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 252 MARKING

2025/08/19
ictfromabc
Ravindu Bandaranayake

1. Which is the correct order from fastest access speed to slowest?
වේගවත්ම ප්‍රවේශ වේගයේ සිට මන්දගාමීම දක්වා නිවැරදි අනුපිළිවෙල කුමක්ද?

   1) Cache > RAM > SSD > Optical disc / වාරක මතකය > RAM > SSD > ප්‍රකාශ තැටි
   2) RAM > Cache > SSD > Optical disc / RAM > වාරක මතකය > SSD > ප්‍රකාශ තැටි
   3) Cache > SSD > Optical disc > RAM / වාරක මතකය > SSD > ප්‍රකාශ තැටි > RAM
   4) SSD > Cache > RAM > Optical disc / SSD > වාරක මතකය > RAM > ප්‍රකාශ තැටි
   5) Cache > RAM > Optical disc > SSD / වාරක මතකය > RAM > ප්‍රකාශ තැටි > SSD

   Answer: 1)

   Explanation / පැහැදිලි කිරීම:

   Cache (Fastest) / වාරක මතකය (වේගවත්ම):
   Cache memory is a small, high-speed memory located inside the CPU. It stores frequently used instructions and data to reduce CPU access time.
   වාරක මතකය යනු CPU තුළ පිහිටා ඇති කුඩා, අධිවේගී මතකයකි. එය CPU ප්‍රවේශ කාලය අඩු කිරීම සඳහා නිතර භාවිතා වන උපදෙස් සහ දත්ත ගබඩා කරයි.

   RAM (Random Access Memory) / RAM (සසම්භාවී ප්‍රවේශ මතකය):
   RAM is much larger than cache and stores data temporarily while programs are running. It is slower than cache but much faster than any storage device like SSDs or hard drives.
   RAM යනු වාරක මතකයට වඩා බෙහෙවින් විශාල වන අතර වැඩසටහන් ක්‍රියාත්මක වන අතරතුර තාවකාලිකව දත්ත ගබඩා කරයි. එය වාරක මතකයට වඩා මන්දගාමී නමුත් SSD හෝ දෘඩ තැටි වැනි ඕනෑම ගබඩා උපාංගයකට වඩා බොහෝ වේගවත් වේ.

   SSD (Solid State Drive) / SSD (ඝන තත්ව ධාවකය):
   An SSD is a non-volatile storage device, meaning it retains data even when power is off. While much faster than traditional hard drives, it's still slower than both cache and RAM.
   SSD යනු නෂ්ට නොවන ගබඩා උපාංගයකි, එනම් බලය අක්‍රිය වූ විට පවා එය දත්ත රඳවා ගනී. සාම්ප්‍රදායික දෘඩ තැටි වලට වඩා බොහෝ වේගවත් වුවද, එය තවමත් වාරකය සහ RAM දෙකටම වඩා මන්දගාමී වේ.

   Optical Disc (Slowest) / ප්‍රකාශ තැටිය (මන්දගාමී):
   Devices like CDs and DVDs are accessed via mechanical movement (laser read/write), making them the slowest form of data storage in this list. Access times are in milliseconds.
   CD සහ DVD වැනි උපාංග යාන්ත්‍රික චලනය (ලේසර් කියවීම/ලිවීම) හරහා ප්‍රවේශ වන අතර, ඒවා මෙම ලැයිස්තුවේ දත්ත ගබඩා කිරීමේ මන්දගාමීම ආකාරය බවට පත් කරයි. ප්‍රවේශ කාලය මිලි තත්පර වලින් වේ.

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

2. Which of the following best describes the correct sequence of operations in the Fetch-Execute Cycle of a CPU?
CPU එකක ආහරණ ක්‍රියාකරවුම් චක්‍රයේ නිවැරදි මෙහෙයුම් අනුපිළිවෙල පහත සඳහන් ඒවායින් වඩාත් හොඳින් විස්තර කරන්නේ කුමක්ද?

