සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved]

Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu

Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 286 MARKING

2025/09/22
ictfromabc
Ravindu Bandaranayake

1. Which of the following was NOT a feature of early computers like the ENIAC?
ENIAC වැනි මුල් පරිගණකවල ලක්ෂණයක් නොවන්නේ පහත ඒවායින් කුමක්ද?

   1) Compact size. සංයුක්ත ප්‍රමාණය.
   2) Use of vacuum tubes. රික්තක නළ භාවිතය.
   3) Large power consumption. විශාල බලශක්ති පරිභෝජනය.
   4) Limited programmability. සීමිත වැඩසටහන් හැකියාව.
   5) High failure rate. ඉහළ අසාර්ථක අනුපාතය

Answer : 1)

Let's explain the statements in the above question.
ඉහත ප්‍රශ්නයේ ඇති ප්‍රකාශ මෙසේ පැහැදිලි කරමු.

1) Compact size. / සංයුක්ත ප්‍රමාණය.
Early computers like the ENIAC were very large, often taking up entire rooms. They were not compact by any means.
ENIAC වැනි මුල් පරිගණක ඉතා විශාල වූ අතර බොහෝ විට සම්පූර්ණ කාමරයම ප්‍රමාණය ගන්නා ලදී. ඒවා කිසිම ආකාරයකින් සංයුක්ත නොවිය.

2) Use of vacuum tubes. / රික්තක නළ භාවිතය.
ENIAC and other early computers used vacuum tubes for circuitry, which were essential for their operation.
ENIAC සහ අනෙකුත් මුල් පරිගණක ඒවායේ ක්‍රියාකාරිත්වය සඳහා අත්‍යවශ්‍ය වූ පරිපථ සඳහා රික්තක නළ භාවිතා කරන ලදී.

3) Large power consumption. / විශාල බලශක්ති පරිභෝජනය.
Early computers consumed a significant amount of power due to the use of vacuum tubes and other components.
රික්තක නළ සහ අනෙකුත් සංරචක භාවිතය හේතුවෙන් මුල් පරිගණක සැලකිය යුතු බලයක් පරිභෝජනය කරන ලදී.

4) Limited programmability. / සීමිත වැඩසටහන් හැකියාව.
Early computers had limited programmability compared to modern computers. Programming was often done manually using switches and patch cords.
නවීන පරිගණක හා සසඳන විට මුල් පරිගණකවල ක්‍රමලේඛන හැකියාව සීමිත විය. ක්‍රමලේඛන බොහෝ විට ස්විච සහ පැච් ලණු භාවිතයෙන් අතින් සිදු කරන ලදී.

5) High failure rate. / ඉහළ අසාර්ථක අනුපාතය
The vacuum tubes used in early computers like the ENIAC had a high failure rate, leading to frequent maintenance issues.
ENIAC වැනි මුල් පරිගණකවල භාවිතා කරන ලද රික්තක නළවල ඉහළ අසාර්ථක අනුපාතයක් තිබූ අතර, එය නිරන්තර නඩත්තු ගැටළු වලට මග පාදයි.

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

2. Considering series and parallel ports, which is not a characteristic of parallel ports?
ශ්‍රේණිගත කෙවෙනි හා සමාන්තරගත කෙවෙනි සැලකූ විට සමාන්තරගත කෙවෙනි වල ලක්ෂණයක් නොවන්නේ?

1) Transmits multiple bits at once. එකවර බිටු කිහිපයක් සම්ප්‍රේෂණය කරයි.
2) More wire is required. වැඩිපුර වයර ප්‍රමාණයක් අවශ්‍ය වේ.
3) is slow. මන්දගාමී වේ.
4) A DB-25 connector is used. DB-25 සම්බන්ධක භාවිතා වේ.
5) More prone to errors due to timing issues.
කාල ගණනයේ ගැටලු හේතුවෙන් දෝෂ වලට ගොදුරු වීමේ ප්‍රවණතාව වැඩිය.

