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 255 MARKING

2025/08/22
ictfromabc
Ravindu Bandaranayake

1. What is the primary function of the BIOS (Basic Input Output System) in a computer system?
පරිගණක පද්ධතියක BIOS (මූලික ආදාන ප්‍රතිදාන පද්ධතිය) හි මූලික කාර්යය කුමක්ද?

   1) Managing peripheral devices such as keyboards and mouse.
   යතුරුපුවරු සහ මූසික වැනි පර්යන්ත උපාංග කළමනාකරණය කිරීම.
   2) Storing user data and files / පරිශීලක දත්ත සහ ගොනු ගබඩා කිරීම.
   3) Controlling the startup process and initializing hardware components.
   ආරම්භක ක්‍රියාවලිය පාලනය කිරීම සහ දෘඩාංග සංරචක ආරම්භ කිරීම.
   4) Providing access to the Internet / අන්තර්ජාලයට ප්‍රවේශය ලබා දීම.
   5) Executing application software / යෙදුම් මෘදුකාංග ක්‍රියාත්මක කිරීම.

   Answer: 3)

   Explanation / පැහැදිලි කිරීම:

   BIOS is responsible for controlling the initial startup process of the computer, which includes initializing hardware components such as the processor, memory, storage devices, and other peripherals.
   සකසනය, මතකය, ගබඩා උපාංග සහ අනෙකුත් පර්යන්ත උපාංග වැනි දෘඩාංග සංරචක ආරම්භ කිරීම වැනි දෑ ඇතුළත් පරිගණකයේ ආරම්භක ක්‍රියාවලිය පාලනය කිරීම සඳහා BIOS වගකිව යුතුය.

   It performs Power-On Self-Test (POST) routines to check the hardware's functionality, loads the operating system from the storage device into memory, and hands over control to the operating system.
   එය දෘඩාංගවල ක්‍රියාකාරිත්වය පරීක්ෂා කිරීම සඳහා Power-On Self-Test (POST) ක්‍රියාවලිය සිදු කරයි, ගබඩා උපාංගයේ සිට මෙහෙයුම් පද්ධතිය මතකයට පූරණය කරයි, සහ මෙහෙයුම් පද්ධතියට පාලනය භාර දෙයි.

   In summary, BIOS ensures that the hardware components of the computer are properly initialized and ready for the operating system to take over.
   සාරාංශයක් ලෙස, BIOS පරිගණකයේ දෘඩාංග සංරචක නිසි ලෙස ආරම්භ කර මෙහෙයුම් පද්ධතිය භාර ගැනීමට සූදානම් බව සහතික කරයි.

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

2. Multiple levels of cache memory are used to perform tasks in modern computers.
   Out of these, most cache memory units are located in ..... .
   …… cache memory is the fastest and the highest expensive cache memory.
   නවීන පරිගණකවල කාර්යයන් ඉටු කිරීම සඳහා බහු මට්ටමේ වාරක මතකය භාවිතා වේ.
   මේ අතරින් බොහෝ වාරක මතක ඒකක …….. තුළ පිහිටා ඇත.
   ..... වාරක මතකය වේගවත්ම සහ මිල වැඩිම වාරක මතකයයි.

   1) Main memory, first level(L1) / ප්‍රධාන මතකය, පළමු මට්ටම (L1)
   2) Processor, second level(L2) / සකසනය, දෙවන මට්ටම (L2)
   3) Processor, third level(L3) / සකසනය, තෙවන මට්ටම (L3)
   4) Processor, first level(L1) / සකසනය, පළමු මට්ටම (L1)
   5) Mother board, third level(L3) / මවු පුවරුව, තුන්වන මට්ටම (L3)

   Answer: 4)

Explanation / පැහැදිලි කිරීම:

Multiple levels of cache memory are used in modern computers to improve performance by reducing the average time to access data from the main memory. The cache memory hierarchy typically consists of three levels: L1, L2, and L3.
ප්‍රධාන මතකයෙන් දත්ත වෙත ප්‍රවේශ වීමට සාමාන්‍ය කාලය අඩු කිරීමෙන් කාර්ය සාධනය වැඩි දියුණු කිරීම සඳහා නවීන පරිගණකවල බහු මට්ටමේ වාරක මතක භාවිතා වේ. වාරක මතක ධුරාවලිය සාමාන්‍යයෙන් මට්ටම් තුනකින් සමන්විත වේ: L1, L2, සහ L3.