1) Execute → Decode → Fetch / ක්‍රියාත්මක කිරීම → විකේතනය → ලබා ගැනීම
2) Fetch → Execute → Decode / ලබා ගැනීම → ක්‍රියාත්මක කිරීම → විකේතනය
3) Decode → Fetch → Execute / විකේතනය → ලබා ගැනීම → ක්‍රියාත්මක කිරීම
4) Fetch → Decode → Execute / ලබා ගැනීම → විකේතනය → ක්‍රියාත්මක කිරීම
5) Decode → Execute → Fetch / විකේතනය → ක්‍රියාත්මක කිරීම → ලබා ගැනීම

Answer: 4)

Explanation / පැහැදිලි කිරීම:

Fetch / ලබා ගැනීම:
The CPU starts by fetching (retrieving) the instruction from the memory. This is the first step because the processor needs to know what operation to perform.
CPU එක ආරම්භ වන්නේ මතකයෙන් උපදෙස් ලබා ගැනීමෙන් ය. මෙය පළමු පියවර වන්නේ සකසනය සිදු කළ යුතු මෙහෙයුම දැන සිටිය යුතු බැවිනි.

Decode / විකේතනය:
After fetching the instruction, the CPU decodes it to understand what actions are required. This step translates the instruction into signals that control other parts of the CPU.
උපදෙස් ලබා ගැනීමෙන් පසු, අවශ්‍ය ක්‍රියාමාර්ග තේරුම් ගැනීමට CPU එක එය විකේතනය කරයි. මෙම පියවර උපදෙස් CPU හි අනෙකුත් කොටස් පාලනය කරන සංඥා බවට පරිවර්තනය කරයි.

Execute / ක්‍රියාත්මක කිරීම:
Finally, the CPU executes the decoded instruction, performing the operation such as arithmetic calculations, moving data, or controlling input/output.
අවසාන වශයෙන්, CPU විකේතනය කළ උපදෙස් ක්‍රියාත්මක කරයි, අංක ගණිත ගණනය කිරීම්, දත්ත ගෙනයාම හෝ ආදාන/ප්‍රතිදානය පාලනය කිරීම වැනි ක්‍රියාවන් සිදු කරයි.

Any other order wouldn't work because you must first get the instruction (fetch) before you can understand (decode) and carry it out (execute). For example, you cannot execute an instruction before fetching it, or decode before fetching it.
වෙනත් ඕනෑම අනුපිළිවෙලක් ක්‍රියා නොකරනු ඇත, මන්ද ඔබට තේරුම් ගැනීමට (විකේතනය කිරීමට) සහ එය ක්‍රියාත්මක කිරීමට (ක්‍රියාත්මක කිරීමට) පෙර ඔබ මුලින්ම උපදෙස් ලබා ගත යුතුය. උදාහරණයක් ලෙස, ඔබට එය ලබා ගැනීමට පෙර උපදෙස් ක්‍රියාත්මක කළ නොහැක, නැතහොත් එය ලබා ගැනීමට පෙර විකේතනය කළ නොහැක.

So, the logical and standard order is Fetch → Decode → Execute. The answer option 4).
එබැවින්, තාර්කික සහ සම්මත අනුපිළිවෙල ලබා ගැනීම → විකේතනය → ක්‍රියාත්මක කිරීම වේ. පිළිතුරු විකල්පය 4) වේ.

3. If the following numbers are arranged in descending order, the answer is,
පහත සංඛ්‍යා අවරෝහණ පිළිවෙලට සැකසූ විට පිළිතුර වන්නේ,

A. BC_{16} B.10101011_2 C. 179_{10} D. 261_8

1) A,D,C,B 2) A,C,B,D 3) C,A,D,B 4) <u>A,C,D,B</u> 5) B,A,C,D

Answer: 4)

Explanation / පැහැදිලි කිරීම:

Hexa-decimal to decimal

BC_{16}	
B	C
16^1	16^0
16×11	1×12
176+12=188	
$= 188_{10}$	

Octal to decimal

261_8		
2	6	1
8^2	8^1	8^0
64×2	8×6	1×1
128+48+1=177		
$= 177_{10}$		

Binary to decimal

10101011_2							
1	0	1	0	1	0	1	1
2^7=128	2^6=64	2^5=32	2^4=16	2^3=8	2^2=4	2^1=2	2^0=1
128	0	32	0	8	0	2	1
128+32+8+2+1=171							
$= 171_{10}$							