Answer : 3)

The correct answer is 3) is slow.
නිවැරදි පිළිතුර නම් 3) මන්දගාමී වේ.

Parallel circuits are typically faster than series circuits because they transmit multiple bits at once. This is a key characteristic that distinguishes parallel circuits from serial circuits, where bits are transmitted one at a time, often making serial circuits slower.
සමාන්තර පරිපථ සාමාන්‍යයෙන් ශ්‍රේණි පරිපථවලට වඩා වේගවත් වන්නේ ඒවා එකවර බිටු කිහිපයක් සම්ප්‍රේෂණය කරන බැවිනි. මෙය අනුක්‍රමික පරිපථ වලින් සමාන්තර පරිපථ වෙන්කර හඳුනා ගන්නා ප්‍රධාන ලක්ෂණයකි, එහිදී බිටු එකින් එක සම්ප්‍රේෂණය වන අතර බොහෝ විට අනුක්‍රමික පරිපථ මන්දගාමී වේ.

Other Characteristics of Parallel Circuits / සමාන්තර පරිපථවල අනෙකුත් ලක්ෂණ:

Option / විකල්පය 1)
Parallel circuits send multiple bits simultaneously, making data transfer faster.
සමාන්තර පරිපථ එකවර බිටු කිහිපයක් යවයි, දත්ත හුවමාරුව වේගවත් කරයි.

Option / විකල්පය 2)
Because each bit has its own path, parallel circuits require more wires compared to serial circuits.
සෑම බිට් එකකටම තමන්ගේම මාර්ගයක් ඇති නිසා, සමාන්තර පරිපථවලට අනුක්‍රමික පරිපථවලට සාපේක්ෂව වැඩි වයර් අවශ්‍ය වේ.

Option / විකල්පය 4)
Parallel ports, such as the DB-25 connector, were commonly used for parallel data transmission, particularly in older printer connections.
DB-25 සම්බන්ධකය වැනි සමාන්තර කෙවෙනි, සමාන්තර දත්ත සම්ප්‍රේෂණය සඳහා, විශේෂයෙන්ම පැරණි මුද්‍රණ සම්බන්ධතා සඳහා බහුලව භාවිතා විය.

Option / විකල්පය 5)
Parallel circuits can experience timing issues, leading to errors, especially over longer distances, because all bits must arrive simultaneously.
සමාන්තර පරිපථවලට කාල ගැටළු අත්විඳිය හැකිය, දෝෂ වලට තුඩු දෙයි, විශේෂයෙන් දිගු දුරක්, සියලු බිටු එකවර පැමිණිය යුතු බැවිනි.

3. The decimal transition corresponding to BA5 hexadecimal transition and the 8 bit two's complement representation corresponding to -17 are respectively,
$BA5_{16}$ ෂඩ් දශමය සංඛ්‍යාවට අනුරූප දශමය සංඛ්‍යාව හා -17 ට අදාළ බිටු 8ක දෙකෙහි අනුපූරක නිරූපණය පිළිවෙලින් වන්නේ,

1) 2881, 10101111_2
2) 2891, 11101101_2
3) 2981, 11101111_2
4) 2981, 11101110_2
5) 2981, 11110111_2

Answer : 3)

Convert Hexa decimal to decimal ෂඩ් දශමය දශමයට පරිවර්තනය											
B				A				5			
1	0	1	1	1	0	1	0	0	1	0	1
$1x2^{11}$	$0x2^{10}$	$1x2^{9}$	$1x2^{8}$	$1x2^{7}$	$0x2^{6}$	$1x2^{5}$	$0x2^{4}$	$0x2^{3}$	$1x2^{2}$	$0x2^{1}$	$1x2^{0}$
2048	0	512	256	128	0	32	0	0	4	0	1
2048 + 512 + 256 + 128 + 32 + 4 + 1 = 2981											

First convert 17 to binary number.
ප්‍රථමයෙන් 17, ද්විමය සංඛ්‍යාවක් බවට පරිවර්තනය කරයි.

