සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved]

Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 268 MARKING

2025/09/04
ictfromabc
Ravindu Bandaranayake

1. Which of the Which of the following statements correctly distinguishes Second Generation computers (1956–1963) from Third Generation computers (1964–1971)?
පහත සඳහන් ප්‍රකාශ අතුරෙන් කුමන ප්‍රකාශය දෙවන පරම්පරාවේ පරිගණක (1956 -1963) තුන්වන පරම්පරාවේ පරිගණක (1964 -1971) වලින් නිවැරදිව වෙන්කර හඳුනා ගනිද?

   1) Second Generation used integrated circuits, while Third Generation used transistors.
   දෙවන පරම්පරාව අනුකලිත පරිපථ භාවිතා කළ අතර, තෙවන පරම්පරාව ට්‍රාන්සිස්ටර භාවිතා කළේය.
   2) Second Generation computers were smaller, faster, and more reliable than Third Generation computers.
   දෙවන පරම්පරාවේ පරිගණක තුන්වන පරම්පරාවේ පරිගණකවලට වඩා කුඩා, වේගවත් සහ විශ්වාසදායක විය.
   3) Second Generation computers primarily used assembly language and machine code, whereas Third Generation computers introduced high-level languages like COBOL and FORTRAN more extensively.
   දෙවන පරම්පරාවේ පරිගණක ප්‍රධාන වශයෙන් එසෙම්බ්ලි භාෂාව සහ යන්ත්‍ර කේත භාවිතා කළ අතර, තෙවන පරම්පරාවේ පරිගණක COBOL සහ FORTRAN වැනි ඉහළ මට්ටමේ භාෂා වඩාත් පුළුල් ලෙස හඳුන්වා දුන්නේය.
   4) Second Generation computers supported multiprogramming, but Third Generation computers did not.
   දෙවන පරම්පරාවේ පරිගණක බහු ක්‍රමලේඛනකරණයට සහය දැක්වූ නමුත් තුන්වන පරම්පරාවේ පරිගණක එසේ කළේ නැත.
   5) Both Second and Third Generation computers used vacuum tubes, but with different configurations.
   දෙවන සහ තුන්වන පරම්පරාවේ පරිගණක දෙකම රික්තක නල භාවිතා කළ නමුත් විවිධ වින්‍යාසයන් සහිතව.

   Answer : 3)

   Explanation

   - Option 1 – Incorrect: Transistors were used in Second Generation, while Integrated Circuits (ICs) were introduced in Third Generation.
   විකල 1 -වැරදියි: දෙවන පරම්පරාවේදී ට්‍රාන්සිස්ටර භාවිතා කරන ලද අතර, තුන්වන පරම්පරාවේදී අනුකලිත පරිපථ (ICs) හඳුන්වා දෙන ලදී.

   - Option 2 – Incorrect: The opposite is true. Third Generation machines were smaller, faster, more reliable, and consumed less power.
   විකල්ප 2 - වැරදියි: ප්‍රතිවිරුද්ධ දෙය සත්‍ය වේ. තුන්වන පරම්පරාවේ යන්ත්‍ර කුඩා, වේගවත්, වඩා විශ්වාසදායක වූ අතර අඩු බලයක් පරිභෝජනය කළේය.

   - Option 3 – Correct: Programming in Second Generation was mostly in assembly and machine code. Third Generation expanded the use of high-level languages like COBOL, FORTRAN, BASIC, making programming more efficient.
   විකල්ප 3 - නිවැරදි: දෙවන පරම්පරාවේ ක්‍රමලේඛනය බොහෝ දුරට එසෙම්බ්ලි සහ යන්ත්‍ර කේතයෙන් සිදු විය. තෙවන පරම්පරාව COBOL, FORTRAN, BASIC වැනි ඉහළ මට්ටමේ භාෂා භාවිතය පුළුල් කළ අතර එමඟින් ක්‍රමලේඛනය වඩාත් කාර්යක්ෂම විය.

- Option 4 – Incorrect: Multiprogramming and time-sharing were features of Third Generation, not the Second.
  විකල්ප 4 -වැරදියි: බහු ක්‍රමලේඛනය සහ කාලය බෙදා ගැනීම දෙවන පරම්පරාවේ නොව තුන්වන පරම්පරාවේ ලක්ෂණ විය.

- Option 5 – Incorrect: Vacuum tubes were part of First Generation only; not used in Second or Third.
  විකල්ප 5 - වැරදියි: රික්තක නල පළමු පරම්පරාවේ කොටසක් පමණක් විය දෙවන හෝ තෙවන පරම්පරාවේ භාවිතා නොකළේය.

