ ඇති සියලුම උපලක්ෂණ මකා දැමීමට SQL ප්‍රකාශය කුමක් ද?

1) `DELETE WHERE Book.Author= 'Edward';`
2) `DELETE 'Edward' FROM Book;`
3) `DELETE *FROM Book WHERE Author= 'Edward';`
4) `DELETE FROM Book WHERE Author = 'Edward';`
5) `DELETE *FROM Author WHERE Book = 'Edward';`

Answer : 4)

Let's explain the statements in the above question.
ඉහත ප්‍රශ්නයේ ඇති ප්‍රකාශ මෙසේ පැහැදිලි කරමු.

- **Statement / ප්‍රකාශය 1)**
  This statement is incorrect because it is missing the FROM clause. In SQL, the DELETE statement needs to specify the table from which to delete rows.
  මෙම ප්‍රකාශය FROM වගන්තිය අස්ථානගත වී ඇති බැවින් එය වැරදිය. SQLහි, DELETE ප්‍රකාශය මඟින් පේළි මකා දැමිය යුතු වගුව සඳහන් කිරීමට අවශ්‍ය වේ.

- **Statement / ප්‍රකාශය 2)**
  This statement is incorrect because DELETE does not work with a value directly. You cannot specify a value like 'Edward' in this way for a DELETE operation.
  DELETE අගයක් සමඟ සෘජුව ක්‍රියා නොකරන නිසා මෙම ප්‍රකාශය වැරදියි. DELETE මෙහෙයුමක් සඳහා ඔබට Edward වැනි අගයක් මේ ආකාරයෙන් සඳහන් කළ නොහැක.

- **Statement / ප්‍රකාශය 3)**
  This statement is incorrect because the DELETE statement does not use the * wildcard. In SQL, you do not need to specify * to delete rows.
  DELETE ප්‍රකාශය * Wildcard භාවිතා නොකරන නිසා මෙම ප්‍රකාශය වැරදියි. SQL හි, පේළි මකා දැමීමට ඔබට * සඳහන් කිරීමට අවශ්‍ය නොවේ.

- **Statement / ප්‍රකාශය 4)**
  This is the correct statement. It correctly specifies the Book table and includes the WHERE clause to filter rows where the Author is 'Edward'. This SQL statement will delete all rows from the Book table where the Author column matches 'Edward'.
  මය නිවැරදි ප්‍රකාශයයි. එය පොත් වගුව නිවැරදිව සඳහන් කරන අතර කර්තෘ 'Edward' සිටින පේළි පෙරීමට WHERE වගන්තිය ඇතුළත් වේ. මෙම SQL ප්‍රකාශය කර්තෘ තීරුව 'Edward' ට ගැළපෙන පොත් වගුවෙන් සියලුම පේළි මකා දමයි.

- **Statement / ප්‍රකාශය 5)**
  This statement is incorrect for several reasons. It tries to delete from the Author table instead of the Book table. The `DELETE` statement does not use.
  හේතු කිහිපයක් නිසා මෙම ප්‍රකාශය වැරදිය එය පොත් වගුව වෙනුවට කර්තෘ වගුවෙන් මකා දැමීමට උත්සාහ කරයි. `DELETE` ප්‍රකාශය භාවිතා නොකරයි.

Therefore, the correct answer, according to the above explanation is **4)**.
එබැවින්, ඉහත පැහැදිලි කිරීම අනුව නිවැරදි පිළිතුර 4) වේ.

