සියලු ම හිමිකම් ඇවිරිණි / *All Rights Reserved*]

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

Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 224 MARKING

2025/07/22
ictfromabc
Ravindu Bandaranayake

1. The statement that best describes the result of fragmentation on a hard disk is,
දෘඩ තැටිය තුළ බණ්ඩනිකරණය වීම නිසා ඇතිවන ප්‍රතිඵලය වඩාත් හොඳින් විස්තර කෙරෙන ප්‍රකාශය වන්නේ,

   1) Some files are deleted from the hard drive.
   සමහර ගොනු දෘඩ තැටියෙන් මැකී යයි.

   2) The data access speed of the hard disk is reduced.
   දෘඩ තැටියේ දත්ත ප්‍රවේශ වේගය අඩු වේ.

   3) The hard disk is not distributed on a local area network.
   ස්ථානීය ප්‍රදේශ ජාලයක දෘඩ තැටිය බෙදා හැරිය නොහැකි වේ.

   4) The number of bad sectors increases.
   භාවිතයට ගත නොහැකි අංග bad sectors සංඛ්‍යාව වැඩි වෙයි.

   5) An existing file is removed and another file is placed.
   පවතින ගොනුවක් ඉවත් වී වෙනත් ගොනුවක් තැන්පත් වේ.

Answer: 2)

Explanation / පැහැදිලි කිරීම:

Fragmentation occurs when files are broken up into pieces and stored in different parts of the hard disk. This happens over time as files are created, modified, and deleted. As a result, the hard disk has to work harder to read these scattered file fragments, which reduces the data access speed.
බණ්ඩනිකරණය සිදුවන්නේ ගොනු කොටස්වලට කැඩී දෘඩ තැටියේ විවිධ කොටස්වල ගබඩා කළ විටය. ගොනු සෑදීම, වෙනස් කිරීම සහ මකා දැමීම නිසා කාලයත් සමඟ මෙය සිදු වේ. එහි ප්‍රතිඵලයක් ලෙස, දත්ත ප්‍රවේශ වේගය අඩු කරන අතර මෙම විසිරුණු ගොනු කොටස් කියවීමට දෘඩ තැටියට වැඩි වෙහෙසක් දැරීමට සිදුවේ.

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

Option / විකල්පය 1)
Fragmentation does not involve deleting files, it just affects how files are stored on the disk.
බණ්ඩනිකරණයට ගොනු මැකී යාම ඇතුළත් නොවේ, එය තැටියේ ගොනු ගබඩා කරන ආකාරයට බලපායි.

Option / විකල්පය 2)
Correct. Fragmentation causes slower access because the disk's read/write head needs to move to multiple locations to read fragmented files.
නිවැරදියි. බණ්ඩනිකරණය වීම මන්දගාමී ප්‍රවේශයක් ඇති කරයි, මන්ද තැටියේ කියවීමේ/ලිවීමේ හිස බණ්ඩනය වූ ගොනු කියවීමට ස්ථාන කිහිපයකට ගෙන යාමට අවශ්‍ය වේ.

Option / විකල්පය 3)
This statement is unrelated to fragmentation, it is about network distribution.
මෙම ප්‍රකාශය බණ්ඩනිකරණයට සම්බන්ධ නැත, එය ජාල සම්ප්‍රේෂණය ගැන ය.

Option / විකල්පය 4)
Fragmentation does not cause bad sectors. Bad sectors are physical damage on the disk surface, while fragmentation is a file storage issue.
බණ්ඩනිකරණය bad sectors ඇති නොකරයි. bad sectors යනු තැටියේ මතුපිට භෞතිකව හානි වීම වන අතර, බණ්ඩනිකරණය යනු ගොනු ගබඩා කිරීමේ ගැටලුවකි.