L1 cache (Level 1 cache) is the fastest and smallest cache memory. It is embedded directly into the CPU and is divided into two parts: one for storing instructions (L1i) and one for storing data (L1d).
L1 වාරකය (Level 1 cache) යනු වේගවත්ම සහ කුඩාම වාරක මතකයයි. එය සෘජුවම CPU තුළට කාවැදී ඇති අතර එය කොටස් දෙකකට බෙදා ඇත. එකක් උපදෙස් ගබඩා කිරීම සඳහා (L1i) සහ එකක් දත්ත ගබඩා කිරීම සඳහා (L1d) වශයෙන් පවතී.

L2 cache (Level 2 cache) is larger in size and slightly slower than L1 cache. It can be located inside the CPU or on a separate chip or coprocessor. L2 cache may have a high-speed alternative system bus connecting it to the CPU.
L2 වාරකය (Level 2 cache) ප්‍රමාණයෙන් විශාල වන අතර L1 වාරකයට වඩා තරමක් මන්දගාමී වේ. එය CPU ඇතුළත හෝ වෙනම චිපයක් හෝ හර සකසනයක්(core processor) මත ස්ථානගත කළ හැක. L2 වාරකය CPU වෙත සම්බන්ධ කරන අධිවේගී විකල්ප පද්ධති බසයක් තිබිය හැක.

L3 cache (Level 3 cache) is specialized memory developed to improve the performance of L1 and L2 caches. It is usually shared between multiple cores in a multicore processor. L3 cache is typically faster than main memory (DRAM) but slower than L1 and L2 caches.
L3 වාරකය (Level 3 cache) යනු L1 සහ L2 වාරකවල ක්‍රියාකාරිත්වය වැඩි දියුණු කිරීම සඳහා නිර්මාණය කරන ලද විශේෂිත මතකයකි. එය සාමාන්‍යයෙන් බහු හර සකසනයක(multicore processor) බහු හර අතර බෙදා ගනී. L3 වාරකය සාමාන්‍යයෙන් ප්‍රධාන මතකයට (DRAM) වඩා වේගවත් නමුත් L1 සහ L2 වාරක වලට වඩා මන්දගාමී වේ.

Location of Cache Memory Units
වාරක මතක ඒකක පිහිටීම

Based on the information provided, most cache memory units are located in the microprocessor. The microprocessor contains the L1, L2, and sometimes L3 caches. The cache memory is embedded directly into the CPU chip, allowing it to operate at the same speed as the CPU.
සපයන ලද තොරතුරු මත පදනම්ව, බොහෝ වාරක මතක ඒකක ක්ෂුද්‍ර සකසනයේ පිහිටා ඇත. ක්ෂුද්‍ර සකසනයේ L1, L2 සහ සමහර විට L3 වාරක අඩංගු වේ. වාරක මතකය කෙළින්ම CPU චිපය තුළට කාවැදී ඇති අතර, එය CPU හා සමාන වේගයකින් ක්‍රියා කිරීමට ඉඩ සලසයි.

Therefore, the correct answer is option 4) Microprocessor, first level(L1).
එබැවින්, නිවැරදි පිළිතුර 4) සකසනය, පළමු මට්ටම (L1) වේ.

3. Convert 152_{10} into BCD.
152_{10} , BCD බවට පරිවර්තනය කරන්න.

1) 000101101000_{BCD}
2) 000010010111_{BCD}
3) $\underline{000101010010}_{BCD}$
4) 010001101110_{BCD}
5) 100110010010_{BCD}

Answer: 3)

Explanation / පැහැදිලි කිරීම:

BCD takes 4 bits to represent one character. So here each digit should be written by 4 bits.
BCD මගින් එක් අනුලක්ෂණයක් නිරූපණය කිරීම සඳහා බිටු 4ක් වැයවේ. එමනිසා මෙහි එක් එක් ඉලක්කම බිටු 4ක් මගින් ලියාගත යුතුය.

152_{10}		
1	5	2
0001	0101	0010
000101010010_{BCD}		

So the answer is 3.
එමනිසා පිළිතුර 3 වේ.