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

2. Which of the following components is directly connected to the motherboard through slots or sockets?
   පහත සඳහන් සංරචක අතරින් තවි හෝ සොකට් හරහා මවු පුවරුවට සෘජුවම සම්බන්ධ වන්නේ කුමන සංරචකද?

   1) Monitor
   2) Printer
   3) CPU
   4) External Hard Drive
   5) Keyboard

   Answer : 3)

   Explanation

   The motherboard is the main circuit board that houses essential components like the CPU (Central Processing Unit), RAM (memory), and expansion cards through sockets and slots
   මවු පුවරුව යනු CPU (මධ්‍යම සැකසුම් ඒකකය), RAM (මතකය) සහ සොකට් සහ තවි හරහා පුළුල් කිරීමේ කාඩ්පත් වැනි අත්‍යවශ්‍ය සංරචක අඩංගු ප්‍රධාන පරිපථ පුවරුවයි

   While monitors, printers, external drives, and keyboards connect to the computer via ports or cables, they are peripheral devices and not directly connected to the motherboard's sockets in the same way the CPU is.
   මොනිටර, මුද්‍රණ යන්ත්‍ර, බාහිර ධාවක සහ යතුරුපුවරු පරිගණකයට කෙවෙනි හෝ කේබල් හරහා සම්බන්ධ වුවද, ඒවා පර්යන්ත උපාංග වන අතර CPU එක මෙන් මවු පුවරුවේ සොකට් වලට කෙලින්ම සම්බන්ධ නොවේ.

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

3. Convert the binary number 1011.101_2 into its decimal, hexadecimal, and octal equivalents. Which of the following options correctly represents all three conversions?
   ද්විමය සංඛ්‍යාව 1011.101_2 එහි දශම, ෂඩ් දශම සහ අෂ්ටමය සමාන බවට පරිවර්තනය කරන්න. පහත සඳහන් විකල්ප අතුරින් පරිවර්තන තුනම නිවැරදිව නියෝජනය කරන්නේ කුමක්ද?

   1) $1011.101_2 = 11.75 = B.C_{16} = 13.6_8$
   2) $1011.101_2 = 11.625 = B.A_{16} = 13.5_8$
   3) $1011.101_2 = 10.625 = A.A_{16} = 12.5_8$
   4) $1011.101_2 = 11.375 = B.6_{16} = 13.3_8$
   5) $1011.101_2 = 11.625 = B.9_{16} = 13.7_8$

   Answer : 2)

   Explanation

   Step 1: Binary to Decimal Conversion
   පියවර 1: ද්විමය සිට දශම දක්වා පරිවර්තනය

Binary	1	0	1	1	1	0	1
Decimal	8	4	2	1	0.5	0.25	0.125
	8+2+1+0.5+0.125 = 11.625						

Step 2 :Binary to Hexadecimal Conversion
පියවර 2: ද්විමය සිට ෂඩ් දශම දක්වා පරිවර්තනය

Binary	1011.101_2							
	1011				1010			
Hexa	8	4	2	1	8	4	2	1
	8+2+1 =B				8+2 = A			
	$B.A_{16}$							

Step 3 :Binary to Octal Conversion
පියවර 3: ද්විමය සිට අෂ්ටමය දක්වා පරිවර්තනය

Binary	1011.101_2								
	001			011			101		
Octal	4	2	1	4	2	1	4	2	1
	1			2+1 =3			4+1=5		
	13.5_8								

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

4. In an 8-bit 2's complement system, what is the range of decimal values that can be represented?
බිටු 8හි දෙකෙහි අනුපූරක පද්ධතියක, නිරූපණය කළ හැකි දශමය අගයන් පරාසය කුමක්ද?

1) -128 to +128 2) -127 to +127 3) <u>-128 to +127</u>
4) -127 to +128 5) -256 to +255

Answer: 3)

In an 8-bit 2's complement system, the range of decimal values is determined by the binary representation, where the most significant bit (MSB) serves as the sign bit.
බිටු 8හි 2හි අනුපූරක පද්ධතියක, දශමය අගයන් පරාසය ද්විමය නිරූපණය මගින් තීරණය කරනු ලැබේ, එහිදී වැඩිම වෙසෙසි බිටුව (MSB) ලකුණු බිටුව ලෙස ක්‍රියා කරයි.