Option / විකල්පය 5)
Fragmentation does not mean files are removed or replaced. It only affects how data is stored, not whether files are deleted or overwritten.
ඛණ්ඩනිකරණය යනු ගොනු ඉවත් කිරීම හෝ ප්‍රතිස්ථාපනය කිරීම නොවේ. එය බලපාන්නේ දත්ත ගබඩා කරන්නේ කෙසේද යන්නට මිස ගොනු මකා දැමීම හෝ එක මත එක ලිවීම යන්න නොවේ.

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

2. Which company invented the world's first microprocessor?
ලොව ප්‍රථම ක්ෂුද්‍ර සකසනය නිර්මාණය කළ ආයතනය කුමක්ද?

1) IBM 2) Intel 3) AMD 4) Honeywell 5) Apple

Answer: 2)

Explanation / පැහැදිලි කිරීම:

The Intel 4004 was the first commercially available microprocessor, which integrated the functions of a CPU (Central Processing Unit) onto a single chip. It could perform a variety of tasks by executing a series of instructions, making it programmable.
Intel 4004 යනු වාණිජමය වශයෙන් ලබා ගත හැකිවූ පළමු ක්ෂුද්‍ර සකසනය වන අතර එය CPU (මධ්‍යම සැකසුම් ඒකකය) හි ක්‍රියාකාරිත්වය තනි විපයකට ඒකාබද්ධ කරන ලදී. එයට ක්‍රමලේඛගත කළ හැකි උපදෙස් මාලාවක් ක්‍රියාත්මක කිරීමෙන් විවිධ කාර්යයන් ඉටු කළ හැකිය.

Release Date / නිකුත් වූ දිනය: November 15, 1971
Clock Speed / වේගය: 740 kHz
Transistors / ට්‍රාන්සිස්ටර: 2,300
Bus Width / බස් පළල: 4-bit
Package / පැකේජය: 16-pin DIP

The creation of the microprocessor revolutionized the electronics industry, leading to the development of personal computers, smartphones, and countless other digital devices.
ක්ෂුද්‍ර සකසනය නිර්මාණය කිරීම ඉලෙක්ට්‍රොනික කර්මාන්තයේ විප්ලවීය වෙනසක් ඇති කළ අතර එය පුද්ගලික පරිගණක, ස්මාර්ට්ෆෝන් සහ නවත් ගණන් කළ නොහැකි ඩිජිටල් උපාංග සංවර්ධනය කිරීමට හේතු විය.

Intel followed up with more powerful microprocessors, such as the 8008, 8080, and the iconic 8086, which laid the groundwork for the x86 architecture that still dominates the PC market today.
Intel විසින් 8008, 8080 සහ iconic 8086 වැනි වඩාත් ප්‍රබල ක්ෂුද්‍ර සකසන සමඟින් පර්යේෂණ සිදු කරන ලද අතර, එය අදටත් පරිගණක වෙළඳපොලේ ආධිපත්‍යය දරන x86 ආකෘති ශිල්පය සඳහා අඩිතාලම දැමීය.

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

3. X = 2's complement of 120
$Y = 176_8$

Select the answer that contains the values obtained when X and Y are subjected to bitwise AND and XOR operations respectively.
X හා Y බිටුඅනුසාරිත AND සහ XOR මෙහෙයුම්වලට ලක් කිරීමේදී ලැබෙන අගයන් පිළිවෙලින් දැක්වෙන පිළිතුර තෝරන්න.

1) $01111000_2, 00000110_2$ 2) $11111110_2, 01111110_2$ 3) $00001000_2, 11110111_2$
4) $11111110_2, 11110110_2$ 5) $11111110_2, 00001001_2$

Answer: 1)

Explanation / පැහැදිලි කිරීම:

First, the corresponding binary values for X and Y should be obtained.
ප්‍රථමයෙන් X හා Y සඳහා අදාල ද්විමය අගයන් ලබා ගත යුතුය.

X
The two's complement of any positive number is the binary representation of that number. So the number should be converted to binary.
ඕනෑම ධන සංඛ්‍යාවක දෙකෙහි අනුපූරකය එම සංඛ්‍යාවේ ද්විමය නිරූපණය වේ. එබැවින් එම සංඛ්‍යාව ද්විමය ආකාරයට පරිවර්තනය කල යුතුය.

