සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved]

Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 229 MARKING

2025/07/27
ictfromabc
Ravindu Bandaranayake

1. The time taken by the actuator arm of a hard disk to move the header to the sector containing the relevant data is called by which name?
දෘඪ තැටියක ඇක්චුවේටර් බාහුව මගින් ශීර්ෂකය අදාළ දත්තය ඇතුළත් sector එක වෙතට යොමුවීමට ගතවන කාලය කුමන නමකින් හැඳින්වේද?

1) Seek time 2) Rotational delay 3) Access time 4) cluster time 5) Sector delay

Answer: 1)

Let's consider each of these applications related to hard disk.
දෘඪ තැටියට අදාළ මෙම එක් එක් යෙදුම් පිළිබඳව සලකා බලමු.

Seek Time:
The hard disk's data is stored in concentric circles known as tracks. When data is requested, the hard disk must move the head to the track containing that data. Seek time is the time it takes for the actuator arm of the hard disk to move the read/write head to the correct track where the data is located. Seek time is one of the most critical factors in determining the speed of a hard disk.
දෘඪ තැටියේ දත්ත, track ලෙස හඳුන්වන පථ තුළ ගබඩා කර ඇත. දත්ත ඉල්ලා සිටින විට, දෘඪ තැටිය එම දත්ත අඩංගු track එකට ශීර්ෂකය ගෙන යා යුතුය. Seek time යනු දෘඪ තැටියේ ඇක්චුවේටර් බාහුවට කියවීමේ/ලිවීමේ හිස දත්ත ඇති නිවැරදි track එකට ගෙනයාමට ගතවන කාලයයි. දෘඪ තැටියක වේගය තීරණය කිරීමේදී seek time වඩාත් තීරණාත්මක සාධකයක් වේ.

Rotational Delay (Latency):
Rotational delay, also known as rotational latency, is the time it takes for the desired sector of the disk to rotate under the read/write head after the head is positioned over the correct track. Once the head is over the right track (after the seek time), the disk must rotate until the correct sector is under the head. The average rotational delay depends on the speed at which the disk spins (measured in RPM).
Rotational delay, rotational latency ලෙසද හැඳින්වේ, නිවැරදි track එකෙහි ශීර්ෂකය ස්ථානගත කිරීමෙන් පසු තැටියේ අපේක්ෂිත sector එක කියවීමේ/ලිවීමේ ශීර්ෂකය යටතට භ්‍රමණය වීමට ගතවන කාලයයි. ශීර්ෂකය නිවැරදි track එකට ගිය පසු (seek time එකෙන් පසු), නිවැරදි sector එක ශීර්ෂකයට යටින් ඇති තෙක් තැටිය භ්‍රමණය විය යුතුය. සාමාන්‍යයෙන් Rotational delay එක තැටිය භ්‍රමණය වන වේගය මත රඳා පවතී (RPM) වලින් මනිනු ලැබේ.

Access Time:
Access time is the total time it takes to retrieve data from the disk. It is the sum of the seek time, the rotational delay and read/write time.
ප්‍රවේශ කාලය යනු තැටියෙන් දත්ත ලබාගැනීමට ගතවන මුළු කාලයයි. එය seek time, rotational delay සහ කියවීමේ/ලිවීමේ කාලයෙහි එකතුවයි.

Terms such as cluster time, sector delay mentioned in the answers here do not exist.
මෙහි පිළිතුරුවල සඳහන් cluster time, sector delay ලෙස යෙදුම් නොපවතී.

So, the correct answer number here is 1).
එමනිසා, මෙහි නිවැරදි පිළිතුරු අංකය 1) වේ.

2. Which device cannot be categorized under both input and output devices?
ආදාන සහ ප්‍රතිදාන උපාංග දෙක යටතේ වර්ගීකරණය කළ නොහැකි උපාංගය කුමක් ද?

1) MICR
2) Headsets
3) Electronic whiteboard
4) Network Cards
5) Modems

Answer: 4)

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

1) MICR
According to this option, it's false. MICR is used to read and input data from documents like checks, so it functions as an input device. It does not typically serve as an output device.
මෙම විකල්පයට අනුව, නිවැරදි නොවේ. චෙක්පත් වැනි ලේඛනවල දත්ත කියවීමට සහ ඇතුළත් කිරීමට MICR භාවිතා කරයි, එබැවින් එය ආදාන උපාංගයක් ලෙස ක්‍රියා කරයි. එය සාමාන්‍යයෙන් ප්‍රතිදාන උපාංගයක් ලෙස ක්‍රියා නොකරයි.