188 > 179 > 177 > 171
$BC_{16} > 179_{10} > 261_8 > 10101011_2$
A > C > D > B

Therefore, the answer is 4).
එබැවින්, පිළිතුර 4) **වේ.**

4. Given the 4-bit binary number 1101, the output of a circuit that takes each bit as input into the logic expression:
4-bit ද්විමය අංකය 1101 **ලබා දී ඇති විට, තාර්කික ප්‍රකාශනයට එක් එක් බිටුව ආදානය ලෙස ගන්නා පරිපථයක ප්‍රතිදානය:**

Y = (A AND NOT B) OR (C XOR D)

where A is the most significant bit and D the least significant bit, the output will be:
A යනු වැඩිම වෙසෙසි බිටුව වන අතර D යනු අඩුම වෙසෙසි බිටුව වන අතර, මෙහි ප්‍රතිදානය වනුයේ:

1) 0 2) <u>1</u> 3) 10 4) 11 5) undefined

Answer: 2)

Explanation / පැහැදිලි කිරීම:

The input is a 4-bit binary number / **ආදානය** 4-bit **ද්විමය අංකයකි:**

1	1	0	1
MSB			LSB
A	B	C	D

The AND operator takes two inputs and returns 1 only if both inputs are 1; otherwise, it returns 0.
AND මෙහෙයවනය ආදාන දෙකක් ගෙන 1 ආපසු ලබා දෙන්නේ ආදාන දෙකම 1 නම් පමණි එසේ නොමැති නම්, එය 0 ආපසු ලබා දෙයි.

The NOT operator inverts the input bit, turning 1 into 0 and 0 into 1, effectively flipping the value.
NOT මෙහෙයවනය ආදාන බිටුව ප්‍රතිලෝම කරයි, 1 0 බවටත් 0 1 බවටත් හරවමින්, අගය ඵලදායී ලෙස පෙරළයි.

The XOR (exclusive OR) operator compares two bits and returns 1 only when exactly one of the bits is 1; if both bits are the same, it returns 0.
XOR (exclusive OR) මෙහෙයවනය බිටු දෙකක් සංසන්දනය කර 1 ආපසු ලබා දෙන්නේ හරියටම බිටුවක් 1 වූ විට පමණි බිටු දෙකම සමාන නම්, එය 0 ආපසු ලබා දෙයි.

Finally, the OR operator takes two inputs and returns 1 if at least one input is 1; it returns 0 only when both inputs are 0.
අවසාන වශයෙන්, OR මෙහෙයවනය ආදාන දෙකක් ගෙන අවම වශයෙන් එක් ආදානයක් 1 නම් 1 ආපසු ලබා දෙයි ආදාන දෙකම 0 වූ විට පමණක් එය 0 ආපසු ලබා දෙයි.

In this expression, these operators combine to produce a result where either the first condition (A AND NOT B) is true, or the second condition (C XOR D) is true, giving a logical output based on the input bits.
මෙම ප්‍රකාශනයේ දී, මෙම මෙහෙයවන ඒකාබද්ධ වී පළමු කොන්දේසිය (A AND NOT B) සත්‍ය වන හෝ දෙවන කොන්දේසිය (C XOR D) සත්‍ය වන ප්‍රතිඵලයක් නිපදවයි, එමඟින් ආදාන බිටු මත පදනම්ව තාර්කික ප්‍රතිදානයක් ලබා දේ.

(A AND NOT B) OR (C XOR D)
(1 AND 0) OR (0 XOR 1)
0 OR 1
= 1

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

5. $(A + \overline{B})(C + D) + A\overline{C}$

Choose the correct statement regarding the following Karnaugh map based on the above Boolean expression.
ඉහත බූලියන් ප්‍රකාශනය මත පදනම්ව පහත සඳහන් කානෝ සිතියම සම්බන්ධයෙන් නිවැරදි ප්‍රකාශය තෝරන්න.

