සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved]

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

Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 428 MARKING

2026/02/11
ictfromabc
Ravindu Bandaranayake

1. The world's first general purpose digital computer, ENIAC was developed by the scientists …………... and ……….. in 1946.
   1946 දී ලොව ප්‍රථම පොදු කාර්ය අංකිත පරිගණකය වන ENIAC, ……………… සහ …………... යන විද්‍යාඥයින් විසින් නිපදවන ලදී.

   Which of the following is most appropriate to fill in the blanks in the above statement?
   දී ඇති ප්‍රකාශනයේ හිස්තැන් පිරවීමට පහත සඳහන් කවරක් වඩාත් සුදුසු වේ ද?

   1) William Shockley, John Bardeen / විලියම් ශොක්ලි, ජෝන් බාඩීන්
   2) John Presper Eckert, John William Mauchly / ජෝන් ප්‍රෙස්පර් එකර්ට්, ජෝන් විලියම් මෝක්ලි
   3) John Ambrose Fleming, Lee De Frost / ජෝන් ඇම්බ්‍රෝස් ෆ්ලෙමින්ග්, ලි ද ෆ්‍රොස්ට්
   4) John Vincent Atanasoff, Clifford Berry / ජෝන් වින්සන්ට් ඇටනොසොෆ්, ක්ලිෆර්ඩ් බෙරී
   5) Walter Brattain, William Shockley / වෝල්ටර් බ්‍රටේන්, විලියම් ශොක්ලි

   Answer: 2)

   The world's first general purpose digital computer, ENIAC was developed by the scientists John Presper Eckert and John William Mauchly in 1946. It used electronics. It had no moving parts.
   1946 දී ලොව ප්‍රථම පොදු කාර්ය අංකිත පරිගණකය වන ENIAC, ජෝන් ප්‍රෙස්පර් එකර්ට් සහ ජෝන් විලියම් මෞච්ලි යන විද්‍යාඥයින් විසින් නිපදවන ලදී. මෙහි ඉලෙක්ට්‍රොනික විද්‍යාව භාවිතා වූ අතර චලනය වන කිසිඳු කොටසක් නොතිබිනි.

   ENIAC was able to perform about 5000 mathematical operations per second. It contains about 18000 vacuum tubes and they were burnt frequently due to the excessive heating.
   තත්පරයකට ගණිතමය මෙහෙයුම් 5000ක් පමණ කිරීමට ENIAC සමත් විය. එහි රික්තක නල 18000 ක් පමණ අඩංගු වන අතර අධික උෂ්ණත්වය නිසා ඒවා නිතර දැවියාමට ලක්විය.

   William Shockley, John Bardeen and Walter Brattain of Bell labs invented the transistor in 1947.
   1947 දී බෙල් රසායනාගාර වලදී විලියම් ෂොක්ලි, ජෝන් බාර්ඩීන් සහ වෝල්ටර් බ්‍රැටේන් ට්‍රාන්සිස්ටරය සොයා ගන්නා ලදී.

   John Ambrose Fleming invented the two-element vacuum tube in 1904. Lee De Frost modified it and made three-element vacuum tube in 1906. This invention started the electronic field.
   ජෝන් ඇම්බ්‍රෝස් ෆ්ලෙමින් විසින් 1904 දී අග්‍ර දෙකකින් යුත් රික්තක නළය සොයා ගන්නා ලදී. ලි ද ෆ්‍රොස්ට් එය නවීකරණය කර 1906 දී අග්‍ර තුනකින් යුත් රික්තක නළය සාදන ලදී. මෙම සොයාගැනීම ඉලෙක්ට්‍රොනික ක්ෂේත්‍රය ආරම්භ කළේය.