00010001_2

Now represent 0 and 1 interchangeably here (one's complement).
දැන් මෙහි 0 හා 1 හුවමාරු කර නිරූපණය කරයි.(එකෙහි අනුපූරකය)

11101110_2

Now 1 is added to this value. (Two’s complement)
දැන් මෙම අගයට 1ක් එකතු කරනු ලැබේ. (දෙකෙහි අනුපූරකය)

$$
\begin{array}{r}
11101110_2 \\
+ \qquad 1_2 \\
\hline
11101111_2 \\
\hline
\end{array}
$$

$-17 = 11101111_{2\text{'s com}}$

4. What will be the answer when the following Boolean expression simplified into simplest form?
පහත දැක්වෙන බූලියානු ප්‍රකාශනය සරල කළ විට පිළිතුර කුමක් වේද?

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

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

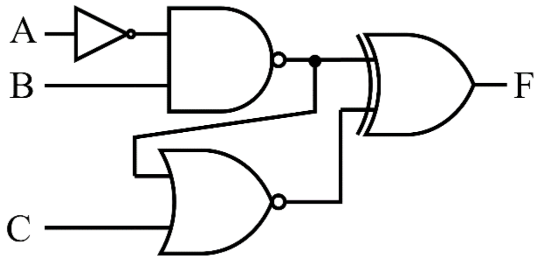
Answer : 1)

This boolean expression can be simplfied as follows.
මෙම බූලියානු ප්‍රකාශනය පහත ආකාරයට සුළු කළ හැකිය.

$\overline{(A + \overline{B}).(B + \overline{C})}$	
$\overline{A + \overline{B}} + \overline{B + \overline{C}}$	De Morgan’s law ද මුවාර් ප්‍රමේයය
$\overline{A}\overline{\overline{B}} + \overline{B + \overline{C}}$	De Morgan’s law ද මුවාර් ප්‍රමේයය
$\overline{A}B + \overline{B + \overline{C}}$	Inverse law ප්‍රතිලෝම න්‍යාය
$\overline{A}B + \overline{B}\overline{\overline{C}}$	De Morgan’s law ද මුවාර් ප්‍රමේයය
$\overline{A}B + \overline{B}C$	Inverse law ප්‍රතිලෝම න්‍යාය

Accordingly, the correct answer is number 1.
ඒ අනුව නිවැරදි පිළිතුර අංක 1 වේ.

5. What cannot A,B and C be for the output of the above circuit to be 1?
ඉහත පරිපථයේ ප්‍රතිදානය 1 වීමට A,B හා C විය නොහැක්කේ කවරක්ද



1) <u>0, 1, 1</u>   2) 1, 0, 1   3) 1, 1, 1   4) 0, 0, 0   5) 0, 0, 1

Answer : 1)

The truth table for the above circuit can be constructed as follows.
ඉහත පරිපථයට අදාල සත්‍යතා වගුව පහත පරිදි ගොඩනගා ගත හැක.

A	B	C	$\overline{A}$	$\overline{\overline{A}B}$	$\overline{\overline{\overline{A}B} + C}$	$\overline{\overline{\overline{A}B} + C} \oplus \overline{\overline{A}B}$
0	0	0	1	1	0	1
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	1	0	0	0
1	0	0	0	1	0	1
1	0	1	0	1	0	1
1	1	0	0	1	0	1
1	1	1	0	1	0	1

So, the answer is 1. / එබැවින්, පිළිතුර 1 වේ.

