සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved]

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

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 274 MARKING

2025/09/10
ictfromabc
Ravindu Bandaranayake

1. Select the true statement/s.
සත්‍ය ප්‍රකාශ/ය තෝරන්න.

A. First generation computers often failed to operate efficiently due to vacuum tubes.
බොහෝ විට රික්තක නළ හේතුවෙන් පළමු පරම්පරාවේ පරිගණකවලට කාර්යක්ෂමව ක්‍රියා කිරීමට නොහැකි විය.

B. Second-generation computers were built on simple batch processing system concept.
දෙවන පරම්පරාවේ පරිගණක සරල කාණ්ඩ සැකසුම් පද්ධති සංකල්පය මත ගොඩනගා ඇත.

C. UNIVAC stands for Universal Arithmetic Computer.
UNIVAC යනු Universal Arithmetic Computer වේ.

D. Computers like EDSAC and EDVAC were made up of semiconductor transistors
EDSAC සහ EDVAC වැනි පරිගණක අර්ධ සන්නායක ට්‍රාන්සිස්ටර වලින් සමන්විත විය

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

Answer: 3)

Let's consider each statement one by one.
එක් එක් ප්‍රකාශය එකින් එක සලකා බලමු.

Statement / ප්‍රකාශය A)
This statement is true. First-generation computers (1940s–1950s) used vacuum tubes, which were large, generated excessive heat, and were prone to frequent failures. These issues affected the efficiency and reliability of early computers.
මෙම ප්‍රකාශය සත්‍ය වේ. පළමු පරම්පරාවේ පරිගණක (1940s–1950s) විශාල, අධික තාපයක් ජනනය කරන ලද, සහ නිතර නිතර අසාර්ථක වීමට ඉඩ ඇති රික්තක නල භාවිතා කරන ලදී. මෙම ගැටළු මුල් පරිගණකවල කාර්යක්ෂමතාව සහ විශ්වසනීයත්වය කෙරෙහි බලපෑවේය.

Statement / ප්‍රකාශය B)
This statement is true. Second-generation computers (1950s–1960s) primarily used transistors and were known for using batch processing systems, where jobs were processed in batches without user interaction, improving efficiency over first-generation computers.
මෙම ප්‍රකාශය සත්‍ය වේ. දෙවන පරම්පරාවේ පරිගණක (1950s–1960s) ප්‍රධාන වශයෙන් ට්‍රාන්සිස්ටර භාවිතා කරන අතර කණ්ඩායම් සැකසුම් පද්ධති භාවිතා කිරීම සඳහා ප්‍රසිද්ධ විය, එහිදී පරිශීලක අන්තර්ක්‍රියාකාරිත්වයකින් තොරව කණ්ඩායම් වශයෙන් රැකියා සකසන ලද අතර පළමු පරම්පරාවේ පරිගණකවලට වඩා කාර්යක්ෂමතාව වැඩි කරයි.

Statement / ප්‍රකාශය C)
This statement is false. UNIVAC actually stands for Universal Automatic Computer, not Universal Arithmetic Computer. It was one of the first commercial computers made in the early 1950s. මෙම ප්‍රකාශය අසත්‍ය වේ. UNIVAC යන්නෙන් අදහස් කරන්නේ Universal Automatic Computer මිස Universal Arithmetic Computer නොවේ. එය 1950 මුල් භාගයේ නිපදවන ලද පළමු වාණිජ පරිගණකවලින් එකකි.

Statement / ප්‍රකාශය D)
This statement is false. EDSAC and EDVAC were early computers that used vacuum tubes, not semiconductor transistors. Transistors were used in later computers, starting with the second generation. මෙම ප්‍රකාශය අසත්‍ය වේ. EDSAC සහ EDVAC යනු අර්ධ සන්නායක ට්‍රාන්සිස්ටර නොව වැකුම් ටියුබ් භාවිතා කළ මුල් පරිගණක ය. දෙවන පරම්පරාවෙන් පටන් ගෙන පසුකාලීන පරිගණකවල ට්‍රාන්සිස්ටර භාවිතා විය.

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

2. Which of the following is correct about Grid computing ?
ජාලගත පරිගණක සම්බන්ධයෙන් පහත සඳහන් දේවලින් නිවැරදි වන්නේ කුමක්ද?