120 into binary							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	1	1	1	1	0	0	0

Y -

176_8 into binary								
1			7			6		
4	2	1	4	2	1	4	2	1
0	0	1	1	1	1	1	1	0

The two obtained binary numbers are subjected to the bitwise XOR and AND operations. In XOR, 1 is output only if the two input bits are different. The output is 0 when there are identical input bits. In AND, 1 is output only if all inputs are 1.
ලබාගත් ද්විමය සංඛ්‍යා දෙක බිටුඅනුසාරිත XOR සහ AND කර්මයට ලක් කල යුතු වේ. XOR හිදී ආදාන බිටු දෙක එකිනෙකට වෙනස් නම් පමණක් 1 ප්‍රතිදානය කරයි. එක සමාන ආදාන බිටු පවතින විට ප්‍රතිදානය 0 වෙයි. AND හිදී ආදාන සියල්ල සමාන නම් පමණක් 1 ප්‍රතිදානය කරයි.

Bitwise XOR							
0	1	1	1	1	0	0	0
0	1	1	1	1	1	1	0
0	0	0	0	0	1	1	0

Bitwise AND							
0	1	1	1	1	0	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	0	0	0

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

4. What is the two's complement of -11?
-11 හි දෙකෙහි අනුපූරකය කුමක් ද?

1) 00001011 2) 11110100 3) <u>11110101</u> 4) 11111101 5) 00001100

Answer: 3)

Explanation / පැහැදිලි කිරීම:

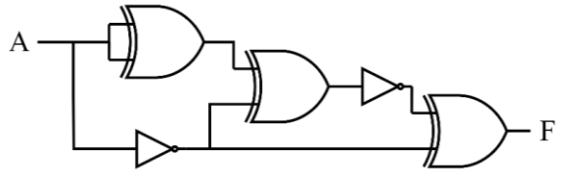
First, let's convert this number to binary.
පළමුව, මෙම සංඛ්‍යාව ද්විමය බවට පරිවර්ථනය කරගනිමු.

11_{10}							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	1	0	1	1
00001011_2							

2's Complement of -11							
0	0	0	0	1	0	1	1
Invert the Bits							
1	1	1	1	0	1	0	0
						+	1
1	1	1	1	0	1	0	1

Therefore, the correct answer, according to the above explanation, is 3).
එබැවින්, ඉහත පැහැදිලි කිරීම අනුව නිවැරදි පිළිතුර 3) වේ.

5. What is the output of F in the following logic circuit?
පහත තාර්කික පරිපථයේ F හි ප්‍රතිදානය කුමක්ද?

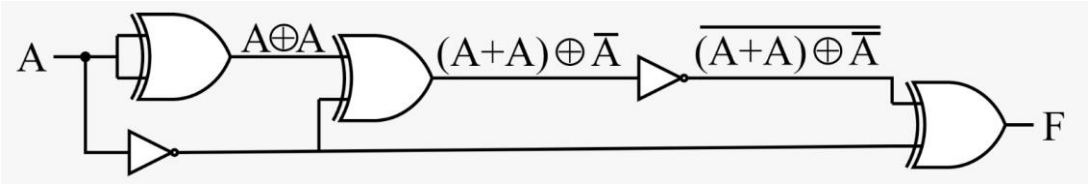


1) A 2) $\overline{A}$ 3) 0 4) $\underline{1}$ 5) Can't say for sure.
නිශ්චිතව ප්‍රකාශ කළ නොහැක.

Answer: 4)

Explanation / පැහැදිලි කිරීම:

Let us first derive the expression for the output F of this logic circuit.
පළමුව මෙම තාර්කික පරිපථයේ F ප්‍රතිදානයට අදාළ ප්‍රකාශනය ලබාගනිමු.



$$F = ((\overline{A \oplus A) \oplus \overline{A}}) \oplus \overline{A}$$

We simplify this expression using Boolean laws.
මෙම ප්‍රකාශනය බූලියානු නීති භාවිතයෙන් සුළු කරමු.