   John Vincent Atanasoff and Clifford Berry invented the first electronic digital computer, ABC (Atanasoff Berry Computer) between 1937 – 1942. But they did not receive the patent because it was not completed.
   ජෝන් වින්සන්ට් ඇටානාසොෆ් සහ ක්ලිෆර්ඩ් බෙරී විසින් 1937 - 1942 අතර කාලය තුළ ABC (Atanasoff Berry Computer) ප්‍රථම ඉලෙක්ට්‍රොනික ඩිජිටල් පරිගණකය සොයා ගන්නා ලදී. නමුත් එය සම්පූර්ණ කර නොතිබූ නිසා ඔවුන්ට පේටන්ට් බලපත්‍රය ලැබුණේ නැත.

   Therefore the correct answer is John Presper Eckert, John William Mauchly.
   එම නිසා නිවැරදි පිළිතුර වන්නේ ජෝන් ප්‍රෙස්පර් එකර්ට්, ජෝන් විලියම් මෞච්ලි යන්නයි.

2. What is the binary equivalent of the hexadecimal number $45A1B_{16}$?
ෂඩ් දශමය $45A1B_{16}$ ට සමාන ද්විමය පිළිතුර කුමක්ද?

1) $0001001101010000101 1_2$
2) $0100010110100001101 1_2$
3) $0010010101001110110 1_2$
4) $0111000101101010011 0_2$
5) $0110100101100011011 1_2$

Answer : 2)

4	5	A	1	B	Hexadecimal
↓	↓	↓	↓	↓	
0100	0101	1010	0001	1011	Binary

01000101101000011011_2

3. What will be the simplified answer of the expression $(1C + \overline{CB})D + BC$?
$(1C + \overline{CB})D + BC$ යන ප්‍රකාශයේ සුළු කළ පිළිතුර කුමක්ද?

1) $D + BC$ 2) $D + \overline{B}C$ 3) $D + \overline{BC}$ 4) $B + BC$ 5) $D + B$

Answer : 1)

$(1C + \overline{CB})D + BC$	
$(1C + \overline{C} + \overline{B})D + BC$	De Morgan's law ද මුවාර් ප්‍රමේයය
$(C + \overline{C} + \overline{B})D + BC$	Identity law සර්වසාම්‍ය න්‍යාය
$(\overline{B} + 1)D + BC$	Complement law ප්‍රතිලෝම න්‍යාය
$1D + BC$	Identity law සර්වසාම්‍ය න්‍යාය
$D + BC$	Identity law සර්වසාම්‍ය න්‍යාය

4. Which of the following is not a feature related to simple batch systems?
සරල කාණ්ඩ පද්ධති හා සම්බන්ධ විශේෂාංගයක් නොවන්නේ පහත ඒවායින් මොනවාද?

1) Uniprogramming / ඒකීය ක්‍රමලේඛනය.
2) High response time / ඉහල ප්‍රතිචාර කාලය.
3) Processor sat idle during I/O. / ආදාන ප්‍රතිදාන අතරතුර සකසනය අලසව සිටී.
4) User directly interacted with the hardware. / පරිශීලක දෘඪාංග සමඟ සෘජු සම්බන්ධතා පවත්වයි.
5) OS loaded and executed programs in tape one at a time.
වරකට එක බැගින් චුම්බක පටියේ ඇති ක්‍රමලේඛ ප්‍රවේශ කර ක්‍රියාත්මක කරයි.

Answer : 4)

Simple Batch Systems/ සරල කාණ්ඩ පද්ධති
Simple batch systems are operating systems that execute a series of jobs in groups, known as batches, without user interaction during processing. This approach increases efficiency by allowing multiple jobs to share resources and reducing idle time for the CPU.
සරල කාණ්ඩ පද්ධති යනු පිරිසැකසුම් කිරීමේදී පරිශීලක අන්තර්ක්‍රියාකාරිත්වයකින් තොරව කණ්ඩායම් ලෙස හැඳින්වෙන, කණ්ඩායම් වශයෙන් කාර්යයන් මාලාවක් ක්‍රියාත්මක කරන මෙහෙයුම් පද්ධති වේ. මෙම ප්‍රවේශය බහු කාර්යයන් වලට සම්පත් බෙදා ගැනීමට ඉඩ දීමෙන් සහ CPU සඳහා අක්‍රිය කාලය අඩු කිරීමෙන් කාර්යක්ෂමතාව වැඩි කරයි.

