සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved]

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 259 MARKING

2025/08/26
ictfromabc
Ravindu Bandaranayake

1. In the CPU, the ALU works closely with other registers. Which of the following statements is correct?
CPU තුළ, ALU අනෙකුත් රෙජිස්ටර් සමඟ සමීපව ක්‍රියා කරයි. පහත සඳහන් ප්‍රකාශවලින් නිවැරදි කුමක්ද?

   1) The ALU stores the next instruction to be executed.
   ක්‍රියාත්මක කළ යුතු ඊළඟ උපදෙස් ALU ගබඩා කරයි.
   2) The ALU retrieves instructions directly from RAM.
   ALU RAM වෙතින් උපදෙස් සෘජුවම ලබා ගනී.
   3) <u>The ALU performs operations, and results are often stored in the Accumulator.</u>
   ALU මෙහෙයුම් සිදු කරන අතර, ප්‍රතිඵල බොහෝ විට ඇකියුම්ලේටරය තුළ ගබඩා වේ.
   4) The ALU is responsible for decoding instructions before execution.
   ක්‍රියාත්මක කිරීමට පෙර උපදෙස් විකේතනය කිරීම සඳහා ALU වගකිව යුතුය.
   5) The ALU controls the timing of all CPU operations.
   සියලුම CPU මෙහෙයුම් වල කාලය ALU පාලනය කරයි.

Answer : 3)

Explanation / පැහැදිලි කිරීම:

The **ALU (Arithmetic Logic Unit)** is like the CPU's calculator. Its only job is to do math (like adding and subtracting) and compare numbers.
ALU **(Arithmetic Logic Unit)** CPU හි කැල්කියුලේටරය වැනිය. එහි එකම කාර්යය වන්නේ ගණිතය (එකතු කිරීම සහ අඩු කිරීම වැනි) කිරීම සහ සංඛ්‍යා සංසන්දනය කිරීමයි.

The **Accumulator** is a special holding spot right next to the ALU. When the ALU finishes a calculation, it immediately places the answer in the Accumulator.
ඇකියුමුලේටරය යනු ALU අසලම ඇති විශේෂ රැඳවුම් ස්ථානයකි. ALU ගණනය කිරීමක් අවසන් කළ විට, එය වහාම පිළිතුර සමුච්චය තුළ තබයි.

The **Control Unit** is the manager. It tells the ALU what to do and when to do it. It also handles getting instructions from memory and sending the final results to the right place.
පාලන ඒකකය කළමනාකරු වේ. එය ALU ට කළ යුතු දේ සහ එය කළ යුතු වේලාව කියයි. එය මතකයෙන් උපදෙස් ලබා ගැනීම සහ අවසාන ප්‍රතිඵල නිවැරදි ස්ථානයට යැවීම ද හසුරුවයි.

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

2. Which of the following statements about punch cards is correct?
සිදුරු කාඩ්පත් පිළිබඳ පහත සඳහන් ප්‍රකාශ අතුරෙන් නිවැරදි වන්නේ කුමක්ද?

A. Punch cards were used as an input device in the first generation of computers.
පළමු පරම්පරාවේ පරිගණකවල ආදාන උපාංගයක් ලෙස පන්චි කාඩ්පත් භාවිතා කරන ලදී.

B. Each hole in a punch card represented a binary digit.
පන්චි කාඩ්පතක සෑම සිදුරක්ම ද්විමය ඉලක්කමක් නියෝජනය කළේය.

C. Punch cards were primarily used in fourth-generation computers with microprocessors.
පන්චි කාඩ්පත් ප්‍රධාන වශයෙන් භාවිතා කරන ලද්දේ ක්ෂුද්‍ර සකසන සහිත 4වන පරම්පරාවේ පරිගණකවලය.

D. Punch cards were invented by Herman Hollerith for census data processing.
සංගණන දත්ත සැකසීම සඳහා හර්මන් හොලරිත් විසින් පන්චි කාඩ්පත් සොයා ගන්නා ලදී.

1) A and B only. A හා B පමණි.
2) A, B and C only. A, B හා C පමණි.
3) B and D only. B හා D පමණි.
4) <u>A, B and D only</u>. A, B හා D පමණි.
5) All of the above. ඉහත සියල්ලම.