2) Headsets
According to this option, it's false. Headsets are used for both input (via microphones) and output (via speakers), so they can be categorized under both input and output devices.
මෙම විකල්පයට අනුව, නිවැරදි නොවේ. හෙඩ්සෙට් ආදාන (මයික්‍රොෆෝන හරහා) සහ ප්‍රතිදානය (ස්පීකර් හරහා) යන දෙකටම භාවිතා කරයි, එබැවින් ඒවා ආදාන සහ ප්‍රතිදාන උපාංග දෙක යටතේ වර්ග කළ හැක.

3) Electronic whiteboard
According to this option, it's false. Electronic whiteboards can be used for input (writing or drawing) and output (displaying information), so they fit both categories.
මෙම විකල්පයට අනුව, එය නිවැරදි නොවේ. ඉලෙක්ට්‍රොනික වයිට්බෝඩ් ආදානය (ලිවීම හෝ ඇඳීම) සහ ප්‍රතිදානය (තොරතුරු ප්‍රදර්ශනය කිරීම) සඳහා භාවිතා කළ හැක, එබැවින් ඒවා කාණ්ඩ දෙකටම ගැළපේ.

4) Network Cards
According to this option, it's true. Network cards are primarily used for input and output of network data but are not categorized as both input and output devices in the traditional sense. They handle data transmission but do not serve as input or output devices in the broader context.
මෙම විකල්පය අනුව, නිවැරදි වේ. ජාල කාඩ්පත් මූලික වශයෙන් ජාල දත්ත ආදානය සහ ප්‍රතිදානය සඳහා භාවිතා කරන නමුත් සම්ප්‍රදායික අර්ථයෙන් ආදාන සහ ප්‍රතිදාන උපාංග ලෙස වර්ගීකරණය නොකෙරේ. ඔවුන් දත්ත සම්ප්‍රේෂණය හසුරුවන නමුත් පුළුල් සන්දර්භය තුළ ආදාන හෝ ප්‍රතිදාන උපාංග ලෙස ක්‍රියා නොකරයි.

5) Modems
According to this option, it's false. Modems can function as both input and output devices by converting digital data to analog signals and vice versa, so they can be categorized under both types.
මෙම විකල්පයට අනුව, නිවැරදි නොවේ. මොඩමයන්ට ඩිජිටල් දත්ත ඇනලොග් සංඥා බවට පරිවර්තනය කිරීමෙන් ආදාන සහ ප්‍රතිදාන උපාංග ලෙස ක්‍රියා කළ හැකි අතර, ඒවා වර්ග දෙකම යටතේ වර්ග කළ හැක.

The correct answer is 4).
නිවැරදි පිළිතුර 4) වේ.

3. What is the most suitable answer for getting the one's complement of a binary number?
ද්විමය සංඛ්‍යාවක එකෙහි අනුපූරකය ලබා ගැනීම සඳහා වඩාත් ගැළපෙන පිළිතුර වන්නේ කුමක් ද?

   1) By flipping all the bits / සියලුම බිටු පෙරලීමෙන්.
   2) By adding 1 to the binary number / ද්විමය අංකයට 1 එකතු කිරීමෙන්.
   3) By subtracting 1 from the binary number / ද්විමය අංකයෙන් 1 අඩු කිරීමෙන්.
   4) By flipping all the bits only if number is negative / සෘණ සංඛ්‍යාවක් නම් පමණක් සියලුම බිටු පෙරලීමෙන්.
   5) By performing a bitwise AND operation with all 1s
      සියලු 1 සමග බිටු අනුසාරිත AND මෙහෙයුමක් සිදු කිරීමෙන්.

   Answer: 4)

   Explanation / පැහැදිලි කිරීම:

   The one's complement of a binary number is a method of representing negative numbers in binary form.
   ද්විමය සංඛ්‍යාවක 1 හි අනුපූරකය යනු සෘණ සංඛ්‍යා ද්විමය ආකාරයෙන් නිරූපණය කිරීමේ ක්‍රමයකි.