6. What is the output of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් ද?

```
x = {}
x['1'] = (3, 4)
x['2'] = (4, 3)
for i in x.keys():
    print(x[i][-1])
```

1) `(3, 4)` `(4, 3)`   2) `4, 3`   3) `3` `4`   4) `4` <u>`3`</u>   5) `syntax error`

Answer : 4)

```
x = {}
```

This line creates an empty dictionary `x`.
මෙම ජේළිය `x` ලෙස හිස් dictionary එකක් නිර්මාණය කරයි.

```
x['1'] = (3, 4)
x['2'] = (4, 3)
```

Here, two key-value pairs are added to the dictionary `x`.
මෙහිදී, `x` dictionary එකට යතුරු-අගය යුගල දෙකක් එකතු කරනු ලැබේ.

Key '1' is associated with the tuple `(3, 4)`.
යතුර '1' tuple `(3, 4)` සමඟ සම්බන්ධ වේ.

Key '2' is associated with the tuple `(4, 3)`.
යතුර '2' tuple `(4, 3)` සමඟ සම්බන්ධ වේ.

`for i in x.keys():`
This line starts a for loop that iterates over all the keys in the dictionary x. The variable `i` will take each key's value during the loop.
මෙම පේළිය x dictionary එකෙහි ඇති සියලුම යතුරු මත පුනරාවර්තනය වන for loop එකක් ආරම්භ කරයි. විචල්‍යය `i` ලූපය අතරතුර එක් එක් යතුරේ අගය ගනී.

`print(x[i][-1])`
`x[i]` retrieves the tuple associated with the current key `i`.
`x[i]` වත්මන් යතුර ස සමඟ සම්බන්ධිත tuple ලබා ගනී.

`x[i][-1]` accesses the last element of that tuple (since -1 is used as an index to get the last item).
`x[i][-1]` එම tuple හි අවසාන මූලාංගය වෙත ප්‍රවේශ වේ (අවසාන අයිතමය ලබා ගැනීම සඳහා -1 දර්ශකයක් ලෙස භාවිතා කරන බැවින්).

`print()` outputs this last element to the console.
`print()` මෙම අවසාන මූලාංගය කොන්සෝලය වෙත ප්‍රතිදානය කරයි.

For key '1', the tuple is `(3, 4)`, so `x['1'][-1]` is 4.
යතුර '1' සඳහා, tuple `(3, 4)`, එබැවින් `x['1'][-1]`, `4` වේ.

For key '2', the tuple is `(4, 3)`, so `x['2'][-1]` is 3.
යතුර '2' සඳහා, tuple `(4, 3)`, එබැවින් `['2'][-1]`, `3` වේ.

Therefore, the output of the code will be,
එබැවින්, කේතයේ ප්‍රතිදානය වනුයේ,

```
4
3
```

So, the correct answer number is 4.
එමනිසා, නිවැරදි පිළිතුරු අංකය 4 වේ.

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

```
string1 = "hello"
string2 = string1
string1 += " world"
print(string2)
```

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

1) `hello`   2) `hello world`   3) `world`   4) `world hello`   5) invalid output

Answer : 1)

Let’s take a look at the above Python code.
අපි ඉහත Python කේතය දෙස බලමු.

`string1 = "hello"`
Creates a string with the value "hello" and assigns it to the variable “string1”.
"hello" අගය string දත්ත ප්‍රරූපය යටතේ “string1” විචල්‍යයට පවරයි.

`string2 = string1`
This assigns the value of “string1” (which is "hello") to “string2”. At this point, both “string1” and “string2” hold the value "hello".
“string1” ("hello”) හි අගය “string2” වෙත පවරයි. මෙම අවස්ථාවේදී “string1” සහ “string2” යන දෙකම "hello" අගය රඳවා තබා ගනී.

```
string1 += " world"
```

This line concatenates " world" to “string1” , so now “string1” becomes "hello world". However, “string2” still holds the original value "hello" because strings are immutable in Python, and this operation creates a new string rather than modifying the original one.
මෙම පේළිය " world", “string1” වෙත සංකලනය කරයි, එබැවින් දැන් “string1”, "hello world" බවට පත්වේ. කෙසේ වෙතත්, “string2” තවමත් මුල් අගය "hello" දරයි. පයිතන් හි string අගයන් වෙනස් කළ නොහැකි නිසා, මෙමගින් මුල් එක වෙනස් නොකර නව එකක් නිර්මාණය කරයි.

```
print(string2)
```