Answer : 4)

Explanation / පැහැදිලි කිරීම:

Punch cards were early input and storage devices made of stiff paper with holes representing binary data (hole = 1, no hole = 0).
පන්චි කාඩ්පත් යනු ද්විමය දත්ත නියෝජනය කරන සිදුරු සහිත ෂන කඩදාසි වලින් සාදන ලද මුල් ආදාන සහ ගබඩා උපාංග විය (සිදුර = 1, සිදුරක් නැත =0.)

Invented by Herman Hollerith for the 1890 U.S. Census, they became the primary way to feed programs and data into first-generation computers like ENIAC and UNIVAC.
1890 එක්සත් ජනපද සංගණනය සඳහා හර්මන් හොලරිත් විසින් සොයා ගන්නා ලද ඒවා, ENIAC සහ UNIVAC. වැනි පළමු පරම්පරාවේ පරිගණකවලට වැඩසටහන් සහ දත්ත පෝෂණය කිරීමේ මූලික ක්‍රමය බවට පත්විය.

A standard IBM punch card had 80 columns, each column storing one character or instruction. They were used to enter programs (FORTRAN, COBOL), store data, and even produce output. However, punch cards were slow, bulky, and error-prone.
සම්මත IBM පන්චි කාඩ්පතක තිරු 80ක් තිබූ අතර, සෑම තිරුවකම එක් අකුරක් හෝ උපදෙස් ගබඩා කරන ලදී. ඒවා වැඩසටහන් (FORTRAN, COBOL) ඇතුළත් කිරීමට, දත්ත ගබඩා කිරීමට සහ ප්‍රතිදානය පවා නිපදවීමට භාවිතා කරන ලදී. කෙසේ වෙතත්, පන්චි කාඩ්පත් මන්දගාමී, විශාල සහ දෝෂ සහිත විය.

Therefore the correct answer is option 4).
එබැවින් නිවැරදි පිළිතුර විකල්පය 4) වේ.

3. Consider the following statements
පහත ප්‍රකාශ සලකා බලන්න :

A. ASCII can represent up to 256 characters if extended.
ASCII දිගු කළහොත් අක්ෂර 256ක් දක්වා නිරූපණය කළ හැකිය.

B. Unicode can represent more than one million characters.
යුනිකෝඩ් මඟින් අක්ෂර මිලියනයකට වඩා නියෝජනය කළ හැකිය.

C. ASCII and Unicode both use the binary number system internally.
ASCII Unicode යන දෙකම අභ්‍යන්තරව ද්විමය සංඛ්‍යා පද්ධතිය භාවිතා කරයි.

D. Unicode cannot represent characters beyond the English alphabet.
යුනිකෝඩ් මගින් ඉංග්‍රීසි හෝඩියෙන් ඔබ්බට අක්ෂර නිරූපණය කළ නොහැක.

1) <u>A, B and C only</u>. A, B හා C පමණි.
2) A and D only. A හා D පමණි.
3) B and C only. B හා C පමණි.
4) All of the above. ඉහත සියල්ලම.
5) A and C only. A හා C පමණි.

Answer : 1)

Explanation / පැහැදිලි කිරීම:

The question asks which statements about ASCII and Unicode are correct. Let's analyze each statement carefully
ASCII සහ Unicode පිළිබඳ කුමන ප්‍රකාශ නිවැරදි දැයි ප්‍රශ්නය අසයි. අපි එක් එක් ප්‍රකාශය ප්‍රවේශමෙන් විශ්ලේෂණය කරමු.

Statement / ප්‍රකාශය A
Standard ASCII uses 7 bits, supporting 128 characters. Extended ASCII uses 8 bits, increasing the total to 256 characters. This statement is true.
සම්මත ASCII බිටු 7ක් භාවිතා කරන අතර, අක්ෂර 128ක් සඳහා සහය දක්වයි. විස්තීරණ ASCII බිටු 8ක් භාවිතා කරයි, මුළු සංඛ්‍යාව අක්ෂර 256 දක්වා වැඩි කරයි. මෙම ප්‍රකාශය සත්‍ය වේ.