0 for positive numbers (including zero) / **ධන සංඛ්‍යා සඳහා** 0 **(ශූන්‍යය ඇතුළුව).**
1 for negative numbers / **සෘණ අංක සඳහා** 1.

Positive range / ධන පරාසය
The smallest positive value is 0 (all bits 0).
කුඩාම ධන අගය 0 **(සියලු බිටු** 0**) වේ.**

The largest positive value is when the sign bit is 0 and all other bits are 1: 01111111
විශාලතම ධන අගය වන්නේ ලකුණු බිටුව 0 **වන අතර අනෙකුත් සියලුම බිටු** 1 **වන විටය:** 01111111

Decimal equivalent: $2^7-1 = 128-1 = 127$
දශමය අගය: $2^7-1 = 128-1 = 127$

Negative range / සෘණ පරාසය
The smallest negative value is when the sign bit is 1 and all other bits are 0: 10000000
කුඩාම සෘණ අගය වන්නේ ලකුණු බිටුව 1 **වන අතර අනෙකුත් සියලුම බිටු** 0 **වන විටය:** 10000000 **වේ.**

Decimal equivalent: $-2^7 = -128$
දශමය අගය: $-2^7 = -128$

The largest negative value is −1 (all bits are 1, 11111111).
විශාලතම සෘණ අගය වන්නේ −1 (සියලු බිටු 1, 11111111) වේ.

Negative range / සෘණ පරාසය: -128 to -1
Positive range / ධන පරාසය: 0 to 127

The range of decimal values in an 8-bit 2's complement system is -128 to +127.
බිටු 8හි 2හි අනුපූරක පද්ධතියක දශමය අගයන් පරාසය -128 සිට +127 දක්වා වේ.

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

5. Which of the following statements about Karnaugh Maps (K-maps) is correct?
කානෝ සිතියම් (K-maps) පිළිබඳ පහත සඳහන් ප්‍රකාශවලින් නිවැරදි වන්නේ කුමක්ද?

1) K-maps are more efficient than Boolean algebra for simplifying expressions with more than 8 variables.
විචල්‍ය 8කට වඩා වැඩි සංඛ්‍යාවක් සහිත ප්‍රකාශන සරල කිරීම සඳහා K-සිතියම් බූලියන් වීජ ගණිතයට වඩා කාර්යක්ෂම වේ.
2) In a K-map, adjacent cells differ in two variables, allowing double elimination at once.
K -සිතියමක, යාබද සෛල විචල්‍ය දෙකකින් වෙනස් වන අතර, එකවර ද්විත්ව ඉවත් කිරීමට ඉඩ සලසයි.
3) A group of 8 adjacent 1s in a K-map can eliminate 3 variables from the simplified expression.
K-සිතියමක යාබද 8 ක කණ්ඩායමකට සරල කළ ප්‍රකාශනයෙන් විචල්‍ය 3 ක් ඉවත් කළ හැකිය.
4) K-maps cannot be used for simplifying functions expressed in SOP form.
SOP ආකාරයෙන් ප්‍රකාශිත කාර්යයන් සරල කිරීම සඳහා K -සිතියම් භාවිතා කළ නොහැක.
5) For 4 variables, the K-map contains 8 cells.
විචල්‍ය 4ක් සඳහා, K-සිතියමේ සෛල 8ක් අඩංගු වේ.

Answer : 3)

Explanation

Karnaugh Maps (K-maps) are a graphical method used to minimize Boolean expressions by visually grouping adjacent cells containing 1s.
කානෝ සිතියම් (K-maps) යනු 1 අඩංගු යාබද සෛල දෘශ්‍යමය වශයෙන් කාණ්ඩගත කිරීමෙන් බූලියන් ප්‍රකාශන අවම කිරීමට භාවිතා කරන චිත්‍රක ක්‍රමයකි.

The number of variables eliminated depends on the size of the group formed.
ඉවත් කරන ලද විචල්‍ය ගණන පිහිටුවන ලද කණ්ඩායමේ ප්‍රමාණය මත රඳා පවතී.

- Option 1: Incorrect. K-maps are efficient only up to 5 or 6 variables. Beyond that, complexity increases, but they are not limited to 3 variables.
විකල්ප 1: වැරදියි. K-සිතියම් කාර්යක්ෂම වන්නේ විචල්‍ය 5ක් හෝ 6ක් දක්වා පමණි. ඉන් ඔබ්බට, සංකීර්ණතාව වැඩි වේ, නමුත් ඒවා විචල්‍ය 3කට සීමා නොවේ.