$= ((\overline{A \oplus A) \oplus \overline{A}}) \oplus \overline{A}$	
$= ((\overline{\overline{A}A + A\overline{A}}) \oplus \overline{A}) \oplus \overline{A}$	$A \oplus B = \overline{A}B + A\overline{B}$
$= ((\overline{0 + 0}) \oplus \overline{A}) \oplus \overline{A}$	Inverse law / **ප්‍රතිලෝම න්‍යාය**
$= (\overline{0 \oplus \overline{A}}) \oplus \overline{A}$	Identity law / **සර්වසාම්‍ය න්‍යාය**
$= (\overline{\overline{0}\overline{A} + 0\overline{\overline{A}}}) \oplus \overline{A}$	$A \oplus B = \overline{A}B + A\overline{B}$
$= (\overline{\overline{0}\overline{A} + 0}) \oplus \overline{A}$	Identity law / **සර්වසාම්‍ය න්‍යාය**
$= \overline{\overline{0}\overline{A}} \oplus \overline{A}$	Identity law / **සර්වසාම්‍ය න්‍යාය**
$= \overline{1\overline{A}} \oplus \overline{A}$	$\overline{0} = 1$
$= \overline{\overline{A}} \oplus \overline{A}$	Identity law / **සර්වසාම්‍ය න්‍යාය**
$= A \oplus \overline{A}$	Double inverse law / **ද්විත්ව ප්‍රතිලෝම න්‍යාය**
$= \overline{A}\,\overline{A} + A\,\overline{\overline{A}}$	$A \oplus B = \overline{A}B + A\overline{B}$
$= \overline{A} + A\overline{\overline{A}}$	Idempotent law / **ප්‍රතිලෝම න්‍යාය**
$= \overline{A} + AA$	Double inverse law / **ද්විත්ව ප්‍රතිලෝම න්‍යාය**
$= \overline{A} + A$	Idempotent law / **තද්ධවභාවී න්‍යාය**
$= 1$	Inverse law / **ප්‍රතිලෝම න්‍යාය**

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

6. What is the answer when the following Boolean expression is trivialized?
පහත බූලිය ප්‍රකාශනය සුළුකල විට පිළිතුර වන්නේ කුමක්ද?

$\overline{N} + \overline{A\overline{S}A} + \overline{\overline{A}}$

1) $\overline{N}$ 2) $\overline{N} + \overline{A} + \overline{\overline{A}}$ 3) $\underline{1}$ 4) $\overline{N} + \overline{A}$ 5) $\overline{A} + \overline{\overline{A}}$

Answer: 3)

Explanation / පැහැදිලි කිරීම:

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

$\overline{N} + \overline{A\overline{S}A} + A$
$\overline{N} + \overline{A} + \overline{\overline{S}} + \overline{A} + A$ De Morgan's Law / ද මුවාර් ප්‍රමේයය
$\overline{N} + \overline{A} + S + \overline{A} + A$ Double inverse law / ද්විත්ව ප්‍රතිලෝම න්‍යාය
$\overline{N} + S + \overline{A} + 1$ Inverse law / ප්‍රතිලෝම න්‍යාය
1 Identity law / සර්වසාම්‍ය න්‍යාය

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

7. Which of the following circuits is used as a "Memory device" in Computers?
පරිගණකයේ "මතක උපාංගයක්" ලෙස භාවිතා කරන්නේ පහත සඳහන් පරිපථවලින් කවරක් ද?

1) Rectifier 2) Flip Flop 3) Comparator 4) Attenuator 5) None of the above.
ඉහත කිසිවක් නොවේ.

Answer: 2)

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

1) Rectifier
This statement is incorrect. A rectifier converts alternating current (AC) to direct current (DC) and is not used for memory storage in computers.
මෙම ප්‍රකාශය වැරදි වේ. සෘජුකාරකයක් (AC) ප්‍රත්‍යාවර්ත ධාරාව (DC)සෘජු ධාරාවක් බවට පරිවර්තනය කරන අතර පරිගණකවල මතක ගබඩා කිරීම සඳහා භාවිතා නොවේ.