7. What will be the output of the following Python code?
පහත දැක්වෙන පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
def set_operations(set1, set2):
    return set1.difference(set2), set2.difference(set1)

a = {1, 2, 3, 4}
b = {3, 4, 5, 6}
diff_a_b, diff_b_a = set_operations(a, b)

print("Difference A-B:", diff_a_b)
print("Difference B-A:", diff_b_a)
```

1) `Difference A-B: {5, 6}`
   `Difference B-A: {1, 2}`

2) `Difference A-B: {3, 4}`
   `Difference B-A: {3, 4}`

3) `Difference A-B: {1, 2}`
   `Difference B-A: {5, 6}`

4) `Difference A-B: {1, 2, 5, 6}`
   `Difference B-A: {}`

5) `Difference A-B: {}`
   `Difference B-A: {}`

Answer : 3)

The `set_operations()` function takes two sets as input (`set1` and `set2`) and returns two values:
`set_operations()` ශ්‍රිතය ආදානය `set1` සහ `set2` ලෙස කට්ටල දෙකක් ගෙන අගයන් දෙකක් ලබා දෙයි:

`set1.difference(set2):`
This returns the elements that are in set1 but not in set2.
මෙය set1 හි ඇති නමුත් set2 හි නොමැති මූලාංග නැවත ලබා දෙයි.

`set2.difference(set1):`
This returns the elements that are in set2 but not in set1.
මෙය set2 හි ඇති නමුත් set1 හි නොමැති මූලාංග නැවත ලබා දෙයි.

In this case/ මේ අවස්ථාවේ දී:
`Set a = {1, 2, 3, 4}`
`Set b = {3, 4, 5, 6}`

`a.difference(b)`
Elements in a but not in b / a හි ඇති නමුත් b හි නොමැති මූලාංග = {1, 2}

`b.difference(a)`
Elements in b but not in a / b හි ඇති නමුත් a හි නොමැති මූලාංග = {5, 6}

Therefore, the correct answer is, 3).
එබැවින්, නිවැරදි පිළිතුර වන්නේ, 3).

8. Consider the following Python program.
පහත පයිතන් වැඩසටහන සලකා බලන්න.