1) Grid computing is a set of computers that working together to perform a specific task.
නිශ්චිත කාර්යයක් ඉටු කිරීම සඳහා එකට වැඩ කරන පරිගණක කට්ටලයක් වේ.
2) Its process is quick.
එහි ක්‍රියාවලිය වේගවත් වේ
3) Has ability to process large quantities of data by decentralizing data.
දත්ත විමධ්‍යගත කිරීමෙන් විශාල දත්ත ප්‍රමාණයක් සැකසීමේ හැකියාව ඇත
4) Has ability to operate a single software from various locations.
විවිධ ස්ථාන වලින් තනි මෘදුකාංගයක් ක්‍රියාත්මක කිරීමේ හැකියාව ඇත
5) All are correct.
සියල්ල නිවැරදියි වේ.

Answer: 5)

Explanation / පැහැදිලි කිරීම:

Grid computing is a set of computers working together to perform a specific task.
පරිගණක කිහිපයක් නිශ්චිත කාර්යයක් ඉටු කර ගැනීම සඳහා එකට වැඩ කිරීම, ජාල පරිගණනය ලෙස හැඳින්වේ

The workload is divided among the computers and each computer has its own contribution toward the common goal.
කාර්ය භාරය පරිගණක අතර බෙදී ඇති අතර සෑම පරිගණකයකටම පොදු ඉලක්කයක් කරා තම දායකත්වය ඇත.

Its advantages are as follows.
එහි වාසි පහත පරිදි වේ

- Process is quick.
ක්‍රියාවලිය ඉක්මන් වේ
- Ability to process large quantities of data by decentralizing data.
දත්ත විමධ්‍යගත කිරීමෙන් විශාල දත්ත ප්‍රමාණයක් සැකසීමේ හැකියාව
- Ability to operate a single software from various locations.
විවිධ ස්ථාන වලින් තනි මෘදුකාංගයක් ක්‍රියාත්මක කිරීමේ හැකියාව

3. Answer 2's complement of the value obtained when the following statement is trivialized?
පහත ප්‍රකාශය සුළු කල විට ලැබෙන අගයේ පිලිතුර 2හි අනුපූරකය ඇසුරෙන් කුමක්වේද?

$53_{10} - 1010_2 =$ …………

1) 00101111_2 2) 11010111_2 3) 01010101_2 4) 11010100_2 5) 00101011_2

Answer: 5)

Explanation / පැහැදිලි කිරීම:

To find the two's complement of a binary number, we need to perform the following steps:
ද්විමය අංකයක දෙකේ අනුපූරකය සොයා ගැනීමට, අපි පහත පියවරයන් සිදු කළ යුතුය:

Convert the number to its one's complement by flipping all the bits (0s become 1s and 1s become 0s).
Add 1 to the one's complement to obtain the two's complement.
සියලුම බිටු පෙරළීමෙන් අංකය එහි අනුපූරකයට පරිවර්තනය කරන්න (0 ව 1 බවටත් සහ 1 0 ව බවටත් පත් වේ) දෙකේ අනුපූරකය ලබා ගැනීම සඳහා 1හි අනුපූරකයට 1ක් එකතු කරන්න.

Let's solve the subtraction first and then find its 2's complement.
අපි මුලින්ම අඩු කිරීම විසඳා පසුව එහි 2හි අනුපූරකය සොයා ගනිමු.

Subtraction:
අඩු කිරීම:

Convert 53_{10} to binary: $53_{10} = 32 + 16 + 4 + 1 = 00110101_2$
53_{10} ද්විමය බවට පරිවර්තනය කරන්න: $53_{10} = 32 + 16 + 4 + 1 = 00110101_2$

Perform the subtraction:
අඩු කිරීම සිදු කරන්න:

```
  00110101
- 00001010
  00101011
```

The 2's complement of a positive value is equal to its binary form. So, the answer is 101011_2.
ධන අගයක 2 හි අනුපූරකය එහි ද්විමය ස්වරූපයට සමාන වේ. එබැවින්, පිළිතුර 101011_2 වේ.