- Option 2: Incorrect. In K-maps, adjacency is defined as cells differing in only one variable, not more.
විකල්ප 2: වැරදියි. K-සිතියම් වල, යාබද බව අර්ථ දැක්වෙන්නේ එක් විචල්‍යයකින් පමණක් වෙනස් වන සෛල ලෙසය, වැඩි නොවේ

- Option 3: Correct. Grouping 8 adjacent cells (an octet) eliminates 3 variables, leaving only 1 variable in the simplified term.
විකල්ප 3: නිවැරදි කරන්න. යාබද සෛල 8ක් (අෂ්ටමයක්) කාණ්ඩගත කිරීමෙන් විචල්‍ය 3ක් ඉවත් වන අතර, සරල කළ පදයේ ඉතිරි වන්නේ 1 විචල්‍යයක් පමණි

- Option 4: Incorrect. K-maps can be used to simplify expressions in both SOP (Sum of Products) and POS (Product of Sums) forms.
  විකල්ප 4: වැරදියි.SOP (නිෂ්පාදන එකතුව) සහ POS (එකතුවල නිෂ්පාදිතය) ආකාර දෙකෙහිම ප්‍රකාශන සරල කිරීමට K-සිතියම් භාවිතා කළ හැක.

- Option 5: Incorrect. A 4-variable K-map has 16 cells, not 8.
  විකල්ප 5: වැරදියි. 4-විචල්‍ය K-සිතියමක සෛල 8ක් නොව, සෛල 16ක් ඇත.

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

6. Which of the following statements is true about programming languages?
   ක්‍රමලේඛන භාෂා සම්බන්ධයෙන් පහත සඳහන් කුමන ප්‍රකාශය සත්‍යද?

   1) High level programming languages can be run directly on any computer.
      ඉහළ මට්ටමේ ක්‍රමලේඛන භාෂා ඕනෑම පරිගණකයක කෙලින්ම ක්‍රියාත්මක කළ හැක.
   2) Assembly language belong to the first generation of computer languages.
      ඇසෙම්බ්ලි භාෂාව පරිගණක භාෂා වල පළමු පරම්පරාවට අයත් වේ.
   3) Machine languages run slower compared to other languages.
      අනෙකුත් භාෂාවලට සාපේක්ෂව යන්ත්‍ර භාෂා මන්දගාමී වේ.
   4) Python does not belong in high level programming languages.
      පයිතන් උසස් මට්ටමේ ක්‍රමලේඛන භාෂාවලට අයත් නොවේ.
   5) Assembler is a translator that translates assembly language into machine language.
      ඇසෙම්බ්ලරය යනු ඇසෙම්බ්ලි භාෂාව යන්ත්‍ර භාෂාවට පරිවර්තනය කරන පරිවර්තකයෙකි.

   Answer : 5)

   Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

   Option / විකල්ප 1)
   Incorrect. High-level languages require a compiler or interpreter to translate code into machine language before execution. They cannot run directly on hardware.
   වැරදියි. ක්‍රියාත්මක කිරීමට පෙර කේත යන්ත්‍ර භාෂාවට පරිවර්තනය කිරීමට ඉහළ මට්ටමේ භාෂාවලට සම්පාදකයක් හෝ අර්ථ විනිවිදකයක් අවශ්‍ය වේ. ඒවාට සෘජුවම දෘඩාංග මත ධාවනය කළ නොහැක.

   Option / විකල්ප 2)
   Incorrect. Assembly language is considered a second-generation language, as it is more advanced than first-generation machine language (binary code).
   වැරදියි. ඇසෙම්බ්ලි භාෂාව පළමු පරම්පරාවේ යන්ත්‍ර භාෂාවට (ද්විමය කේතය) වඩා දියුණු බැවින් එය දෙවන පරම්පරාවේ භාෂාවක් ලෙස සැලකේ.

   Option / විකල්ප 3)
   Incorrect. Machine language is the lowest-level language and runs the fastest because it directly interacts with hardware. Other languages need translation, which adds overhead.
   වැරදියි. යන්ත්‍ර භාෂාව පහළම මට්ටමේ භාෂාව වන අතර එය දෘඩාංග සමඟ සෘජුව අන්තර්ක්‍රියා කරන නිසා වේගයෙන්ම ධාවනය වේ. වෙනත් භාෂාවලට පරිවර්තන අවශ්‍ය වේ, එය පොදු කාර්ය එකතු කරයි.