```
player_file = open("playerData.txt", "r")
summary_file = open("summaryData.txt", "w")
threshold = 10
for line in player_file:
    if int(line.split()[1]) > threshold:
        summary_file.write(line)
player_file.close()
summary_file.close()
```

Which of the following statements are correct about the above program ?
ඉහත වැඩසටහන සම්බන්ධයෙන් පහත සඳහන් කුමන ප්‍රකාශයන් නිවැරදිද ?

A. The program reads player data from `playerData.txt` one line at a time.
වැඩසටහන `playerData.txt` වෙතින් ක්‍රීඩක දත්ත වරකට එක පේළියකින් කියවයි.

B. Player scores greater than 10 are written to `summaryData.txt`.
10ට වඩා වැඩි ක්‍රීඩකයන්ගේ ලකුණු `summaryData.txt` වෙත ලියා ඇත.

C. If `playerData.txt` does not exist, the program will raise an error at runtime.
`playerData.txt` නොපවතියි නම්, වැඩසටහන ධාවන වේලාවේදී දෝෂයක් මතු කරයි.

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) <u>All of the above.</u>
ඉහත සියල්ලම.

Answer : 5)

Let's consider the code line by line / කේතය පේළියෙන් පේළිය සලකා බලමු:

- **`player_file = open("playerData.txt", "r"):`**
  Opens playerData.txt in read mode.
  playerData.txt කියවීමේ ආකාරයෙන් විවෘත කරයි.

- **`summary_file = open("summaryData.txt", "w"):`**
  Opens summaryData.txt in write mode.
  ලිවීමේ ආකාරයෙන් විවෘත කරයි.

- **`threshold = 10:`**
  Sets a threshold value of 10.
  threshold අගය 10 සකසයි.

- **`for line in player_file:`**
  Reads the file playerData.txt line by line.
  playerData.txt ගොනුව පේළියෙන් පේළිය කියවයි.

- **`if int(line.split()[1]) > threshold:`**
  Splits each line by whitespace and checks if the second element (index 1) converted to an integer is greater than 10.
  සෑම පේළියක්ම whitespace මගින් බෙදන අතර දෙවන මූලද්‍රව්‍යය (දර්ශක 1) පූර්ණ සංඛ්‍යාවකට පරිවර්තනය කර 10 ට වඩා වැඩි දැයි පරීක්ෂා කරයි.

- **`summary_file.write(line):`**
  Writes the line to summaryData.txt if the condition is met.
  කොන්දේසිය සපුරා ඇත්නම් රේඛාව summaryData.txt වෙත ලියයි.

- Closes both files using player_file.close() and summary_file.close().
  player_file.close() සහ summary_file.close() භාවිතයෙන් ගොනු දෙකම වසා දමයි.

Analysis of Statements / ප්‍රකාශ විශ්ලේෂණය:

Statement / ප්‍රකාශය A
True. The for line in player_file: statement reads the file line by line.
සත්‍ය වේ. for line in player_file: ප්‍රකාශය ගොනුව පේළියෙන් පේළිය කියවයි.

Statement / ප්‍රකාශය B
True. The condition if int(line.split()[1]) > threshold: checks if the second item in the line is greater than 10, and if true, the line is written to summaryData.txt.
සත්‍ය වේ. කොන්දේසිය if int(line.split()[1]) > threshold: පේළියේ දෙවන අයිත්තමය 10 ට වඩා වැඩි දැයි පරීක්ෂා කරයි, සහ සත්‍ය නම්, රේඛාව summaryData.txt වෙත ලියා ඇත.

Statement / ප්‍රකාශය C
True. If the file playerData.txt does not exist, trying to open it in read mode ("r") will raise a FileNotFoundError.
සත්‍ය වේ. playerData.txt ගොනුව නොපවතියි නම්, එය read mode ("r") විවෘත කිරීමට උත්සාහ කිරීම FileNotFoundError මතු කරයි.

The correct answer is 5) All of the above, as all the statements are true.
සියලුම ප්‍රකාශ සත්‍ය බැවින් නිවැරදි පිළිතුර 5) ඉහත සියල්ලම වේ.

9. Which of the following pairs correctly lists examples of code editors?
පහත සඳහන් යුගල අතුරින් කේත සංස්කාරක සඳහා පමණක් උදාහරණ නිවැරදිව ලැයිස්තුගත කර ඇත්තේ කුමක්ද?

1) Atom, Audacity
2) VS Code, Sublime Text
3) CorelDRAW, Notepad
4) PowerPoint, Atom
5) Atom, Eclipse

Answer : 2)

Explanation / පැහැදිලි කිරීම

A code editor is a type of software used by programmers to write, edit, and modify source code for computer programs.
කේත සංස්කාරකයක් යනු පරිගණක වැඩසටහන් සඳහා මූලාශ්‍ර කේත ලිවීමට, සංස්කරණය කිරීමට සහ වෙනස් කිරීමට ක්‍රමලේඛකයින් විසින් භාවිතා කරන මෘදුකාංග වර්ගයකි.