4. A programmer wants to send the word “HELLO” from one computer to another. The computers only understand numbers, so each letter is converted into a unique binary code before transmission. This system the programmer is using is called:
ක්‍රමලේඛකයෙකුට “HELLO” යන වචනය එක් පරිගණකයකින් තවත් පරිගණකයකට යැවීමට අවශ්‍ය වේ. පරිගණක තේරුම් ගන්නේ සංඛ්‍යා පමණී, එබැවින් සම්ප්‍රේෂණයට පෙර සෑම අකුරක්ම අද්විතීය ද්විමය කේතයක් බවට පරිවර්තනය වේ. ක්‍රමලේඛකයා භාවිතා කරන මෙම පද්ධතිය හැඳින්වෙන්නේ:

1) Unicode 2) ASCII 3) Binary Compression 4) Hexadecimal 5) Machine Learning

Answer: 2)

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

1) **Unicode**
Unicode is similar but supports more characters from multiple languages, not just basic English letters.
Unicode සමාන නමුත් මූලික ඉංග්‍රීසි අකුරු පමණක් නොව බහු භාෂාවලින් වැඩි අක්ෂර සඳහා සහය දක්වයි.

2) **ASCII**
A system that assigns a unique number (binary code) to each character. Computers use these numbers to represent and transmit text.
එක් එක් අක්ෂරයට අනන්‍ය අංකයක් (ද්විමය කේතය) පවරන පද්ධතියකි. පරිගණක මෙම සංඛ්‍යා භාවිතා කරන්නේ පෙළ නිරූපණය කිරීමට සහ සම්ප්‍රේෂණය කිරීමටයි.

3) **Binary Compression / ද්විමය සම්පීඩනය**
Binary Compression is about reducing data size, not converting letters.
ද්විමය සම්පීඩනය යනු දත්ත ප්‍රමාණය අඩු කිරීම ගැන මිස අකුරු පරිවර්තනය කිරීම නොවේ.

4) **Hexadecimal / ෂඩ් දශම**
Hexadecimal is a number system (base 16), not a text encoding.
ෂඩ් දශම යනු සංඛ්‍යා පද්ධතියකි (පාදම 16), පෙළ කේතනයක් නොවේ.

5) **Machine Learning / යන්ත්‍ර ඉගෙනීම**
Machine Learning has nothing to do with character-to-binary conversion.
යන්ත්‍ර ඉගෙනීමට අක්ෂර-ද්විමය පරිවර්තනය සමඟ කිසිදු සම්බන්ධයක් නැත.

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

5. A programmer is designing a smart home system. She realizes that turning off both the heater and the air conditioner at the same time is equivalent to “not turning on the heater OR not turning on the air conditioner.” Which Boolean law explains this equivalence?
ක්‍රමලේඛකයෙක් ස්මාර්ට් නිවාස පද්ධතියක් නිර්මාණය කරමින් සිටී. හීටරය සහ වායු සමීකරණ යන්ත්‍රය යන දෙකම එකවර ක්‍රියා විරහිත කිරීම “හීටරය සක්‍රිය නොකිරීමට හෝ වායු සමීකරණ යන්ත්‍රය සක්‍රිය නොකිරීමට සමාන බව ඇය තේරුම් ගනී. මෙම සමානතාවය පැහැදිලි කරන්නේ කුමන බූලියන් නීතියද?

1) Redundancy Law / සමතිරික්ත නීතිය
2) <u>De Morgan’s Law / ද මුවාර් ප්‍රමේය</u>
3) Complement Law / ප්‍රතිලෝම න්‍යාය
4) Distributive Law / විඝටන න්‍යාය
5) Identity Law / සර්වසාම්‍ය න්‍යාය

Answer: 2)

Explanation / පැහැදිලි කිරීම:

De Morgan’s Laws describe how NOT interacts with AND and OR in Boolean logic.
බූලියන් තර්කනයේ දී NOT, AND සහ OR සමඟ අන්තර්ක්‍රියා කරන ආකාරය ද මුවාර් න්‍යාය මගින් විස්තර කෙරේ.

In the smart home example: / ස්මාර්ට් නිවාස උදාහරණයේ:

Not turning on the heater AND air conditioner → $\overline{\text{Heater}}$ AND $\overline{\text{AC}}$
හීටරය සහ වායු සමීකරණ යන්ත්‍රය ක්‍රියාත්මක නොකිරීම → $\overline{\text{Heater}}$ AND $\overline{\text{AC}}$

Equivalent to “Not heater OR not AC” after applying De Morgan’s Law.
ද මුවාර් න්‍යාය යෙදීමෙන් පසු “හීටරය නොවේ හෝ AC නොවේම” ට සමාන වේ.