4. $P = 10111001_2$
$Q = 11101110_2$

If $R = 01010111_2$ and $S = 10101000_2$, what is the most suitable answer to fill in the blanks given below?
$R = 01010111_2$ සහ $S = 10101000_2$ නම්, පහත දක්වා ඇති හිස් තැන් පිරවීමට වඩාත් සුදුසු පිළිතුර කුමක්ද?

R = P …❶…Q S = P …❷…Q

1) ❶ - AND, ❷ - OR
2) ❶ - XOR, ❷ - OR
3) ❶ - AND, ❷ - XOR
4) ❶ - XOR, ❷ - XNOR
5) ❶ - NOT, ❷ - AND

Answer: 4)

Explanation / පැහැදිලි කිරීම:

Bitwise AND	P	1 0 1 1 1 0 0 1
	Q	1 1 1 0 1 1 1 0
Result		1 0 1 0 1 0 0 0

Bitwise OR	P	1 0 1 1 1 0 0 1
	Q	1 1 1 0 1 1 1 0
Result		1 1 1 1 1 1 1 1

Bitwise XOR	P	1 0 1 1 1 0 0 1
	Q	1 1 1 0 1 1 1 0
Result		0 1 0 1 0 1 1 1

Bitwise XNOR	P	1 0 1 1 1 0 0 1
	Q	1 1 1 0 1 1 1 0
Result		1 0 1 0 1 0 0 0

So, we can clearly see that R and S are obtained by performing bitwise XOR and XNOR on P and Q respectively.
එබැවින්, R සහ S ලබාගෙන ඇත්තේ P සහ Q මත පිළිවෙළින් XOR සහ XNOR යන බිටු අනුසාරිත මෙහෙයුම් සිදු කිරීමෙන් බව අපට පැහැදිලිය.

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

5. Which of the following represents the most simplified expression of $A(A \oplus B)C + A\overline{B}C$?
පහත සඳහන් ප්‍රකාශ වලින් කවරක් $A(A \oplus B)C + A\overline{B}C$ හි වඩාත් සරල කළ ප්‍රකාශනය නියෝජනය කරයිද?

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

Answer: 1)

Explanation / පැහැදිලි කිරීම:

$= A(A \oplus B)C + A\overline{B}C$	
$= A(\overline{A}B + A\overline{B})C + A\overline{B}C$	$A \oplus B = \overline{A}B + A\overline{B}$
$= (A\overline{A}B + AA\overline{B})C + A\overline{B}C$	Distributive Law / **විෂටන න්‍යාය**
$= (A\overline{A}B + A\overline{B})C + A\overline{B}C$	Idempotent Law / **තද්දේවභාවී න්‍යාය**
$= (0B + A\overline{B})C + A\overline{B}C$	Inverse Law / **ප්‍රතිලෝම න්‍යාය**
$= (0 + A\overline{B})C + A\overline{B}C$	Identity Law / **සර්වසාම්‍ය න්‍යාය**
$= (A\overline{B})C + A\overline{B}C$	Identity Law / **සර්වසාම්‍ය න්‍යාය**
$= A\overline{B}C + A\overline{B}C$	Associative Law / **සංෂටන න්‍යාය**
$= A\overline{B}C$	Idempotent Law / **තද්දේවභාවී න්‍යාය**

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

6. What is the output when these two boolean expressions are fed into an OR gate ?
OR ද්වාරයක් සඳහා ආදාන ලෙස මෙම බූලීය ප්‍රකාශන දෙක ලබා දුන් විට ප්‍රතිදානය කුමක් වේද?