   In one's complement representation, positive numbers are represented exactly the same way as they are in standard binary notation. The bits remain unchanged for positive numbers.
   1 හි අනුපූරක නියෝජනයේදී, ධනාත්මක සංඛ්‍යා සාමාන්‍ය ද්විමය අංකනයෙහි දැක්වෙන ආකාරයටම නිරූපණය කෙරේ. ධන සංඛ්‍යා සඳහා බිටු නොවෙනස්ව පවතී.

   In one's complement, a negative number is represented by taking the one's complement of its positive counterpart. For example, if the binary number `1010` represents +10, its one's complement `0101` would represent -10 in a one's complement system.
   1 හි අනුපූරකයේ, සෘණ සංඛ්‍යාවක් නිරූපණය කරනුයේ එහි ධන ප්‍රතිපක්ෂයේ අනුපූරකය ගැනීමෙන් ය. උදාහරණයක් ලෙස, ද්විමය අංකය `1010`, +10 නියෝජනය කරන්නේ නම්, එහි 1 හි අනුපූරකය `0101` 1 හි අනුපූරක පද්ධතියක -10 නියෝජනය කරයි.

   Therefore, the correct answer is 4) By flipping all the bits only if the number is negative.
   එබැවින්, නිවැරදි පිළිතුර වන්නේ 4) සියලුම බිටු පෙරලීමෙන් අංකය සෘණ නම් පමණි.

4. $56_{16} + A2_{16} - 21_{16} =$ ..... ?

   1) $F8_{16}$ 2) $D6_{16}$ 3) $\underline{D7_{16}}$ 4) $F9_{16}$ 5) $6D_{16}$

   Answer: 3)

   Explanation / පැහැදිලි කිරීම:

$$\begin{array}{rcl} & 5 & 6_{16} \\ + & A & 2_{16} \\ \hline & F & 8_{16} \end{array}$$

$$\begin{array}{rcl} & F & 8_{16} \\ - & 2 & 1_{16} \\ \hline & D & 7_{16} \end{array}$$

   Therefore, the correct answer is 3) $D7_{16}$.
   එබැවින්, නිවැරදි පිළිතුර 3) $D7_{16}$ වේ.

5. $D\overline{D} + \overline{C(BA + DA)}$

After simplifying the above boolean expression, what is the total number of gates required to construct it using OR and NOT gates?
ඉහත බූලිය ප්‍රකාශනය සුළු කිරීමෙන් පසුව එය OR සහ NOT ද්වාර භාවිතයෙන් ගොඩ නැගීමට අවශ්‍ය මුළු ද්වාර ගණන කොපමණ ද?

1) 5 2) 7 3) 9 4) 10 5) 11

Answer: 2)

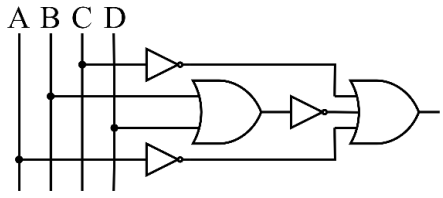
Explanation / පැහැදිලි කිරීම:

Let's simplify the above Boolean expression using Boolean Laws.
බූලියානු නීති යොදාගනිමින් ඉහත බූලියානු ප්‍රකාශනය සුළු කරමු.