CD \ AB	00	01	11	10
00	A	B	C	D
01	E	F	G	H
11	I	J	K	L
10	M	N	O	P

1) Only A, B, C, D, H, and L should be 1.
A, B, C, D, H සහ L පමණක් 1 විය යුතුය.
2) G, J, K, O should be 0.
G, J, K, O 0 විය යුතුය.
3) Only A, E, F, I, and M should be 1.
A, E, F, I, සහ M පමණක් 1 විය යුතුය.
4) C, D, H, K, and L should be 0.
C, D, H, K, සහ L 0 විය යුතුය.
5) <u>All of the above statements are incorrect.</u>
ඉහත ප්‍රකාශ සියල්ල වැරදියි.

Answer: 5)

Explanation / පැහැදිලි කිරීම:

To verify the values in the Karnaugh map, we first construct the truth table by evaluating the given Boolean expression for all possible input combinations of A, B, C, and D.
කානෝ සිතියමේ අගයන් සත්‍යාපනය කිරීම සඳහා, අපි මුලින්ම A, B, C, සහ D හි සියලු හැකි ආදාන සංයෝජන සඳහා ලබා දී ඇති බූලියන් ප්‍රකාශනය ඇගයීමෙන් සත්‍යතා වගුව ගොඩනඟමු.

A	B	C	D	$(A + \overline{B})$	$(C + D)$	$A\overline{C}$	Q
0	0	0	0	1	0	0	0
0	0	0	1	1	1	0	1
0	0	1	0	1	1	0	1
0	0	1	1	1	1	0	1
0	1	0	0	0	0	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	1	1
1	0	1	0	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	0	1	0	1	1
1	1	0	1	1	1	1	1
1	1	1	0	1	1	0	1
1	1	1	1	1	1	0	1

According to the above truth table, C, D, E, G, H, I, K, L, M, O and P should be 1.
ඉහත සත්‍යතා වගුවට අනුව, C, D, E, G, H, I, K, L, M, O සහ P 1 විය යුතුය.

Therefore the correct answer is 5) All of the above statements are incorrect.
එබැවින් නිවැරදි පිළිතුර 5) ඉහත ප්‍රකාශ සියල්ල වැරදියි.

6. Select the most simplified sum of product (SOP) Boolean expression for $C\overline{A} + (BD + \overline{BA})A$.
$C\overline{A} + (BD + \overline{BA})A$ වඩාත් සරල කළ ගුණිතයන්ගේ එකතු(SOP) බූලියානු ප්‍රකාශනය තෝරන්න.

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

Answer: 4)

Explanation / පැහැදිලි කිරීම:

Let's simplify the given Boolean expression.
ලබා දී ඇති බූලියන් ප්‍රකාශනය සරල කරමු.

$= C\overline{A} + (BD + \overline{BA})A$
$= C\overline{A} + (BD + \overline{B} + \overline{A})A$ De Morgan's law / ද මුවාර් න්‍යාය
$= C\overline{A} + (\overline{B} + D + \overline{A})A$ Redundancy law / සමතිරික්ත න්‍යාය
$= C\overline{A} + A\overline{B} + AD + A\overline{A}$ Distributive law / විඝටන න්‍යාය
$= C\overline{A} + A\overline{B} + AD + 0$ Inverse law / ප්‍රතිලෝම න්‍යාය
$= C\overline{A} + A\overline{B} + AD$ Identity law / සර්වසාම්‍ය න්‍යාය
$= A\overline{B} + AD + \overline{A}C$ Commutative law / න්‍යාදේශ න්‍යාය

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

7.
```
dct = {}
dct['1'] = (1, 2)
dct['2'] = (2, 1)
for x in dct.keys():
      print(dct[x][1], end='')
```

What is the output of the above python code?
ඉහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

1) 21 2) 12 3) [21] 4) 1,2 5) 2,1

Answer: 1)

Let's take a step-by-step process of this Python code.
මෙම පයිතන් කේතය මගින් සිදුවන ක්‍රියාවලිය පියවරින් පියවර සලකා බලමු.

`dct = {}`

This codeline initializes an empty dictionary called dct.
මෙම කේත පේළිය මගින් dct නමින් හිස් dictionary එකක් නිර්මාණය කරයි.

`dct['1'] = (1, 2)`

This line assigns the tuple (1, 2) to the key '1' in the dictionary.
මෙමගින් (1,2) tuple එක, dct හි '1' යතුරේ අගය ලෙස ගබඩා කරගනියි.