Statement / ප්‍රකාශය B
It can represent over a million unique characters. This statement is true.
එයට අද්විතීය අක්ෂර මිලියනයකට වඩා නියෝජනය කළ හැකිය. මෙම ප්‍රකාශය සත්‍යයකි.

Statement / ප්‍රකාශය C
All characters in both ASCII and Unicode are stored digitally using binary representation in computer memory. This statement is true.
ASCII සහ Unicode යන දෙකෙහිම සියලුම අක්ෂර පරිගණක මතකයේ ද්විමය නිරූපණය භාවිතයෙන් ඩිජිටල් ලෙස ගබඩා කර ඇත. මෙම ප්‍රකාශය සත්‍ය වේ.

Statement / ප්‍රකාශය D
This is false, because Unicode supports thousands of characters from multiple languages.
මෙය අසත්‍යයකි, මන්ද යුනිකෝඩ් බහු භාෂා වලින් අක්ෂර දහස් ගණනකට සහය දක්වයි.

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

4. The 8-bit two's complement number 11101010_2 is given. What is its equivalent decimal value in the original signed form?
8-බිටු දෙකේ අනුපූරක අංකය 11101010_2 ලබා දී ඇත. මුල් ස්වරූපය ආකාරයෙන් එහි සමාන දශම අගය කුමක්ද?

1) -22 2) -21 3) -54 4) -86 5) +234

Answer : 1)

Explanation / පැහැදිලි කිරීම:

- Check sign bit (MSB) / සලකුණු බිට් පරීක්ෂා කරන්න :
  The number is 11101010_2. / අංකය 11101010_2 වේ.
  Since the MSB = 1, it's a negative number in two's complement form.
  MSB = 1 බැවින්, එය දෙකේ අනුපූරක ආකාරයෙන් සෘණ සංඛ්‍යාවකි.

- Find the magnitude (original binary) / විශාලත්වය සොයන්න (මුල් ද්විමය) :
  Rule: To get the original number from two's complement, invert the bits and add 1.
  රීතිය: දෙකේ අනුපූරකයෙන් මුල් අංකය ලබා ගැනීමට, බිටු පෙරළා 1 එකතු කරන්න.
  Invert: 11101010 - 00010101
  Add 1: $00010101 + 1 = 00010110_2$

- Convert to decimal / දශමයට පරිවර්තනය කරන්න:
  $00010110_2 = 22_{10}$

- Apply the negative sign (since MSB = 1) / සෘණ ලකුණ යොදන්න (MSB = 1 බැවින්) :
  Final result / අවසාන පිළිතුර = –22

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

5. In a security alarm system implemented using an SR Latch, which of the following statements is correct?
SR Latch එකක් භාවිතයෙන් ක්‍රියාත්මක කරන ලද ආරක්ෂක අනතුරු ඇඟවීමේ පද්ධතියක, පහත සඳහන් ප්‍රකාශවලින් නිවැරදි කුමක්ද?

   1) The Set (S) input activates the alarm, and the Reset (R) input deactivates it.
   Set (S) ආදානය අනතුරු ඇඟවීම සක්‍රිය කරන අතර, Reset (R) ආදානය එය අක්‍රිය කරයි.
   2) The SR Latch can only store binary 0, not 1.
   SR Latch හට ගබඩා කළ හැක්කේ 1 නොව ද්විමය 0 පමණි.
   3) If both S and R are 1 simultaneously, the alarm enters a stable state.
   S සහ R දෙකම එකවර 1 නම්, අනතුරු ඇඟවීම ස්ථාවර තත්වයකට ඇතුළු වේ.
   4) SR Latches cannot be used in security alarm systems.
   ආරක්ෂක අනතුරු ඇඟවීමේ පද්ධතිවල SR Latch භාවිතා කළ නොහැක.
   5) The output of the SR Latch changes randomly regardless of inputs.
   ආදාන නොසලකා SR Latch හි ප්‍රතිදානය අහඹු ලෙස වෙනස් වේ.

   Answer : 1)

   Explanation / පැහැදිලි කිරීම:

   S = 1, R = 0
   Activates (sets) the alarm.
   එලාමය සක්‍රිය කරයි (sets).

   S = 0, R = 1
   Deactivates (resets) the alarm.
   එලාමය අක්‍රිය කරයි (resets).