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

6. What is the output of the Python program given below?
පහත දක්වා ඇති පයිතන් වැඩසටහනේ ප්‍රතිදානය කුමක්ද?

```
x = 0
y = (x + 2) ** -1
print(x, y, sep='|', end=':')
print("Yes") if y > x else print("No") if y == x else print("Maybe")
```

1) `0|0.5:Yes`   2) `0|0.5:Maybe`   3) `0|2:Yes`   4) `0|2:No`   5) `0|1:Maybe`

Answer: 1)

Explanation / පැහැදිලි කිරීම:

```
x = 0

y = (0 + 2) ** -1 = 2 ** -1 = 0.5
```

First print → `0|0.5:` (because `sep='|'` and `end=':'`)
පළමු මුද්‍රණය → `0|0.5:` (`sep='|'` සහ `end=':'` නිසා)

Since `y > x (0.5 > 0)`, it prints "`Yes`"
Since `y > x (0.5 > 0)` නිසා, එය" `Yes`" ලෙස මුද්‍රණය කරයි.

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 පයිතන් කේතයේ ප්‍රතිදානය කුමක් වේද?

```
a = [1, 2, [3, 4]]
b = a.copy()
b[2][1] = 99
print(a)
print(b)
```

1) `[1, 2, [3, 4]]`
`[1, 2, [3, 99]]`

2) `[1, 2, [3, 99]]`
`[1, 2, [3, 99]]`

3) `[1, 2, 99]`
`[1, 2, [3, 99]]`

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

5) Error

Answer: 2)

Explanation / පැහැදිලි කිරීම:

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

b = a.copy()
```
makes a shallow copy (outer list is copied, inner list is still shared).
පිටපතක් සාදයි (පිටත ලැයිස්තුව පිටපත් කර ඇත, අභ්‍යන්තර ලැයිස්තුව තවමත් බෙදාගෙන ඇත).

```
b[2][1] = 99
```
changes the inner list (shared), so both a and b see the change.
අභ්‍යන්තර ලැයිස්තුව (බෙදාගත්) වෙනස් කරයි, එබැවින් a සහ b යන දෙදෙනාම වෙනස දකිති.

Final lists: / අවසන් ලැයිස්තු:

```
a = [1, 2, [3, 99]]
b = [1, 2, [3, 99]]
```

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?

```
x = 10
y = 20
def calculate(a, b):
    global x
    x = a + b
    y = x - b
    return y
z = calculate(x, y)
print(x, y, z)
```

1) `10 20 10`
2) `30 20 10`
3) `30 10 20`
4) `30 10 10`
5) `Error: local variable 'y' referenced before assignment`

Answer: 2)

Explanation / පැහැදිලි කිරීම:

`x = 10, y = 20`

`calculate(a, b):`

global `x` → updates (යාවත්කාලීන කිරීම්) global `x`: `x = 10 + 20 = 30`

`y = x - b = 30 - 20 = 10` → local `y`, global `y` stays `20`

Returns `y = 10` → assigned to `z`

Final values: / අවසාන අගයන්:

`x = 30` (global updated / global යාවත්කාලීන කරන ලදී)
`y = 20` (unchanged / නොවෙනස්වේ)
`z = 10`

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

9. What is the output of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
x = [1,2,3]
y = x.copy()
y.append(4)
y.pop(2)
print(y)
```

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

Answer: 3)

Let's consider the process of this python code step by step.
මෙම පයිතන් කේතයේ ක්‍රියාවලිය පියවරෙන් පියවර සලකා බලමු.

`x = [1,2,3]`
This creates a list `x` with the elements `[1, 2, 3]`.
මෙය මූලාංග `[1, 2, 3]` සමඟ `x` ලැයිස්තුවක් නිර්මාණය කරයි.

`y = x.copy()`
This creates a copy of the list `x` and assigns it to the new variable `y`. Now, `y` is a separate list with the same elements as `x`, i.e., `[1, 2, 3]`.
මෙය `x` ලැයිස්තුවේ පිටපතක් සාදා එය නව විචල්‍ය `y` වෙත පවරයි. දැන්, `y` යනු `x` ට සමාන මූලාංග සහිත වෙනම ලැයිස්තුවකි, එනම් `[1, 2, 3]`.