Expression : $D\overline{D} + \overline{C(BA + DA)}$

$= D\overline{D} + \overline{C} + (\overline{BA + DA})$	De morgan's Law ඩි මෝගන් න්‍යාය
$= 0 + \overline{C} + (\overline{BA + DA})$	Inverse Law ප්‍රතිලෝම න්‍යාය
$= \overline{C} + (\overline{BA + DA})$	Identity Law සර්වසාම්‍ය න්‍යාය
$= \overline{C} + (\overline{BA}.\overline{DA})$	De morgan's Law ඩි මෝගන් න්‍යාය
$= \overline{C} + ((\overline{B} + \overline{A}).(\overline{D} + \overline{A}))$	De morgan's Law ඩි මෝගන් න්‍යාය
$= \overline{C} + (\overline{B}\overline{D} + \overline{B}\overline{A} + \overline{A}\overline{D} + \overline{A}\overline{A})$	Distributive Law විෂටන න්‍යාය
$= \overline{C} + (\overline{B}\overline{D} + \overline{B}\overline{A} + \overline{A}\overline{D} + \overline{A})$	Idempotent Law තදේවභාවී න්‍යාය
$= \overline{C} + (\overline{B}\overline{D} + \overline{B}\overline{A} + \overline{A}(\overline{D} + 1))$	Distributive Law විෂටන න්‍යාය
$= \overline{C} + (\overline{B}\overline{D} + \overline{B}\overline{A} + \overline{A}(1))$	Identity Law සර්වසාම්‍ය න්‍යාය
$= \overline{C} + (\overline{B}\overline{D} + \overline{B}\overline{A} + \overline{A})$	Identity Law සර්වසාම්‍ය න්‍යාය
$= \overline{C} + (\overline{B}\overline{D} + \overline{A}(\overline{B} + 1)$	Distributive Law විෂටන න්‍යාය
$= \overline{C} + (\overline{B}\overline{D} + \overline{A}(1)$	Identity Law සර්වසාම්‍ය න්‍යාය
$= \overline{C} + (\overline{B}\overline{D} + \overline{A})$	Identity Law සර්වසාම්‍ය න්‍යාය
$= \overline{C} + \overline{B}\overline{D} + \overline{A}$	
$= \overline{C} + \overline{\overline{B}\overline{D}} + \overline{A}$	(Apply double complement law)
$= \overline{C} + \overline{\overline{B} + \overline{D}} + \overline{A}$	De morgan's Law ඩි මෝගන් න්‍යාය
$= \overline{C} + \overline{B + \overline{D}} + \overline{A}$	Double complement Law ද්විත්ව ප්‍රතිලෝම න්‍යාය
$= \overline{C} + \overline{B + D} + \overline{A}$	Double complement Law ද්විත්ව ප්‍රතිලෝම න්‍යාය

Now, let's draw the respective logical circuit for the above simplified expression using only OR Gates and NOT gates.
දැන්, OR ද්වාර සහ NOT ද්වාර පමණක් භාවිතා කර ඉහත සරල කළ ප්‍රකාශනය සඳහා අදාළ තාර්කික පරිපථය අඳිමු.



The total number of logic gates required to implement above expression using OR and NOT gates is 5.
OR සහ NOT ද්වාර භාවිතයෙන් ඉහත ප්‍රකාශනය ක්‍රියාත්මක කිරීමට අවශ්‍ය මුළු තාර්කික ද්වාර ගණන 5 කි.

The answer should be number 2.
පිළිතුර අංක 2 යුතුය.

6. $\overline{\overline{CA}\overline{C}} + \overline{\overline{B}\overline{B}}$

What is the answer when the above Boolean expression is trivialized?
ඉහත බූලිය ප්‍රකාශනය සුළු කළ විට පිළිතුර වන්නේ කුමක් ද?

1) $\overline{\overline{B}}$ 2) $\overline{A\overline{C}} + 0$ 3) $\overline{CA}$ 4) $\underline{1}$ 5) 0

Answer: 4)

Explanation / පැහැදිලි කිරීම:

This Boolean expression can be simplified as follows.
මෙම බූලියානු ප්‍රකාශනය පහත ආකාරයට සුළු කළ හැකිය.

$\overline{\overline{CA}\overline{C}} + \overline{\overline{B}\overline{B}}$

$\overline{\overline{CA}\overline{C}} + B\overline{B}$	Inverse Law
$\overline{C} + \overline{A} + \overline{\overline{C}} + B\overline{B}$	Demorgan Law
$\overline{C} + \overline{A} + C + B\overline{B}$	Inverse Law
$\overline{A} + B\overline{B} + 1$	Complement Law
1	Identity Law

So, the correct answer number here is 4.
එමනිසා, මෙහි නිවැරදි පිළිතුරු අංකය 4 වේ.

7. Which of the following statements is equivalent to the Boolean statement?
පහත සඳහන් කුමන ප්‍රකාශය බූලිය ප්‍රකාශයට සමාන ද?