   S = R = 0
   Maintains the previous state.
   පෙර තත්වය පවත්වා ගනී.

   S = R = 1
   Considered invalid/indeterminate, should be avoided.
   අවලංගු/අවිනිශ්චිත ලෙස සලකනු ලැබේ, වළක්වා ගත යුතුය.

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

6. Which combination of logic gates implements a Full Adder using two Half Adders?
අර්ධ ආකලක දෙකක් භාවිතා කරමින් පූර්ණ ආකලකයක් ක්‍රියාත්මක කරන්නේ කුමන තාර්කික ද්වාර සංයෝජනයෙන්ද?

   1) Two XOR gates, two AND gates, one OR gate
   2) One XOR gate, two AND gates, two OR gates
   3) Three XOR gates, three AND gates, two OR gates
   4) Two OR gates and three NAND gates
   5) Two AND gates, two OR gates, one NOT gate

   Answer : 1)

   Explanation / පැහැදිලි කිරීම:

A Full Adder adds three binary inputs: A, B, and Cin(carry-in), producing a Sum and a Carry-out. One efficient way to implement a Full Adder is by using two Half Adders and an OR gate.
පූර්ණ ආකලකයක් ද්විමය ආදාන තුනක් එකතු කරයි: A, B, සහ Cin(carry-in), Sum සහ Carry-out නිපදවයි. පූර්ණ ආකලකයක් ක්‍රියාත්මක කිරීමට එක් කාර්යක්ෂම ක්‍රමයක් වන්නේ අර්ධ ආකලක දෙකක් සහ OR ද්වාරයක් භාවිතා කිරීමයි.

In this design, the first Half Adder takes inputs A and B, producing an intermediate Sum1 and Carry1.
මෙම සැලසුමේදී, පළමු අර්ධ ආකලකය A සහ B ආදාන ලබාගෙන අතරමැදි Sum1 සහ Carry1. නිපදවයි.

The second Half Adder adds Sum1 and the carry-in Cin, producing the final Sum and an intermediate Carry2.
දෙවන අර්ධ ආකලකය Sum1 සහ carry-in Cin, එකතු කරයි, අවසාන එකතුව සහ අතරමැදි Carry2 නිපදවයි.

Finally, the Full Adder's Carry-out is obtained by combining Carry1 and Carry2 using an OR gate. The two Half Adders each require one XOR gate for the sum and one AND gate for the carry, so in total, this design uses two XOR gates, two AND gates, and one OR gate.
අවසාන වශයෙන්, පූර්ණ ආකලකයේ Carry-out එක ලබා ගන්නේ OR ගේට්ටුවක් භාවිතා කර Carry1 සහ Carry2 ඒකාබද්ධ කිරීමෙනි. අර්ධ ආකලක දෙකටම එකතුව සඳහා එක් XOR ගේට්ටුවක් සහ රැගෙන යාම සඳහා එක් AND ගේට්ටුවක් අවශ්‍ය වේ, එබැවින් සමස්තයක් වශයෙන්, මෙම සැලසුම XOR ගේට්ටු දෙකක්, AND ගේට්ටු දෙකක් සහ එක් OR ගේට්ටුවක් භාවිතා කරයි.

This combination efficiently realizes the Full Adder function with a minimum number of logic gates.
මෙම සංයෝජනය අවම තාර්කික ද්වාර සංඛ්‍යාවක් සමඟින් පූර්ණ ආකලකයේ ශ්‍රිතය කාර්යක්ෂමව සාක්ෂාත් කර ගනී.