Both VS Code (Visual Studio Code) and Sublime Text are popular code editor. Other options listed involve applications that are not primarily used for code editing (e.g., Audacity is for audio editing, CorelDRAW is for graphic design, PowerPoint is for presentations).
VS Code (Visual Studio Code) සහ Sublime Text යන දෙකම ජනප්‍රිය කේත සංස්කාරක වේ. ලැයිස්තුගත කර ඇති අනෙකුත් විකල්ප අතර කේත සංස්කරණය සඳහා ප්‍රධාන වශයෙන් භාවිතා නොකරන යෙදුම් ඇතුළත් වේ (උදා: Audacity ශ්‍රව්‍ය සංස්කරණය සඳහා, CorelDRAW ග්‍රැෆික් නිර්මාණය සඳහා, PowerPoint, presentations සඳහා යොදාගනී.).

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

10. Which of the following statements about CSS specificity and rule application is true?
CSS නිශ්චිතභාවය සහ රීති යෙදුම පිළිබඳ පහත සඳහන් ප්‍රකාශවලින් සත්‍ය කුමක්ද?

1) An element with a class selector always overrides an ID selector.
පන්ති වරකයක් සහිත මූලාංගයක් සැම විටම ID වරකයක් අභිබවා යයි.
2) Inline styles have lower specificity than external CSS files.
පේළිගත විලාසයන් බාහිර CSS ගොනු වලට වඩා අඩු නිශ්චිතතාවයක් ඇත.
3) If two selectors have the same specificity, the one defined last in the CSS file wins.
වරක දෙකක් එකම නිශ්චිතතාවයක් තිබේ නම්, CSS ගොනුවේ අවසාන වශයෙන් අර්ථ දක්වා ඇති එක ජය ගනී.
4) Universal selector (*) has higher specificity than element selectors.
Universal වරකය (*) මූලාංග වරකවලට වඩා ඉහළ නිශ්චිතතාවයක් ඇත.
5) External CSS always overrides inline styles regardless of specificity.
නිශ්චිතභාවය නොසලකා බාහිර CSS සැම විටම පේළිගත විලාසයන් අභිබවා යයි.

Answer: 3)

Explanation / පැහැදිලි කිරීම:

This is correct. When two rules apply to the same element and have equal specificity, the browser chooses the one that appears later in the CSS file. The idea is that newer rules should have the power to override earlier ones if they target the same thing with the same weight.
මෙය නිවැරදියි. එකම මූලාංගයකට නීති දෙකක් අදාළ වන විට සහ සමාන නිශ්චිතතාවයක් ඇති විට, බ්‍රව්සරය CSS ගොනුවේ පසුව දිස්වන එක තෝරා ගනී. අදහස නම්, නව නීති එකම බරකින් එකම දේ ඉලක්ක කරන්නේ නම් පෙර ඒවා අභිබවා යාමට බලය තිබිය යුතු බවයි.

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

Why the other options are incorrect / අනෙක් විකල්ප වැරදි වන්නේ ඇයි

Option/ විකල්ප 1)
This is incorrect. In CSS, ID selectors are more specific than class selectors. An ID selector carries more weight compared to a class selector
මෙය වැරදියි. CSS හි, ID වරක පන්ති වරකවලට වඩා නිශ්චිත වේ. ID වරකයක් පන්ති වරකයකට සාපේක්ෂව වැඩි බරක් දරයි

Option/ විකල්ප 2)
This is incorrect. Inline styles are applied directly to an element using the style attribute. Inline styles have higher specificity than both internal and external CSS rules.
මෙය වැරදියි. විලාස ගුණාංගය භාවිතා කරමින් මූලාංගයකට සෘජුවම යොදනු ලැබේ. අභ්‍යන්තර සහ බාහිර CSS නීති දෙකටම වඩා ඉහළ නිශ්චිතතාවයක් පේළිගත විලාස සතුව ඇත.

Option/ විකල්ප 4)
This is incorrect. The universal selector has the lowest possible specificity. Element selectors have a higher specificity than *. So any element-specific rule will override a rule using the universal selector if there's a conflict.
මෙය වැරදියි. Universal වරකවලට හැකි අවම නිශ්චිතතාවයක් ඇත. මූලාංග වරකවලට * වඩා ඉහළ නිශ්චිතතාවයක් ඇත. එබැවින් ගැටුමක් තිබේ නම් ඕනෑම මූලාංග-විශේෂිත රීතියක් universal වරක භාවිතා කරමින් රීතියක් අභිබවා යනු ඇත.

Option/ විකල්ප 5)
This is incorrect because external CSS styles do not always override inline styles. In CSS specificity hierarchy, inline styles have the highest priority compared to styles defined in external or internal style sheets.
බාහිර CSS මෝස්තර සැම විටම පේළිගත විලාස අභිබවා නොයන නිසා මෙය වැරදියි. CSS නිශ්චිතතා ධුරාවලිය තුළ, බාහිර හෝ අභ්‍යන්තර විලාස පත්‍රවල අර්ථ දක්වා ඇති විලාසයන්ට සාපේක්ෂව, පේළිගත විලාසයන්ට ඉහළම ප්‍රමුඛතාවය හිමි වේ.