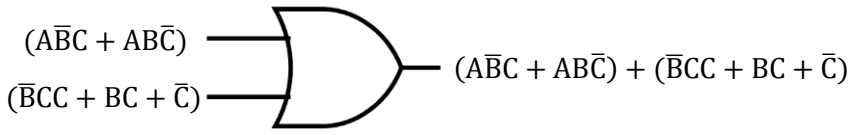
$(A\overline{B}C + AB\overline{C})$
$(\overline{B}CC + BC + \overline{C})$

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

Answer: 4)

Explanation / පැහැදිලි කිරීම:

This is solved as follows. / **මෙය පහත පරිදි විසඳනු ලැබේ.**



$A\overline{B}C + AB\overline{C} + \overline{B}CC + BC + \overline{C}$	
$A\overline{B}C + AB\overline{C} + \overline{B}C + BC + \overline{C}$	Idempotent Law / **තද්දේවභාවී න්‍යාය**
$AB\overline{C} + \overline{B}C + BC + \overline{C}$	Redundancy Law / **සමතිරික්ත න්‍යාය**
$\overline{B}C + BC + \overline{C}$	Redundancy Law / **සමතිරික්ත න්‍යාය**
$\overline{B} + BC + \overline{C}$	Redundancy Law / **සමතිරික්ත න්‍යාය**
$\overline{B} + C + \overline{C}$	Redundancy Law / **සමතිරික්ත න්‍යාය**
$\overline{B} + 1$	Complement Law / **ප්‍රතිලෝම න්‍යාය**
1	Identity Law / **සර්වසාම්‍ය න්‍යාය**

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

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

```
list_a = [10, 20, 30, 40]
list_b = [1, 2, 3, 4]
i = 0
result = []

while i < len(list_a):
    if i % 2 == 0:
        result.append(list_a[i] * list_b[i])
    else:
        result.append(list_a[i] + list_b[i])
    i += 1

print(result)
```

1) `[10, 44, 22, 90]`
2) `[10, 22, 90, 40]`
3) `[10, 22, 60, 44]`
4) `[10, 22, 90, 44]`
5) `[20, 22, 30, 44]`

Answer: 4)

Explanation / පැහැදිලි කිරීම:

The code initializes two lists, `list_a` and `list_b`, and an empty list `result`. It iterates over the indices of `list_a` with a `while loop`. For even indices, it multiplies the corresponding elements from `list_a` and `list_b` and appends the result to `result`. For odd indices, it adds the corresponding elements instead. This process results in `result` being `[10, 22, 90, 44]`, which is printed at the end.

කේතය `list_a` සහ `list_b` ලැයිස්තු දෙකක් සහ හිස් `result` ලෙසින් ලැයිස්තුවක් ආරම්භ කරයි. එය `while` ලූපයක් සමඟ `list_a` හි දර්ශක හරහා පුනරාවර්තනය වේ. ඉරට්ටේ දර්ශක සඳහා, එය `list_a` සහ `list_b` වෙතින් අනුරූප මූලාංග ගුණ කර ප්‍රතිඵලය `result` වෙත එකතු කරයි. ඔත්තේ දර්ශක සඳහා, එය වෙනුවට අනුරූප මූලාංග එකතු කරයි. මෙම ක්‍රියාවලිය අවසානයේ මුද්‍රණය වන `[10, 22, 90, 44]` ලෙස `result` ඇති කරයි.

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

8. Consider the following Python code.
පහත පයිතන් කේතය සලකා බලන්න.

```
numbers = [1, 2, 3, 4, 5]
index = 0
result = []

while index < len(numbers):
    if numbers[index] % 2 == 0:
        result.append(numbers[index] ** 2)
    else:
        result.append(numbers[index] ** 3)
    index += 1

print(result)
```

What will be the output of the above code?
ඉහත කේතයේ ප්‍රතිදානය කුමක් වේවිද?

1) [1, 4, 9, 16, 25]
2) [1, 4, 27, 16, 125]
3) [1, 4, 9, 16, 125]
4) [1, 4, 27, 16, 25]
5) [1, 8, 27, 64, 125]

Answer: 2)

Explanation / පැහැදිලි කිරීම:

The provided Python code iterates over the numbers list, checking if each number is even or odd. If a number is even, its square is appended to the `result` list; otherwise, its cube is appended. The final `result` list contains the squares of even numbers and the cubes of odd numbers.
සපයා ඇති පයිතන් කේතය සංඛ්‍යා ලැයිස්තුව මත පුනරාවර්තනය වන අතර, එක් එක් සංඛ්‍යාව ඉරට්ටේ ද ඔත්තේ ද යන්න පරීක්ෂා කරයි. සංඛ්‍යාවක් ඉරට්ටේ නම්, එහි වර්ගය ප්‍රතිඵල ලැයිස්තුවට එකතු කෙරේ එසේ නොමැති නම්, එහි ඝනකය එකතු කෙරේ. අවසාන ප්‍රතිඵල ලැයිස්තුවේ ඉරට්ටේ සංඛ්‍යා වර්ග සහ ඔත්තේ සංඛ්‍යා ඝනක අඩංගු වේ.