`y.append(4)`
This appends the value 4 to the end of the list `y`. After this operation, `y` becomes `[1, 2, 3, 4]`.
මෙය `y` ලැයිස්තුවේ අවසානයට 4 අගය එකතු කරයි. මෙම මෙහෙයුමෙන් පසුව, `y`, `[1, 2, 3, 4]` බවට පත් වේ.

`y.pop(2)`
The `pop()` method removes and returns the element at the specified index. Here, it removes the element at index 2 in the list `y`. The element at index 2 is 3. After removing it, `y` becomes `[1, 2, 4]`.
`pop()` ක්‍රමය මඟින් නිශ්චිත දර්ශකයේ ඇති මූලාංගය ඉවත් කර ආපසු ලබා දේ. මෙන්න, එය `y` ලැයිස්තුවේ 2 දර්ශකයේ මූලාංගය ඉවත් කරයි. 2 දර්ශකයෙහි ඇති මූලාංගය 3 වේ. එය ඉවත් කිරීමෙන් පසු `y`, `[1, 2, 4]` බවට පත් වේ.

`print(y)`
This prints the final list `y`, which is `[1, 2, 4]`.
මෙය අවසාන `y` ලැයිස්තුව ප්‍රතිදානය කරයි, එනම් `[1, 2, 4]`.

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

10. Which of the following statements about the Python range() function is correct?
Python range() ශ්‍රිතය පිළිබඳ පහත සඳහන් ප්‍රකාශවලින් නිවැරදි කුමක්ද?

1) `range(start, stop, step)` can generate a decreasing sequence only if step is positive.
`range(start, stop, step)` පියවර ධන නම් පමණක් අඩු වන අනුපිළිවෙලක් ජනනය කළ හැකිය.

2) The stop value in `range()` is always **included** in the generated sequence.
`range()` හි නැවතුම් අගය සැමවිටම ජනනය කරන ලද අනුපිළිවෙලට ඇතුළත් වේ.

3) Using a negative step in `range()` allows you to generate a sequence in reverse order.
`range()` හි සෘණ පියවරක් භාවිතා කිරීමෙන් ඔබට ප්‍රතිලෝම අනුපිළිවෙලින් අනුපිළිවෙලක් ජනනය කිරීමට ඉඩ සලසයි.

4) `range(5)` generates the sequence `[1, 2, 3, 4, 5]`.
`range(5)` `[1, 2, 3, 4, 5]`අනුපිළිවෙල ජනනය කරයි.

5) The `range()` function can only be used inside for loops.
`range()` ශ්‍රිතය භාවිතා කළ හැක්කේ ලූප සඳහා ඇතුළත පමණි.

Answer: 3)

Explanation / පැහැදිලි කිරීම:

`range(start, stop, step)`
Generates a sequence of numbers.
සංඛ්‍යා අනුපිළිවෙලක් ජනනය කරයි.

Negative step / සෘණ පියවර
Allows counting backwards (reverse order).
ආපසු ගණන් කිරීමට ඉඩ දෙයි (ප්‍රතිලෝම අනුපිළිවෙල).

Option 1)
A positive step cannot generate a decreasing sequence; it must be negative.
ධනාත්මක පියවරකට අඩුවන අනුපිළිවෙලක් ජනනය කළ නොහැක❑ එය සෘණ විය යුතුය.

Option 2)
The stop value is excluded in range sequences.
පරාස අනුපිළිවෙලවල් තුළ නැවතුම් අගය බැහැර කර ඇත.

Option 4)
`range(5)` generates `[0, 1, 2, 3, 4]`, not `[1, 2, 3, 4, 5]`.
`range(5)` `[0, 1, 2, 3, 4]` ජනනය කරයි, `[1, 2, 3, 4, 5]` නොවේ.

Option 5)
`range()` can be used anywhere an iterable is needed, not just in for loops.
`range()` ලූප සඳහා පමණක් නොව, පුනරාවර්තනයක් අවශ්‍ය ඕනෑම තැනක භාවිතා කළ හැක.

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

❋ ❋ ❋