2) Flip Flop
This statement is correct. A flip-flop is a basic memory device in computers, often used in digital electronics to store
binary data (0 or 1). It can maintain a stable state and is a fundamental building block of memory circuits.
මෙම ප්‍රකාශය නිවැරදි වේ. flip-flop යනු පරිගණකවල මූලික මතක උපාංගයකි, බොහෝ විට ද්විමය දත්ත 0 හෝ 1 ගබඩා කිරීම සඳහා ඩිජිටල් ඉලෙක්ට්‍රොනික උපකරණවල භාවිතා වේ. එය ස්ථාවර තත්වයක් පවත්වා ගත හැකි අතර මතක පරිපථවල මූලික ගොඩනැඟිලි කොටසකි.

3) Comparator
This statement is incorrect. A comparator is a device that compares two voltages or currents and outputs a binary signal based on the comparison. It is not used for memory storage.
මෙම ප්‍රකාශය වැරදි වේ. comparator යනු වෝල්ටියතා හෝ ධාරා දෙකක් සංසන්දනය කර සංසන්දනය මත පදනම්ව ද්විමය සංඥාවක් නිකුත් කරන උපකරණයකි. එය මතක ගබඩා කිරීම සඳහා භාවිතා නොවේ.

4) Attenuator
This statement is incorrect. An attenuator reduces the power of a signal without significantly distorting its waveform. It is used for signal control, not memory storage.
මෙම ප්‍රකාශය වැරදි වේ. attenuator එහි තරංග ආකාරය සැලකිය යුතු ලෙස විකෘති නොකර සංඥාවේ බලය අඩු කරයි. එය භාවිතා කරන්නේ සංඥා පාලනය සඳහා මිස මතක ගබඩා කිරීම සඳහා නොවේ.

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

8. What is the difference between a list and a tuple in Python?
Python හි ලැයිස්තුවක් සහ tuple එකක් අතර වෙනස කුමක්ද?

1) Lists are mutable (changeable), tuples are immutable (unchangeable)
ලැයිස්තු වෙනස් කළ හැකි අතර tuple වෙනස් කළ නොහැකි වේ.
2) Tuples can store different data types, lists can only store one type.
Tuple හි විවිධ දත්ත වර්ග ගබඩා කළ හැකි අතර ලැයිස්තුවල ගබඩා කළ හැක්කේ එක් වර්ගයක් පමණි.
3) Lists are used for storing functions, tuples are for data.
ලැයිස්තු ශ්‍රිත ගබඩා කිරීම සඳහා භාවිතා වන අතර tuple දත්ත සඳහා වේ.
4) Tuples are faster than lists.
Tuple ලැයිස්තු වලට වඩා වේගවත්ය.
5) There is no significant difference.
සැලකිය යුතු වෙනසක් නොමැත.

Answer: 1)

Let's consider each statement in the answers.
පිළිතුරු වල ඇති එක් එක් ප්‍රකාශ සලකා බලමු.

Statement / ප්‍රකාශය 1)
In Python, lists and tuples are both used to store collections of items. However, they have a key difference in terms of mutability:
Python හි, list සහ tuple යන දෙකම අයිතම එකතු කිරීම් ගබඩා කිරීමට භාවිතා කරයි. කෙසේ වෙතත්, විකෘතිතාවය අනුව මේවාට ප්‍රධාන වෙනසක් ඇත.

Lists are mutable, which means you can change their content after they are created. You can add, remove, or modify items in a list.
List වෙනස් කළ හැකි ය, එයින් අදහස් වන්නේ ඒවා නිර්මාණය කිරීමෙන් පසු ඒවායේ අන්තර්ගතය වෙනස් කළ හැකි බවයි. list එකක අයිතම එකතු කිරීමට, ඉවත් කිරීමට හෝ වෙනස් කිරීමට හැකිය.