`dct['2'] = (2, 1)`

This line assigns the tuple (2, 1) to the key '2' in the dictionary.
මෙමගින් (2,1) tuple එක, dct හි '2' යතුරේ අගය ලෙස ගබඩා කරගනියි.

`for x in dct.keys():`

Iterates over the number of keys in the dictionary.
Dictionary හි ඇති යතුරු ගණනට අනුව පුනරාවර්තනය කරයි.

`print(dct[x][1], end='')`

This code line prints the second element of each tuple corresponding to the current key x in the loop, without a newline character at the end.
මෙම කේත පේළිය මගින් නව රේඛාවකට යාමෙන් තොරව ලූපයේ වත්මන් යතුර x ට අනුරූප වන එක් එක් tuple හි දෙවන අයිතමය මුද්‍රණය කරයි.

When x is '1', `dct[x]` accesses the value associated with the key '1', which is the tuple (1, 2). Then, `dct[x][1]` extracts the second element of this tuple, which is 2, and prints it.
x = 1 වන විට, `dct[x]` මගින් tuple (1,2) හි යතුර වන '1' යතුර හා සම්බන්ධ අගයට ප්‍රවේශ වේ. ඉන්පසුව, `dct[x][1]` මගින් මෙම tuple හි දෙවන අයිතමය වන 2 මුද්‍රණය කරයි.

When x is '2', `dct[x]` accesses the value associated with the key '2', which is the tuple (2, 1). Then, `dct[x][1]` extracts the second element of this tuple, which is 1, and prints it. Therefore, the output of this code will be 21.
x = 2 වන විට, `dct[x]` මගින් tuple (2,1) හි යතුර වන '2' යතුර හා සම්බන්ධ අගයට ප්‍රවේශ වේ. ඉන්පසුව, `dct[x][1]` මගින් මෙම tuple හි දෙවන අයිතමය වන 1 මුද්‍රණය කරයි. එබැවින්, මෙම කේතයේ ප්‍රතිදානය 21 වනු ඇත.

So the correct answer number is 1.
එමනිසා නිවැරදි පිළිතුරු අංකය 1 වේ.

8. What is the output of the following Python code if the input was 15?
ආදානය 15 නම් පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් ද?

```
ans=int(input())
sum=(ans // (ans % 4))** 4
print(sum)
```

1) 625 2) 0 3) 625.0 4) 20 5) 9

Answer: 1)

Let's break down the Python code step by step.
අපි පයිතන් කේතය පියවරෙන් පියවර බිඳ දමමු.

Reading Input / ආදානය කියවීම.
The input is 15, so ans is assigned the value 15.
ආදානය 15 වේ, එබැවින් ans ට 15 අගය පවරනු ලැබේ.

Calculating sum / එකතුව ගණනය කිරීම.
ans % 4 is the remainder when 15 is divided by 4. 15 % 4 = 3 (since 15 divided by 4 gives 3 as the remainder). Now, calculate the integer division ans // (ans % 4). 15 // 3 = 5 (since 15 divided by 3 gives 5 as the quotient). Finally, raise the result to the power of 4(5 ** 4 = 5 * 5 * 5 * 5 = 625).
ans % 4 යනු 15 4න් බෙදූ විට ඉතිරිය. 15 % 4 = 3 (15 4න් බෙදූ විට 3 ඉතිරිය ලෙස ලැබෙන බැවින්). දැන්, නිබිල බෙදීම ans // (ans % 4) ගණනය කරන්න. 15 // 3 = 5 (15 3 න් බෙදූ විට 5 නඃංකය ලෙස ලැබෙන බැවින්). අවසාන වශයෙන්, ප්‍රතිඵලය 4 බලයට ඔසවන්න(5 ** 4 = 5 * 5 * 5 * 5 = 625) .

Output / ප්‍රතිදානය
print(sum) prints the value of sum, which is 625.So the output of the code when the input is 15 is 625.
print(sum) sum හි අගය මුද්‍රණය කරයි, එය 625 වේ. එබැවින් ආදානය 15 වන විට කේතයේ ප්‍රතිදානය 625 වේ.