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

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

```
def generate_pattern(n):
    result = []
    for i in range(n):
        row = []
        for j in range(i + 1):
            row.append(j + 1)
        result.append(row)
    return result

pattern = generate_pattern(4)

for line in pattern:
    print(line)
```

1)
```
[1]
[2]
[3]
[4]
```

2)
```
[1]
[1, 2]
[1, 2, 3]
[1, 2, 3, 4]
```

3)
```
[1]
[1, 2]
[1, 2, 3]
[1, 2, 3]
```

4)
```
[1]
[2, 3]
[4, 5, 6]
[7, 8, 9, 10]
```

5)
```
[1]
[2, 2]
[3, 3, 3]
[4, 4, 4, 4]
```

Answer : 2)

Explanation / පැහැදිලි කිරීම:

The given Python code defines a function `generate_pattern(n)` that generates a pattern of lists, where each list contains consecutive numbers starting from `1` up to the length of the current row. For an input `n = 4`, the outer loop runs from `0` to `3`, and in each iteration, it creates a list with numbers starting from 1 to the value of the current iteration index plus 1. These lists are then added to the result list. The code then prints each row of the generated pattern. The output of this code is,
ලබා දී ඇති පයිතන් කේතය ලැයිස්තු රටාවක් ජනනය කරන `generate_pattern(n)` ශ්‍රිතයක් නිර්වචනය කරයි, එහිදී එක් එක් ලැයිස්තුවේ 1 සිට වත්මන් පේළියේ දිග දක්වා අඛණ්ඩ අංක අඩංගු වේ. ආදානය `n = 4` සඳහා, පිටත ලූපය `0` සිට `3` දක්වා දිවෙන අතර, එක් එක් පුනරාවර්තනය තුළ, එය 1 සිට වත්මන් පුනරාවර්තන දර්ශකයේ අගය සහ 1 දක්වා සංඛ්‍යා සහිත ලැයිස්තුවක් සාදයි. ඉන්පසු මෙම ලැයිස්තු එකතු කරනු ලැබේ. ප්‍රතිඵල ලැයිස්තුව ඉන්පසු කේතය ජනනය කරන ලද රටාවේ එක් එක් පේළිය මුද්‍රණය කරයි. මෙම කේතයේ ප්‍රතිදානය වන්නේ,

```
[1]
[1, 2]
[1, 2, 3]
[1, 2, 3, 4]
```

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

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

```
def find_max(nums):
    max_val = nums[0]
    for n in nums:
        if n > max_val:
            max_val = n
    return max_val

vals = [-3, -1, -4, -2, -5]
print(find_max(vals))
```

1) `-1` 2) `1` 3) `-2` 4) `2` 5) `0`

Answer : 1)

Explanation / පැහැදිලි කිරීම:

The function `find_max(nums)` finds the maximum value in a list of numbers. It starts by assigning the first element of the list (`nums[0]`) to `max_val`. Then, it iterates through the list, comparing each number (`n`) with `max_val`. If `n` is greater than `max_val`, it updates `max_val` to that value. In this case, the list `vals = [-3, -1, -4, -2, -5]` is processed, and since `-1` is the largest number in the list, the function returns `-1`. Therefore, the output is `-1`.
`find_max(nums)` ශ්‍රිතය සංඛ්‍යා ලැයිස්තුවක උපරිම අගය සොයා ගනී. එය ආරම්භ වන්නේ ලැයිස්තුවේ පළමු මූලද්‍රව්‍යය (`nums[0]`) `max_val` වෙත පැවරීමෙනි. ඉන්පසුව, එය එක් එක් සංඛ්‍යාව (`n`) `max_val` සමඟ සංසන්දනය කරමින් ලැයිස්තුව හරහා පුනරාවර්තනය වේ. `n` `max_val` ට වඩා වැඩි නම්, එය එම අගයට `max_val` යාවත්කාලීන කරයි. මෙම අවස්ථාවෙහිදී, ලැයිස්තුව `vals = [-3, -1, -4, -2, -5]` සකසන ලද අතර, `-1` ලැයිස්තුවේ විශාලතම අංකය වන බැවින්, ශ්‍රිතය `-1` ලබා දෙයි. .
එබැවින්, ප්‍රතිදානය `-1` වේ.

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

9. Which of the following statements about Python file handling is **true**?
පයිතන් ගොනු හැසිරවීම පිළිබඳ පහත සඳහන් ප්‍රකාශවලින් සත්‍ය කුමක්ද?

1) The `close()` method is optional because Python automatically saves and closes files immediately after writing.
ලිවීමෙන් පසු පයිතන් ස්වයංක්‍රියව ලිපිගොනු සුරැකීම සහ වසා දැමීම නිසා `close()` ක්‍රමය විකල්පමය වේ.