Tuples are immutable, which means once they are created, their content cannot be changed. You cannot add, remove, or modify items in a tuple.
Tuples වෙනස් කළ නොහැකි ය, එනම් ඒවා නිර්මාණය කළ පසු ඒවායේ අන්තර්ගතය වෙනස් කළ නොහැක. Tuple එකක අයිතම එකතු කිරීමට, ඉවත් කිරීමට හෝ වෙනස් කිරීමට නොහැක.

Statement / ප්‍රකාශය 2)
Both lists and tuples can store different data types. There is no restriction that a list can only store one type of data.
List සහ tuple යන දෙකටම විවිධ දත්ත වර්ග ගබඩා කළ හැක. list එකකට එක් දත්ත වර්ගයක් පමණක් ගබඩා කළ හැකි බවට සීමාවක් නොමැත.

Statement / ප්‍රකාශය 3)
Both lists and tuples can store functions as well as other types of data. This is not a distinguishing feature between them.
List සහ tuple යන දෙකටම කාර්යයන් මෙන්ම අනෙකුත් දත්ත වර්ග ගබඩා කළ හැක. මෙය ඔවුන් අතර කැපී පෙනෙන ලක්ෂණයක් නොවේ.

Statement / ප්‍රකාශය 4)
While tuples can be faster than lists due to their immutability, this is not the primary or most significant difference between them. The main difference lies in their mutability.
Tuple ඒවායේ වෙනස් නොවන බව නිසා list වලට වඩා වේගවත් විය හැකි අතර, මෙය ඔවුන් අතර ඇති ප්‍රාථමික හෝ වඩාත්ම වැදගත් වෙනස නොවේ. ප්‍රධාන වෙනස පවතින්නේ ඔවුන්ගේ විකෘතිතාවයයි.

Statement / ප්‍රකාශය 5)
There is a significant difference between lists and tuples: lists are mutable, and tuples are immutable.
List සහ tuple අතර සැලකිය යුතු වෙනසක් ඇත. list විකෘති වන අතර tuple වෙනස් කළ නොහැකි ය.

Therefore, the correct answer, according to the above explanation, is 1).
එබැවින්, ඉහත පැහැදිලි කිරීම අනුව නිවැරදි පිළිතුර 1) වේ.

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

```
numbers = [10, 20, 30, 40, 50]
print(numbers)
numbers.append(60)
print(numbers)
numbers.insert(7, 70)
print(numbers)
```

1)
```
[10, 20, 30, 40, 50]
[10, 20, 30, 40, 50, 60]
[10, 20, 30, 40, 50, 60, 7]
```

2)
```
(10, 20, 30, 40, 50)
(10, 20, 30, 40, 50, 60)
(10, 20, 30, 40, 50, 60, 7)
```

3)
```
[10, 20, 30, 40, 50]
[10, 20, 30, 40, 50, 60]
[10, 20, 30, 40, 50, 70]
```

4)
```
[10, 20, 30, 40, 50]
[10, 20, 30, 40, 50, 60]
[10, 20, 30, 40, 50, 60, 70]
```

5)
```
(10, 20, 30, 40, 50)
(10, 20, 30, 40, 50, 60)
(10, 20, 30, 40, 50, 60, 70)
```

Answer: 4)

Explanation / පැහැදිලි කිරීම:

The explanation of the output from the code is as follows.
කේතයෙන් ලැබෙන ප්‍රතිදානය පිළිබඳව පැහැදිලි කිරීම පහත පරිදි වේ.

A list named `numbers` is created with the values `[10, 20, 30, 40, 50]`. The list is printed, so the output is `[10, 20, 30, 40, 50]`.
`[10, 20, 30, 40, 50]` අගයන් සමඟ `numbers` නම් ලැයිස්තුවක් සාදනු ලැබේ. ලැයිස්තුව මුද්‍රණය කර ඇත, එබැවින් ප්‍රතිදානය `[10, 20, 30, 40, 50]` වේ.