0.A+1.(D.B+C.1)

1) C 2) D 3) D.B 4) $\underline{\text{D.B+C}}$ 5) 1

Answer: 4)

Explanation / පැහැදිලි කිරීම:

This Boolean expression can be simplified as follows.
මෙම බූලියානු ප්‍රකාශනය පහත ආකාරයට සුළු කළ හැකිය.

$= 0A + 1(DB + C1)$	
$= 0 + 1(DB + C1)$	Identity Law (සර්වසාමය නියාය)
$= 1(DB + C1)$	Identity Law (සර්වසාමය නියාය)
$= DB + C1$	Identity Law (සර්වසාමය නියාය)
$= DB + C$	Identity Law (සර්වසාමය නියාය)

The correct answer is 4).
නිවැරදි පිළිතුර 4) වේ.

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

```
a = 100
b = 5
print((a >> b) + (a >> 2))
```

1) 20 2) 25 3) 28 4) 30 5) 18

Answer: 3)

Explanation / පැහැදිලි කිරීම:

In this Python code, the bitwise right shift operator >> is used, which effectively divides the number by powers of 2.
මෙම පයිතන් කේතයේ, bitwise right shift operator >> **භාවිතා කර ඇති අතර, එමඟින් සංඛ්‍යාව** 2 **හි බලවලින් වලදායී ලෙස බෙදනු ලැබේ.**

The variable a is assigned the value 100, and b is assigned 5
a **විචල්‍යයට** 100 **අගය පවරන අතර** b **ට** 5 **පවරන ලදී.**

When we evaluate a >> b, we are shifting the bits of 100 to the right by 5 positions.
a >> b **අගයක් ඇගයීමේදී, අපි** 100 **හි බිටු ස්ථාන** 5 **කින් දකුණට මාරු කරන්නෙමු.**

100 >>5

100	64	32	16	8	4	2	1						
	1	1	0	0	1	0	0						
	64	32	16	8	4	2	1						
						1	1	0	0	1	0	0	→
	$11_2 = 3$												

Since right-shifting by n bits is the same as performing integer division by $\mathbf{2^n}$
N **බිටු වලින් දකුණට මාරු කිරීම** $\mathbf{2^n}$ **වලින් පූර්ණ සංඛ්‍යා බෙදීම සිදු කිරීමට සමාන බැවින්**

This becomes $100/2^5$
මෙය $100/2^5$ **බවට පත්වේ**
=100 /32 = 3 (only the integer part is kept / **පූර්ණ සංඛ්‍යා කොටස පමණක් තබා ඇත.**)

100 >>2

100	64	32	16	8	4	2	1			
	1	1	0	0	1	0	0			
	64	32	16	8	4	2	1			
			1	1	0	0	1	0	0	→
	$11001_2 = 16 + 8 + 1 = 25$									

Similarly, a >> 2 means / **ඒ හා සමානව,** a >> 2 **යන්නෙන් අදහස් කරන්නේ:**
$100 / 2^2 = 100 / 4$
=25

Finally, the program adds both results: 3 + 25 = 28. Therefore, the output printed by this code is 28.
අවසාන වශයෙන්, වැඩසටහන ප්‍රතිඵල දෙකම එකතු කරයි: 3 + 25 = 28. **එබැවින්, මෙම කේතය මඟින් මුද්‍රණය කරන ලද ප්‍රතිදානය** 28 **වේ.**

NOTE: Operator precedence / මෙහෙයුම් ප්‍රමුඛතාවය:

Parentheses	( )
Multiplication Division	* /
Addition subtraction	+ -
Left shift Right shift	<< >>

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

9. Which of the following is not true about interpreters?
අර්ථ විනිවිදීම සම්බන්ධයෙන් පහත සඳහන් ප්‍රකාශවලින් සත්‍ය නොවන්නේ කුමක් ද?

1) It translates program statements, line by line, into machine code.
එය වැඩසටහන් ප්‍රකාශයන්, පේළියෙන් පේළිය, යන්ත්‍ර කේතයට පරිවර්තනය කරයි.

2) Program execution is relatively slow as analysis takes place every time the program runs.
වැඩසටහන ක්‍රියාත්මක වන සෑම අවස්ථාවකම විශ්ලේෂණය සිදු වන බැවින් වැඩසටහන් ක්‍රියාත්මක කිරීම සාපේක්ෂව මන්දගාමී වේ.