Let's go through the given options. / ලබා දී ඇති විකල්ප හරහා යමු.

1) Uniprogramming / ඒකීය ක්‍රමලේඛනය.

This is a feature of simple batch systems where only one program is executed at a time. The system runs one job and then processes the next job in the queue.
මෙය වරකට එක් වැඩසටහනක් පමණක් ක්‍රියාත්මක වන සරල කණ්ඩායම් පද්ධතිවල ලක්ෂණයකි. පද්ධතිය එක් කාර්යයක් ක්‍රියාත්මක කරන අතර පසුව පෝලිමේ ඊළඟ කාර්යය සකසයි.

2) High response time. / ඉහල ප්‍රතිචාර කාලය.
Simple batch systems tend to have high response times because jobs are processed in sequence, and the user has to wait for the batch job to complete before seeing the results.
කාර්යයන් අනුපිළිවෙලින් සකසන බැවින් සරල කණ්ඩායම් පද්ධතිවලට ඉහළ ප්‍රතිචාර කාලයක් ඇත, සහ ප්‍රතිඵල දැකීමට පෙර පරිශීලකයාට කණ්ඩායම් කාර්යය සම්පූර්ණ වන තෙක් බලා සිටීමට සිදු වේ.

3) Processor sat idle during I/O / ආදාන ප්‍රතිදාන අතරතුර සකසනය අලසව සිටී.
In simple batch systems, when a program is performing I/O operations, the CPU may remain idle since it cannot switch to another job until the current job completes.
සරල කණ්ඩායම් පද්ධතිවල, වැඩසටහනක් ආදාන ප්‍රතිදාන මෙහෙයුම් සිදු කරන විට, CPU වත්මන් කාර්යය අවසන් වන තෙක් වෙනත් කාර්යයකට මාරු විය නොහැකි බැවින් එය අක්‍රියව පැවතිය හැක.

4) User directly interacted with the hardware. / පරිශීලක දෘඩාංග සමඟ සෘජු සම්බන්ධතා පවත්වයි.
In simple batch systems, users do not interact directly with the hardware. Instead, they submit jobs to a batch system that processes them without user intervention during execution.
සරල කණ්ඩායම් පද්ධති වලදී, පරිශීලකයන් දෘඩාංග සමඟ සෘජුව අන්තර්ක්‍රියා නොකරයි. ඒ වෙනුවට, ඔවුන් ක්‍රියාත්මක කිරීමේදී පරිශීලක මැදිහත්වීමකින් තොරව ඒවා සකසන කණ්ඩායම් පද්ධතියකට කාර්යයන් ඉදිරිපත් කරයි.

5. Which of the following is an example of a "One-to-Many" (1:M) relationship?
පහත සඳහන් ඒවායින් ඒක-බහු(1:M) සම්බන්ධතාවයක් සඳහා උදාහරණයක් වන්නේ කුමක්ද?

1) Citizen and NIC number / පුරවැසියා සහ හැඳුනුම්පත් අංකය
2) Student and Course / සිසුවා සහ පාඨමාලාව
3) Customer and Order / පාරිභෝගිකයා සහ ඇණවුම
4) Book and BookId / පොත සහ පොතේ අනන්‍ය අංකය
5) Person and Fingerprint / පුද්ගලයා සහ ඇඟිලි සලකුණ

Answer: 3)

Explanation / පැහැදිලි කිරීම.

Statement 01 / ප්‍රකාශනය 01
One citizen has only one NIC number, and one NIC number belongs to only one citizen. This is a One-to-One (1:1) relationship.
එක් පුරවැසියෙකුට ඇත්තේ එක් හැඳුනුම්පත් අංකයක් පමණි. එමෙන්ම එක් හැඳුනුම්පත් අංකයක් අයිති වන්නේ එක් පුරවැසියෙකුට පමණි. මෙය ඒක-ඒක (1:1) සම්බන්ධතාවයකි.