The correct output is `[1, 4, 27, 16, 125]`.
නිවැරදි ප්‍රතිදානය `[1, 4, 27, 16, 125]` වේ.

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

9. What will be the output of this Python code?
මෙම පයිතන් කේතයේ ප්‍රතිදානය කුමක් වේවිද?

```
x = " * "
for i in range(3):
      print(x.strip() * 5)
```

1) xxxxx
   xxxxx
   xxxxx

2) *****
   *****
   *****

3) * * * * *
   * * * * *
   * * * * *

4) x x x x x
   x x x x x
   x x x x x

5) syntax error

Answer: 2)

Explanation / පැහැදිලි කිරීම:

Answer is 2). This code stores an asterisk with spaces on left and right in a variable called x and then a for loop is created by using the range function and the value 3 is passed to the range function, this will make the for loop run 3 times as the range function creates a list from 0-2, next inside the for loop a print function is used, inside the print function they have used the .strip method on the variable x what this does is that it will eliminate all the white spaces stored in the string inside x this means there will be no spaces before and after the asterisk mark, and this edited string is multiplied by 5 which will make the code print the asterisk mark without spaces 5 times in each iteration, therefore the whole output of the code will be 3 lines of 5 asterisk marks without spaces.
පිළිතුර 2) වේ. මෙම කේතය x නම් විචල්‍යයක වම සහ දකුණේ හිස්තැන් සහිත තරු ලකුණක් ගබඩා කරයි, පසුව පරාස ශ්‍රිතය භාවිතයෙන් for loop එකක් සාදන අතර 3 අගය පරාසය ශ්‍රිතයට ලබා දෙයි, මෙය for loop ධාවනය 3 වරට පත් කරයි. පරාසයේ ශ්‍රිතය 0-2 දක්වා ලැයිස්තුවක් සාදන විට, පසුව for loop එක තුළ මුද්‍රණ ශ්‍රිතයක් භාවිතා වේ, print ශ්‍රිතය තුළ x විචල්‍යයේ .strip ක්‍රමය භාවිතා කර ඇත්තේ x හි අනවශ්‍ය හිස්තැන් සියල්ල ඉවත් කිරීමටයි. මෙයින් අදහස් වන්නේ තරු ලකුණට පෙර සහ පසු හිස් ඉඩක් නොමැති බවයි කේතයේ හිස්තැන් නොමැතිව තරු ලකුණු 5 ක පේළි 3 ක් දක්වනු ඇත.

10. What will be the output if `x = -2` and `y = 5` ?
`x = -2` සහ `y = 5` නම් ප්‍රතිදානය කුමක් වේවිද?

```
if x > 0:
  if y > 0:
    print("Both x and y are positive")
  else:
    print("x is positive, y is not")
else:
  print("x is not positive")
```

1) `Both x and y are positive`
2) `x is positive, y is not`
3) `x is not positive`
4) An error will occur / දෝෂයක් ඇතිවේ.
5) The code will run indefinitely / කේතය අවසානයක් නොමැතිව ක්‍රියාත්මක වේ.

Answer: 3)

Explanation / පැහැදිලි කිරීම:

If `x = -2` and `y = 5`, the code first checks if `x > 0`. Since `x` is `-2`, which is not greater than `0`, this condition is false.
`x = -2` සහ `y = 5` නම්, කේතය පළමුව පරීක්ෂා කරන්නේ `x > 0` ද යන්නයි. `x` යනු `-2` වේ. එය `0` ට වඩා වැඩි නොවන බැවින්, මෙම කොන්දේසිය අසත්‍ය වේ.

Therefore, the code skips the `nested if` statement and directly executes the `else` block, printing `"x is not positive"`.
එම නිසා, කේතය `nested if` ප්‍රකාශය මඟ හරින අතර සෘජුවම `else` ක්‍රියාත්මක කරයි, `"x is not positive"` මුද්‍රණය කරයි.

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