3) It reads the source codes and generates an executable file.
එය ප්‍රභව කේත කියවා ක්‍රියාත්මක කළ හැකි ගොනුවක් ජනනය කරයි.

4) Programs analyzed by interpreters are easier to debug.
අර්ථ විනිවිදීම විසින් විශ්ලේෂණය කරන ලද වැඩසටහන් දෝෂ නිරාකරණය කිරීමට පහසුය.

5) CPU utilization is relatively less.
CPU භාවිතය සාපේක්ෂව අඩුය.

Answer: 3)

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

Option / විකල්පය 1)
True. Interpreters work by reading and executing the source code line by line. This means that each line of code is converted into machine code and executed immediately before moving on to the next line.
සත්‍ය වේ. අර්ථ විනිවිදීම ක්‍රියා කරන්නේ ප්‍රභව කේත පේළියෙන් පේළියෙන් කියවා ක්‍රියාත්මක කිරීමෙනි. මෙයින් අදහස් කරන්නේ සෑම කේතයක්ම යන්ත්‍ර කේතයක් බවට පරිවර්තනය කර ඊළඟ පේළියට යාමට පෙර ක්‍රියාත්මක වන බවයි.

Option / විකල්පය 2)
True. Because interpreters translate and execute code one line at a time, they must perform analysis repeatedly each time the program is run. This results in slower execution compared to compiled programs, where the analysis is done only once during compilation.
සත්‍ය වේ. අර්ථ විනිවිදීම විසින් එක් පේළියකින් කේතයක් පරිවර්තනය කර ක්‍රියාත්මක කරන බැවින්, ඔවුන් වැඩසටහන ක්‍රියාත්මක වන සෑම අවස්ථාවකම නැවත නැවතත් විශ්ලේෂණය කළ යුතුය. සම්පාදනය කිරීමේදී එක් වරක් පමණක් විශ්ලේෂණය සිදු කරනු ලබන සම්පාදනය කරන ලද වැඩසටහන් හා සසඳන විට මෙහි ප්‍රතිඵලයක් ලෙස ක්‍රියාත්මක වීම මන්දගාමී වේ.

Option / විකල්පය 3)
False. Interpreters do not generate an executable file. They execute the source code directly without creating a separate machine code file. This contrasts with compilers, which convert the entire source code into an executable file that can be run independently of the source code.
අසත්‍ය වේ. අර්ථ විනිවිදීම ක්‍රියාත්මක කළ හැකි ගොනුවක් ජනනය නොකරයි. ඔවුන් වෙනම යන්ත්‍ර කේත ගොනුවක් සාදා නොගෙන කෙලින්ම ප්‍රභව කේතය ක්‍රියාත්මක කරයි. මෙය සම්පාදක සමඟ ප්‍රතිවිරුද්ධ වන අතර එමඟින් සම්පූර්ණ මූල කේතයම ක්‍රියාත්මක කළ හැකි ගොනුවක් බවට පරිවර්තනය කරයි, එය ප්‍රභව කේතයෙන් ස්වාධීනව ධාවනය කළ හැකිය.

Option / විකල්පය 4)
True. Since interpreters execute code line by line, it is easier to identify and correct errors as they occur. The immediate feedback provided by interpreters makes debugging more straightforward compared to compiled languages, where errors are reported after the entire program is compiled.
සත්‍ය වේ. අර්ථ විනිශ්චය විසින් පේළියෙන් කේත ක්‍රියාවට නංවන බැවින්, ඒවා සිදුවන විට දෝෂ හඳුනාගෙන ඒවා නිවැරදි කිරීම පහසුය. අර්ථ විනිශ්චය විසින් සපයනු ලබන ක්ෂණික ප්‍රතිපෝෂණය සම්පාදනය කරන ලද භාෂාවලට සාපේක්ෂව නිදොස්කරණය වඩාත් සරල කරයි, සම්පූර්ණ වැඩසටහන සම්පාදනය කිරීමෙන් පසු දෝෂ වාර්තා වේ.