This prints the value of “string2”, which is still "hello"
මෙය තවමත් "hello" වන“string2” අගය මුද්‍රණය කරයි

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

8.

```
planets = ["   Mars   ", "  Venus  ", "  Jupiter  ", "  Saturn  "]
cleaned_planets = [planet.strip() for planet in planets]
print(planets)
print(cleaned_planets)
```

Given the above Python code, what does the `cleaned_planets` list contain after execution?
ඉහත පයිතන් කේතය ක්‍රියාත්මක කිරීමෙන් පසු `cleaned_planets` ලැයිස්තුවේ අඩංගු වන්නේ කුමක්ද?

1) `["Mars", "Venus", "Jupiter", "Saturn"]`
2) `[" Mars ", " Venus ", " Jupiter ", " Saturn "]`
3) `["Mars ", "Venus ", "Jupiter ", "Saturn "]`
4) `[" Mars ", "Venus", "Jupiter", " Saturn "]`
5) `[" Mars", " Venus", " Jupiter", " Saturn"]`

Answer : 1)

In the given Python code, the **`cleaned_planets`** list is created by removing extra spaces from the beginning and end of each planet name in the **`planets`** list. The **`strip()`** method is used to do this for each planet. As a result, the **`cleaned_planets`** list contains the names **`"Mars", "Venus", "Jupiter"`** and **`"Saturn"`** without any extra spaces.
ලබා දී ඇති පයිතන් කේතයේ, **`planets`** ලැයිස්තුවේ එක් එක් ග්‍රහලෝක නාමයේ ආරම්භයේ සහ අවසානයේ අමතර හිස්තැන් ඉවත් කිරීමෙන් **`cleaned_planets`** යන ලැයිස්තුව නිර්මාණය වේ. එක් එක් ග්‍රහලෝක සඳහා මෙය සිදු කිරීම සඳහා **`strip()`** ක්‍රමය භාවිතා කරයි. එහි ප්‍රතිඵලයක් වශයෙන්, **`cleaned_planets`** ලැයිස්තුවේ **`"Mars", "Venus", "Jupiter"`** සහ **`"Saturn"`** නම් අමතර හිස්තැන් නොමැතිව ලබා දේ.

`["Mars", "Venus", "Jupiter", "Saturn"]`

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

9. What is the output of the following Python code? / පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් ද?

```
x = 5
y = 0
while x > 0:
     if y % 2 == 0:
      y += x
    else:
      y -= x
    x -= 1
print (y)
```

1) 1 2) 6 3) -5 4) -6 5) 0

Answer : 1)

The Python code provided uses a while loop to manipulate the variables x and y. Initially, x is 5 and y is 0. The loop continues while x is greater than 0. Inside the loop, the code checks if y is even or odd. If `y` is even, y is incremented by x; if y is odd, y is decremented by x. After each iteration, x is decremented by 1. The sequence of operations results in the following changes to y: 0 (even) + 5 = 5, 5 (odd) - 4 = 1, 1 (odd) - 3 = -2, -2 (even) + 2 = 0, and 0 (even) + 1 = 1. Therefore, the final output of the code is 1.
සපයා ඇති පයිතන් කේතය x සහ y විචල්‍යයන් හැසිරවීමට while ලූපයක් භාවිතා කරයි. මුලදී, x යනු 5 වන අතර y යනු 0 වේ. ලූපය දිගටම පවතින අතර x විචල්‍ය 0 ට වඩා වැඩි වේ. ලූපය තුළ, කේතය y ඉරට්ටේ හෝ ඔත්තේ ද යන්න පරීක්ෂා කරයි. y ඉරට්ටේ නම්, y x මගින් වැඩි වේ. y ඔත්තේ නම්, y x මගින් අඩු කරනු ලැබේ. එක් එක් පුනරාවර්තනයෙන් පසුව, x විචල්‍ය 1 කින් අඩු කරනු ලැබේ. මෙහෙයුම් අනුපිළිවෙලෙහි ප්‍රතිඵලය y: 0 (even) + 5 = 5, 5 (odd) - 4 = 1, 1 (odd) - 3 = -2, -2 (even) + 2 = 0, සහ 0 (even) + 1 = 1. එබැවින්, කේතයේ අවසාන ප්‍රතිදානය 1 වේ.