   Option / විකල්ප 4)
   Incorrect. Python is a high-level programming language designed for readability and abstraction from hardware details.
   වැරදියි. පයිතන් යනු දෘඩාංග විස්තර වලින් කියවීමේ හැකියාව සහ වියුක්ත කිරීම සඳහා නිර්මාණය කර ඇති ඉහළ මට්ටමේ ක්‍රමලේඛන භාෂාවකි.

   Option / විකල්ප 5)
   Correct. An assembler converts assembly language instructions into machine code, making it executable by the computer.
   නිවැරදියි. ඇසෙම්බ්ලරය ඇසෙම්බ්ලි භාෂා උපදෙස් යන්ත්‍ර කේතයක් බවට පරිවර්තනය කරයි, එය පරිගණකයට ක්‍රියාත්මක කළ හැකිය.

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

7. Consider the following Python code snippet
පහත පයිතන් කේත කොටස සලකා බලන්න :

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

if numbers[2:5] == [30, 40, 50]:
    print("Match Found")
else:
    print("No Match")
```

What will be the output of the program?
වැඩසටහනේ ප්‍රතිදානය කුමක් වේද?

1) No Match
2) [30, 40, 50]
3) Error: Invalid slicing
4) None
5) Match Found

Answer : 5)

Explanation
In the given program, the list numbers contains seven elements.
දී ඇති වැඩසටහනේ, ලැයිස්තු අංකවල අංග හතක් අඩංගු වේ.

When the slicing operation numbers[2:5] is performed, Python retrieves the sublist starting from index 2 up to, but not including, index 5.
ස්ලයිසිං මෙහෙයුම් අංක [2:5] සිදු කරන විට, පයිතන් විසින් දර්ශකය 2 සිට දර්ශකය 5 ඇතුළත් නොකර උප ලැයිස්තුව ලබා ගනී.

This results in the slice [30, 40, 50]. The if statement then compares this slice with another list [30, 40, 50].
මෙය [30, 40, 50]. කොටසට හේතු වේ. ඉන්පසු results ප්‍රකාශය මෙම කොටස වෙනත් ලැයිස්තුවක් සමඟ සංසන්දනය කරයි [30, 40, 50].

Since both lists contain identical elements in the same order, the condition evaluates to True. As a result, the statement inside the if block is executed, and the output displayed on the screen is “Match Found.”
ලැයිස්තු දෙකෙහිම එකම අනුපිළිවෙලින් සමාන මූලාංග අඩංගු වන බැවින්, කොන්දේසිය සත්‍ය ලෙස ඇගයීමට ලක් කෙරේ. එහි ප්‍රතිඵලයක් ලෙස, if බ්ලොක් එක තුළ ඇති ප්‍රකාශය ක්‍රියාත්මක වන අතර, තිරයේ දිස්වන ප්‍රතිදානය “Match Found.”වේ.

Therefore, the correct answer is 5) / එබැවින් පිළිතුර 5) වේ.

8. What will be the output of the following code?
පහත කේතයේ ප්‍රතිදානය කුමක් වේද?

```
x = -9
if x < -10:
    if x % 3 == 0:
        x += 6
    else:
        x -= 3
else:
    if x % 2 == 0:
        x = x // 2
    else:
        x = abs(x) + 3
print(x)
```

1) -3 2) 9 3) 12 4) 6 5) 3

Answer : 3)

Explanation / පැහැදිලි කිරීම

The code first checks if `x` is less than `-10`. Since `x` is `-9`, which is not less than `-10`, the code moves to the `else` block. In this block, it checks if `x % 2 == 0` (if `x` is an even number). Since `-9` is odd, the code executes the `else` statement, which sets `x = abs(x) + 3`. The `abs(x)` function converts `-9` to `9`, and adding `3` results in `12`. Therefore, the final value of `x` is `12`, which is printed as the output.
කේතය පළමුව `x` `-10`ට වඩා අඩු දැයි පරීක්ෂා කරයි. `x` යනු `-9` වේ, එය `-10` ට නොඅඩු බැවින්, කේතය `else` කොටස වෙත ගමන් කරයි. මෙම කොටසෙ දී, එය `x % 2 == 0` (`x` ඉරට්ටේ අංකයක් නම්) පරීක්ෂා කරයි. `-9` ඔත්තේ බැවින්, කේතය `x = abs(x) + 3` සකසන `else` ප්‍රකාශය ක්‍රියාත්මක කරයි. `abs(x)` ශ්‍රිතය `-9` 9ට බවට පරිවර්තනය කරයි, සහ 3ක් එකතු කිරීමෙන් `12` ලැබේ. එබැවින්, ප්‍රතිදානය ලෙස මුද්‍රණය කරන `x` හි අවසාන අගය `12` වේ.

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

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