Option / විකල්පය 5)
True. Interpreters often lead to less CPU utilization during program execution because they do not need to compile the entire program beforehand. However, this can vary depending on the specific program and the nature of the tasks it performs. The execution speed is generally slower, but the overhead of compiling the program is avoided.
සත්‍ය වේ. අර්ථ විනිශ්චය බොහෝ විට වැඩසටහන් ක්‍රියාත්මක කිරීමේදී අඩු CPU භාවිතයකට තුඩු දෙයි, මන්ද ඔවුන්ට සම්පූර්ණ වැඩසටහනම කලින් සම්පාදනය කිරීමට අවශ්‍ය නොවේ. කෙසේ වෙතත්, මෙය විශේෂිත වැඩසටහන සහ එය ඉටු කරන කාර්යයන් වල ස්වභාවය අනුව වෙනස් විය හැක. ක්‍රියාත්මක කිරීමේ වේගය සාමාන්‍යයෙන් මන්දගාමී වේ, නමුත් වැඩසටහන සම්පාදනය කිරීමේ කාර්ය භාරය මග හැරේ.

Therefore, the only statement that is false is Option 3) because interpreters do not generate an executable file; they execute code directly without creating a separate file.
එබැවින්, අසත්‍ය එකම ප්‍රකාශය වන්නේ විකල්ප 3) අර්ථ විනිශ්චය ක්‍රියාත්මක කළ හැකි ගොනුවක් ජනනය නොකරන බැවිනි. ඔවුන් වෙනම ගොනුවක් නිර්මාණය නොකර කෙලින්ම කේතය ක්‍රියාත්මක කරයි.

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

```
my_list = [1, 2, 3, 4]
for i in my_list:
    if i == 3:
        continue
    print(i, end=" ")
else:
    print("Loop completed!")
```

1) `1 2 Loop completed!`
2) `1 2 4 Loop completed!`
3) `1 2 3 4 Loop completed!`
4) `1 2 4`
5) Error!

Answer: 2)

Explanation / පැහැදිලි කිරීම:

Let's consider the explanation of the above python code one by one.
ඉහත පයිතන් කේත රේඛාවේ පැහැදිලි කිරීම එකින් එක සලකා බලමු.

`my_list = [1, 2, 3, 4]`
This creates a list with the elements `[1, 2, 3, 4]`.
මෙය මූලාංග `[1, 2, 3, 4]` සමඟ ලැයිස්තුවක් නිර්මාණය කරයි.

`for i in my_list`
This loop iterates over each element in `my_list`. The variable `i` takes each value in the list, one by one.
මෙම ලූපය `my_list` හි එක් එක් මූලාංග හරහා පුනරාවර්තනය වේ. `i` විචල්‍යය ලැයිස්තුවේ එක් එක් අගය, එකින් එක ගනී.

`if i == 3: continue`
This condition checks if the value of `i` is 3. If it is, the continue statement is executed, which skips the rest of the code in the loop for that iteration and moves to the next iteration.
මෙම කොන්දේසිය `i` හි අගය 3 වේ දැයි පරීක්ෂා කරයි. එය එසේ නම්, දිගටම ප්‍රකාශය ක්‍රියාත්මක වේ, එම පුනරාවර්තනය සඳහා ලූපයේ ඉතිරි කේතය මඟ හැර ඊළඟ පුනරාවර්තනය වෙත ගමන් කරයි.

`print(i, end=" ")`
If the value of `i` is not 3, this line prints the value of i followed by a space (instead of a newline).
`i` හි අගය 3 නොවේ නම්, මෙම පේළිය `i` හි අගය පසුව හිස්තැනක් (නව රේඛාවක් වෙනුවට) මුද්‍රණය කරයි.

`else: print("Loop completed!")`
After the loop completes all iterations, the else block is executed, and "Loop completed!" is printed.
ලූපය සියලු පුනරාවර්තන සම්පූර්ණ කළ පසු, else block එක ක්‍රියාත්මක වන අතර, "Loop completed!" මුද්‍රණය කර ඇත.

Output
The loop will print 1, 2 and 4 (because it skips printing 3).
ලූපය 1, 2 සහ 4 මුද්‍රණය කරයි (එය 3 මුද්‍රණය මඟහරින නිසා).

After the loop finishes, it prints `Loop completed!`.
ලූපය අවසන් වූ පසු, එය මුද්‍රණය කරන්නේ `Loop completed!`.

So, the correct answer is 2).
ඒ අනුව නිවැරදි පිළිතුර වන්නේ 2).