Statement 02 / ප්‍රකාශනය 02
One student can enroll in many courses, and one course can have many students. This is a Many-to-Many relationship.
එක් ශිෂ්‍යයෙකුට පාඨමාලා කිහිපයක් හැදෑරිය හැකි අතර, එක් පාඨමාලාවක් ශිෂ්‍යයන් කිහිපදෙනෙකුට හැදෑරිය හැක. මෙය බහු-බහු සම්බන්ධතාවයකි.

Statement 03 / ප්‍රකාශනය 03
One customer can place multiple (many) orders over time, but each specific order belongs to only one customer. This is a classic One-to-Many relationship.
එක් පාරිභෝගිකයෙකුට ඇණවුම් රාශියක් සිදු කළ හැකිය. නමුත් ඕනෑම එක් ඇණවුමක් අයිති වන්නේ එක් පාරිභෝගිකයෙකුට පමණි. මෙය ඒක-බහු සම්බන්ධතාවයකි.

Statement 04 / ප්‍රකාශනය 04
Each book has a unique Book Id, and each Book Id has only one book. So it is One-One
එක් පොතකට ඇත්තේ එක් අනන්‍ය අංකයකි. එක් අනන්‍ය අංකයකට ඇත්තේ එක් පොතකි. එම නිසා මෙය ඒක-ඒක සම්බන්ධතාවයකි.

Statement 05 / ප්‍රකාශනය 05
Each person has a unique set of fingerprints, and those fingerprints belong to only that specific person. This is a One-to-One relationship.
සෑම පුද්ගලයෙකුටම අනන්‍ය ඇඟිලි සලකුණු පවතින අතර, එම ඇඟිලි සලකුණු අයිති වන්නේ එම පුද්ගලයාටම පමණි. මෙය ඒක-ඒක සම්බන්ධතාවයකි.

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

```
def count_vowels(string):
    vowels = "aeiou"
    count = 0
    for char in string:
        if char.lower() in vowels:
            count += 1
    return count
result = count_vowels("Hello World")
print(result)
```

1) 2 2) 3 3) 4 4) 5 5) Error: Undefined variable

Answer: 2)

Code Analysis / කේත විශ්ලේෂණය

`vowels = "aeiou"`
This defines the set of vowels (lowercase) that the function will check for in the input string.
මෙය ශ්‍රිතය ආදාන string හි පරීක්ෂා කරන vowels කට්ටලය (කුඩා අකුරු) නිර්වචනය කරයි.

`count = 0`
Initializes a counter to count the number of vowels.
vowels ගණන ගණනය කිරීම සඳහා කවුන්ටරයක් ආරම්භ කරයි.

`for char in string:`
Iterates through each character in the input string.
ආදාන string හි එක් එක් අක්ෂරය හරහා පුනරාවර්තනය වේ.

`if char.lower() in vowels:`
Converts the current character to lowercase and checks if it exists in the string `vowels`. This ensures the function is case-insensitive.
වත්මන් අක්ෂරය lowercase වලට පරිවර්තනය කර එය `vowels` string එකෙහි තිබේදැයි පරීක්ෂා කරයි. මෙය ශ්‍රිතය අක්ෂර-සංවේදී නොවන බව සහතික කරයි.

`count += 1`
Increments the `count` if a vowel is found.
vowels එකක් හමු වුවහොත් `count` වැඩි කරයි.

`return count`
Returns the total count of vowels.
මුළු vowels ගණන ආපසු ලබා දෙයි.

So, when "Hello World" is given as the input, the program counts its vowels. It has 3 vowels: 'e', 'o' and 'o'.
එමනිසා, "Hello World" ආදානය ලෙස ලබා දුන් විට, වැඩසටහන එහි vowels ගණන් කරයි. එයට vowels 3 ක් ඇත: 'e', 'o' සහ 'o'.

Therefore, the program prints 3.
එබැවින්, වැඩසටහන 3 මුද්‍රණය කරයි.

The correct answer is 2).
නිවැරදි පිළිතුර 2) වේ.