2. The method `append(60)`, this method adds the number `60` to the end of the list. After appending, the list becomes `[10, 20, 30, 40, 50, 60]`. The updated list is printed, so the output is `[10, 20, 30, 40, 50, 60]`.
`append(60)` ක්‍රමය, මෙම ක්‍රමය ලැයිස්තුවේ අවසානයට අංක `60` එකතු කරයි. එම එක් කිරීමෙන් පසු, ලැයිස්තුව `[10, 20, 30, 40, 50, 60]`. බවට පත් වේ, එබැවින් ප්‍රතිදානය `[10, 20, 30, 40, 50, 60]` වේ.

3. The method `insert(7, 70)`, this method tries to insert the number `70` at index `7`. Since the current list length is only 6 (0 to 5 index), Python adds `70` to the end of the list. The list becomes `[10, 20, 30, 40, 50, 60, 70]`. The final list is printed, so the output is `[10, 20, 30, 40, 50, 60, 70]`.
`insert(7, 70)` ක්‍රමය, මෙම ක්‍රමය `70` index number `7`ට ඇතුළත් කිරීමට උත්සාහ කරයි. වත්මන් ලැයිස්තුවේ දිග 6ක් පමණක් වන බැවින් (`0` සිට `5` දක්වා index), Python ලැයිස්තුවේ අවසානයට `70` එකතු කරයි. ලැයිස්තුව `[10, 20, 30, 40, 50, 60, 70]` බවට පත් වේ. අවසාන ලැයිස්තුව මුද්‍රණය කර ඇත, එබැවින් ප්‍රතිදානය `[10, 20, 30, 40, 50, 60, 70]` වේ.

So, the code results in three print outputs,
එබැවින්, කේතය මුද්‍රණ ප්‍රතිදාන තුනක් ඇති කරයි,

```
[10, 20, 30, 40, 50]
[10, 20, 30, 40, 50, 60]
[10, 20, 30, 40, 50, 60, 70]
```

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

```
print("Hello, World!"[-1])
```

1) `Hello, World!` 2) `!` 3) `d` 4) `Error` 5) None of these.

Answer: 2)

The correct answer is 2) `"!"`.
නිවැරදි පිළිතුර 2) `"!"` වේ.

Explanation / පැහැදිලි කිරීම:

The output of the Python code `print ("Hello, World!"[-1])` is `"!"`. This is because in Python, negative indexing starts from the end of the string. The index of `-1` refers to the last character of the string, which in this case is `"!"`.
`print ("Hello, World!"[-1])` Python කේතයේ ප්‍රතිදානය `"!"` වේ. මෙයට හේතුව Python හි සෘණ string සුචිගත කිරීම ආරම්භ වන්නේ අග සිට වීමයි. `-1` හි දර්ශකය තන්තුවෙහි අවසාන අක්ෂරයට යොමු කරයි, මෙම අවස්ථාවෙහිදී එය `"!"` වේ.

In Python, strings are indexed with the first character having index `0`. However, Python also supports negative indexing, with `-1` referring to the last character, `-2` referring to the second last character, and so on. Therefore, when we use `-1` as the index, Python returns the last character of the string.
Python හි, strings සුචිගත කර ඇත්තේ `0` වන දර්ශකය සහිත පළමු අක්ෂරය සමඟිනි. කෙසේ වෙතත්, strings සෘණ සුචිකරණයට සහය දක්වයි, `-1` අවසාන අක්ෂරයට යොමු කිරීම, `-2` දෙවන අවසාන අක්ෂරය වෙත යොමු කිරීම සහ යනාදිය. එබැවින්, අපි `-1` දර්ශකය ලෙස භාවිතා කරන විට, Python strings අවසාන අක්ෂරය ලබා දෙයි.

In the given string `"Hello, World!"`, the last character is `"!"`. So, that is what the code prints.
ලබා දී ඇති `"Hello, World!"` string එකෙහි අවසාන අක්ෂරය වන්නේ `"!"`. එබැවින්, කේතය මුද්‍රණය කරන්නේ එයයි.