2) Opening a file in "r" mode allows both reading and writing, and truncates the file to zero length.
"r" ආකාරයෙන් ගොනුවක් විවෘත කිරීමෙන් කියවීම සහ ලිවීම යන දෙකම ඉඩ ලබා දෙන අතර ගොනුව ශූන්‍ය දිගට කපා දමයි.

3) Opening a file in "w" mode deletes the existing contents of the file before writing new data.
"w" ආකාරයෙන් ගොනුවක් විවෘත කිරීමෙන් නව දත්ත ලිවීමට පෙර ගොනුවේ පවතින අන්තර්ගතය මකා දමයි.

4) The "a" mode allows reading from the file.
"a" ආකාරයෙන් ගොනුවෙන් කියවීමට ඉඩ දෙයි.

5) Opening a file in "r" mode will create a new file if it does not exist.
"r" ආකාරයෙන් ගොනුවක් විවෘත කිරීමෙන් එය නොපවතී නම් නව ගොනුවක් නිර්මාණය වේ.

Answer : 3)

Explanation / පැහැදිලි කිරීම:

"w" mode opens a file for writing and overwrites any existing content.
"w" ආකාරය ලිවීම සඳහා ගොනුවක් විවෘත කරන අතර පවතින ඕනෑම අන්තර්ගතයක් උඩින් ලියයි.

`close()` is not automatic, you should close files or use with.
`close()` ස්වයංක්‍රිය නොවේ, ඔබ ගොනු වසා දැමිය යුතුය නැතහොත් භාවිතා කළ යුතුය.

"r" mode only reads, doesn't write or delete.
"r" ආකාරය කියවන්නේ පමණි, ලියන්නේ හෝ මකා දමන්නේ නැත.

"a" mode can only write at the end, not read.
"a" ආකාරය අවසානයේ පමණක් ලිවිය හැකිය, කියවිය නොහැක.

"r" mode won't create a file if it doesn't exist.
"r" ආකාරය ගොනුවක් නිර්මාණය නොකරනු ඇත, එය නොපවතී නම්.

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

10. Which of the following best describes the purpose of the `close()` function in Python file handling?
පයිතන් ගොනු හැසිරවීමේදී `close()` ශ්‍රිතයේ අරමුණ පහත සඳහන් ඒවායින් වඩාත් හොඳින් විස්තර කරන්නේ කුමක්ද?

    1) It deletes the file after use.
    එය භාවිතයෙන් පසු ගොනුව මකා දමයි.
    2) It saves the file permanently and makes it read-only.
    එය ගොනුව ස්ථිරවම සුරකින අතර එය කියවීමට පමණක් ඉඩ සලසයි.
    3) It flushes any unwritten data from the buffer and releases the file resource.
    එය බෆරයෙන් ඕනෑම ලිඛිත දත්ත ඉවත් කර ගොනු සම්පත මුදා හරියි.
    4) It reopens the file for reading after writing.
    එය ලිවීමෙන් පසු කියවීම සඳහා ගොනුව නැවත විවෘත කරයි.
    5) It automatically appends new data to the file.
    එය ස්වයංක්‍රියව ගොනුවට නව දත්ත එකතු කරයි.

    Answer : 3)

    Explanation / පැහැදිලි කිරීම:

    When you write to a file, Python may temporarily keep the data in a buffer
    ඔබ ගොනුවක් ලියන විට, Python තාවකාලිකව දත්ත බෆරයක තබා ගත හැකිය.

    `close()` makes sure this buffered data is flushed (written) to the actual file.
    `close()` මෙම බෆරය කළ දත්ත සත්‍ය ගොනුවට ෆ්ලෂ් කර ඇති බව සහතික කරයි.

    It also releases the system resource so other programs or parts of your code can use the file safely.
    එය පද්ධති සම්පත ද මුදා හරින බැවින් අනෙකුත් වැඩසටහන් හෝ ඔබේ කේතයේ කොටස් ගොනුව ආරක්ෂිතව භාවිතා කළ හැකිය.

    Without calling `close()`, you risk data not being saved properly.
    `close()` ඇමතීමෙන් තොරව, ඔබ දත්ත නිසි ලෙස සුරැකෙන්නේ නැති අවදානමක් ඇත.

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