So, the correct answer is 1).
ඒබැවින්, නිවැරදි පිළිතුර 1) වේ.

10. What is the output of the below Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
x=0
while x<10:
    if x%2==0 or x%3==2:
        print(x, end='')
    x=x+1
```

1) 024568 2) 0124678 3) 0123456789 4) 013579 5) 02468

Answer : 1)

Let's consider the explanation of the above python code one by one.
ඉහත පයිතන් කේත රේඛාවේ පැහැදිලි කිරීම එකින් එක සලකා බලමු.

`x = 0`
Initially `x` is set to `0`.
මුලදී x, 0 ලෙස සකසා ඇත.

`while x < 10:`
The while loop will continue as long as `x` is less than `10`.
x 10 ට වඩා අඩු වන තාක් ලූපය දිගටම පවතිනු ඇත.

Inside the loop, the if statement checks two conditions.
ලූපය තුළ, if ප්‍රකාශය කොන්දේසි දෙකක් පරීක්ෂා කරයි

- **`x % 2 == 0:`**
  Checks if `x` is even (divisible by 2 with no remainder).
  x ඉරට්ටේ දැයි පරීක්ෂා කරයි (ඉතිරි නොමැතිව 2 න් බෙදිය හැකිය).
- **`x % 3 == 2:`**
  Checks if `x` gives a remainder of 2 when divided by 3.
  3 න් බෙදූ විට x, ඉතිරියක් ලෙස 2 ලබා දෙන්නේ දැයි පරීක්ෂා කරයි.

If either condition is `True`, `x` is printed without a newline (due to `end=''`).
එක් කොන්දේසියක් වත් `True` නම්, x නව රේඛාවක් නොමැතිව මුද්‍රණය වේ (`end=''` නිසා).

`x` is incremented by `1` after each iteration.
එක් එක් පුනරාවර්තනයට පසුව x, 1 කින් වැඩි වේ.

x	x % 2 == 0	x % 3 == 2	Final Condition	Print
0	0 % 2 == 0 (True)	0 % 3 == 2 (False)	True	0
1	1 % 2 == 0 (False)	1 % 3 == 2 (False)	False	
2	2 % 2 == 0 (True)	2 % 3 == 2 (True)	True	2
3	3 % 2 == 0 (False)	3 % 3 == 2 (False)	False	
4	4 % 2 == 0 (True)	4 % 3 == 2 (False)	True	4
5	5 % 2 == 0 (False)	5 % 3 == 2 (True)	True	5
6	6 % 2 == 0 (True)	6 % 3 == 2 (False)	True	6
7	7 % 2 == 0 (False)	7 % 3 == 2 (False)	False	
8	8 % 2 == 0 (True)	8 % 3 == 2 (True)	True	8
9	9 % 2 == 0 (False)	9 % 3 == 2 (False)	False	

The code will output `024568`.
කේතය `024568` ලෙස ප්‍රතිදානය කරයි.

Each number is printed sequentially without spaces or newlines between them.
සෑම අංකයක්ම ඒවා අතර හිස් හෝ නව රේඛා නොමැතිව අනුපිළිවෙලින් මුද්‍රණය කරයි.

So, the correct answer is 1).
ඒ අනුව, නිවැරදි පිළිතුර 1) වේ.