```
def foo(x):
    x[0] = ['def']
    x[1] = ['abc']
    return id(x)
q = ['abc', 'def']
print(id(q) == foo(q))
```

1) 2
2) True
3) False
4) None
5) Error

Answer : 2)

Explanation / පැහැදිලි කිරීම

The function `foo(x)` modifies the contents of the list `x` by changing its first two elements and then returns the memory address (`id(x)`) of the list. Since Python lists are mutable, any changes made to the list inside the function do not affect its memory location. The variable `q` is a list that is passed to the function `foo(q)`. Although the contents of `q` are modified, its memory address remains the same. The comparison `id(q) == foo(q)` evaluates to `True` because the `id(q)` before and after calling the function is identical, showing that the list retains its original memory location despite being altered.
`foo(x)` ශ්‍රිතය `x` ලැයිස්තුවේ පළමු මූලාංග දෙක වෙනස් කිරීමෙන් එහි අන්තර්ගතය වෙනස් කර ලැයිස්තුවේ මතක ලිපිනය (`id(x)`) ලබා දෙයි. පයිතන් ලැයිස්තු විකෘති වන බැවින්, ශ්‍රිතය තුළ ඇති ලැයිස්තුවට කරන ලද ඕනෑම වෙනස්කමක් එහි මතක ස්ථානයට බලපාන්නේ නැත. `q` විචල්‍යය යනු `foo(q)` ශ්‍රිතයට යවන ලැයිස්තුවකි. `q` හි අන්තර්ගතය වෙනස් කර ඇතත්, එහි මතක ලිපිනය එලෙසම පවතී. `id(q) == foo(q)` සංසන්දනය සත්‍ය ලෙස ඇගයීමට ලක් කරන්නේ ශ්‍රිතය ඇමතීමට පෙර සහ පසු `id(q)` එක සමාන වන නිසා, ලැයිස්තුව වෙනස් කර තිබියදීත් එහි මුල් මතක ස්ථානය රඳවා ගන්නා බව පෙන්වයි.

Output / ප්‍රතිදානය : `True`

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

10. Consider the following Python code snippet
පහත පයිතන් කේත කොටස සලකා බලන්න

```
def modify_list(lst):
    lst.append(100)
    lst = [200, 300]
    return lst

my_list = [1, 2, 3]
result = modify_list(my_list)
print(my_list, result)
```

What will be the output of this code?
මෙම කේතයේ ප්‍රතිදානය කුමක් වේද?

1) `[1, 2, 3] [200, 300]`
2) `[1, 2, 3, 100] [200, 300]`
3) `[1, 2, 3, 100] [1, 2, 3, 100]`
4) `[1, 2, 3] [1, 2, 3]`
5) `[200, 300] [200, 300]`

Answer : 2)

Explanation
- Step 1: my_list is initially [1, 2, 3].
  පියවර 1: my_list මුලින් [1, 2, 3] වේ.

- Step 2: Inside the function, lst.append(100) modifies the original list because lists are mutable and passed by reference. So my_list becomes [1, 2, 3, 100].
  පියවර 2: ශ්‍රිතය තුළ, lst.append(100) මුල් ලැයිස්තුව වෙනස් කරයි, මන්ද ලැයිස්තු විකෘති කළ හැකි අතර යොමුව මගින් සම්මත කළ හැකි බැවිනි. එබැවින් my_list [1, 2, 3, 100]. බවට පත්වේ.

- Step 3: Then, lst = [200, 300] reassigns the local variable lst to a new list. This does not affect my_list.
  පියවර 3: ඉන්පසු, lst = [200, 300] lst දේශීය විචල්‍යය නව ලැයිස්තුවකට නැවත පවරයි. මෙය my_list ට බලපාන්නේ නැත.

- Step 4: The function returns the new list [200, 300].
  පියවර 4: ශ්‍රිතය new list [200, 300] ආපසු ලබා දෙයි.

- Step 5: Printing my_list shows the original list modified by the append operation [1, 2, 3, 100], and result shows [200, 300].
  පියවර 5: my_list මුද්‍රණය කිරීමෙන් append මෙහෙයුම මගින් වෙනස් කරන ලද මුල් ලැයිස්තුව [1, 2, 3, 100] පෙන්වන අතර, ප්‍රතිඵලය [200, 300] පෙන්වයි.

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