So, the correct answer is 1) 625
මේ අනුව, නිවැරදි පිළිතුර 1) 625 වේ

9. What will be the output of the following Python program?
පහත දැක්වෙන පයිතන් වැඩසටහනේ ප්‍රතිදානය කුමක් වේද?

```
def mystery_function(n):
    if n == 10:
        return 0
    else:
        return n + mystery_function(n+1)

print(mystery_function(3))
```

1) 0 2) 13 3) 10 4) 42 5) Infinity loop

Answer: 4)

The Python program defines a recursive function mystery_function(n).
Python වැඩසටහන මගින් පුනරාවර්තන ශ්‍රිතයක් mystery_function(n) නිර්වචනය කරයි.

When called with n = 3, the function recursively adds n to the result of mystery_function(n + 1) until n reaches 10, at which point it returns 0 and the recursion stops.
n = 3 සමඟින් ඇමතූ විට, ශ්‍රිතය mystery_function(n + 1) හි ප්‍රතිඵලයට n, 10 වෙත ළඟා වන තෙක් ප්‍රත්‍යාවර්තිව n එකතු කරයි, එම අවස්ථාවේදී එය 0 නැවත පැමිණ ප්‍රතිවර්තනය නතර වේ.

The function then returns the sum of all values from 3 to 9, which is `3 + 4 + 5 + 6 + 7 + 8 + 9=42`.
ශ්‍රිතය පසුව 3 සිට 9 දක්වා සියලු අගයන්හි එකතුව ලබා දෙයි, එය `3 + 4 + 5 + 6 + 7 + 8 + 9=42`

Therefore, the output of the program is 4) `42`.
එබැවින්, වැඩසටහනේ ප්‍රතිදානය 4) `42` වේ.

10. What will happen when the following Python code runs?
පහත Python කේතය ක්‍රියාත්මක වූ විට කුමක් සිදුවේද?

```
def add_items(item, lst=[]):
    lst.append(item)
    return lst

print(add_items(2, 3))
```

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

Answer: 5)

The function `add_items` takes two parameters: `item` and an optional list `lst` with a default empty list. Inside, it adds the item to lst using the `append()` method and returns the updated list.
`add_items` ශ්‍රිතය පරාමිති දෙකක් ගනී: `item` සහ පෙරනිමි හිස් ලැයිස්තුවක් සහිත `lst` යන විකල්ප ලැයිස්තුවක්. ඇතුළත, එය `append()` ක්‍රමය භාවිතයෙන් item එක lst වෙත එකතු කර යාවත්කාලීන කළ ලැයිස්තුව ආපසු ලබා දෙයි.

When you call `add_items(2, 3)`, you pass 2 as item and 3 as lst. But 3 is an integer, not a list. Integers do not have an `append()` method, so the function tries to do 3.append(2), which causes an error.
ඔබ `add_items(2, 3)` අමතන විට, ඔබ item ලෙස 2 සහ lst ලෙස 3 ලබා දෙයි. නමුත් 3 යනු ලැයිස්තුවක් නොවේ. නිඛිල වලට `append()` ක්‍රමයක් නොමැත, එබැවින් ශ්‍රිතය 3.append(2) කිරීමට උත්සාහ කරයි, එය දෝෂයක් ඇති කරයි.

This results in an `AttributeError` because Python cannot find the append method for an integer. The function expects a list, but it got a number instead.
මෙය `AttributeError` එකක් ඇති කරයි, මන්ද Python හට පූර්ණ සංඛ්‍යාවක් සඳහා append ක්‍රමය සොයාගත නොහැක. ශ්‍රිතය ලැයිස්තුවක් අපේක්ෂා කරයි, නමුත් එයට අංකයක් ලැබුණි.

To fix this, you should pass a list as the second argument, like `add_items(2, [3])`, or just call `add_items(2)` to use the default empty list.
මෙය නිවැරදි කිරීමට, ඔබ `add_items(2, [3])` වැනි දෙවන තර්කය ලෙස ලැයිස්තුවක් ලබා දිය යුතුය, නැතහොත් පෙරනිමි හිස් ලැයිස්තුව භාවිතා කිරීමට `add_items(2)` අමතන්න.

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