7. Which of the following Python expressions correctly checks if x is not equal to y,
පහත දැක්වෙන පයිතන් ප්‍රකාශන වලින් x , y ට සමාන නොවේදැයි නිවැරදිව පරීක්ෂා කරන්නේ,

1) `x <> y` 2) `x !== y` 3) `x /= y` 4) `x != y` 5) `x ==! y`

Answer : 4)

Here's a brief explanation on the options given.
ලබා දී ඇති පිළිතුරු පිළිබඳ කෙටි පැහැදිලි කිරීමක් .

1) **`x <> y`**
This syntax was used in older versions of some programming languages (early versions of Python and SQL), but it is not valid in modern Python.
මෙම වාක්‍ය ඛණ්ඩය සමහර ක්‍රමලේඛන භාෂාවල පරණ අනුවාදයන් (පයිතන් සහ SQL හි මුල් අනුවාදයන්) භාවිතා කරන ලද නමුත් එය නූතන පයිතන් භාෂාවේදී වලංගු නොවේ.

2) **`x !== y`**
This syntax is used in some programming languages like JavaScript, but it is **not valid in Python**. Python does not use the “!==” operator for inequality.
මෙම වාක්‍ය ඛණ්ඩය වැනි සමහර ක්‍රමලේඛන භාෂාවල භාවිතා වේ, නමුත් එය පයිතන් හි වලංගු නොවේ. පයිතන් අසමානතාවය සඳහා “!==” භාවිතා නොකරයි.

3) **`x /= y`**
An assignment operation in Python, divides x by y and assigns the result back to x. Not a comparison operator.
පයිතන් හි පැවරුම් මෙහෙයුමකි (assignment operation), විසින් බෙදී ප්‍රතිඵලය නැවත x වෙත පවරයි. සංසන්දනාත්මක මෙහෙයවනයක් නෙවෙයි.

4) **`x != y`**
This is the correct way to check if x is not equal to y in Python. This comparison operator evaluates to True if the values of x and y are not equal, and False otherwise.
Python හි x, y ට සමාන නොවේදැයි පරීක්ෂා කිරීමට මෙය නිවැරදි ක්‍රමයයි. මෙම සංසන්දන ක්‍රියාකරු x සහ y හි අගයන් සමාන නොවේ නම් True ලෙසත්, එසේ නොමැති නම් False ලෙසත් ඇගයීමට ලක් කරයි.

5) **`x ==! Y`**
Not a valid syntax in Python.
Python හි වලංගු වාක්‍ය ඛණ්ඩයක් නොවේ.

Therefore the correct answer is `x != y`.
එබැවින් නිවැරදි පිළිතුර `x != y` වේ.

8. What is the output of the following Python expression?
පහත සඳහන් පයිතන් ප්‍රකාශනයේ ප්‍රතිදානය කුමක්ද?

```
print(10 >> 8 // 4 << 3)
```

1) `True` 2) `32` 3) Compile error 4) `False` 5) 16

Answer : 5)

Here the operator precedence should be followed.
මෙහිදී මෙහෙයවන ප්‍රමුඛතා අනුපිළිවෙල පිළිපැදිය යුතුය.

```
print(10 >> 8 // 4 << 3)
print(10 >> 2 << 3)
```

$10_{10} = 1010_2$ Right Shift

		1	0	1	0	
→	0	0	1	0	~~1~~	~~0~~

$0010_2 = 2_{10}$

```
print(2 << 3)
```

$2_{10} = 10_2$ Left Shift

			1	0	
1	0	0	0	0	←

$10000_2 = 16_{10}$

```
print(16)
```

Accordingly the output is 16. So the answer is 5.
ඒ අනුව ප්‍රතිදානය වන්නේ 16 වේ. එම නිසා පිළිතුර වන්නේ 5 වේ.

9. What does the <style> tag in HTML signify in terms of CSS?
CSS වලට අනුව HTML හි <style> උසුලනය අදහස් කරන්නේ කුමක්ද?

1) It embeds CSS styles directly within an HTML document.
එය HTML ලේඛනයක් තුළ සෘජුවම CSS විලාසයන් ඇතුළත් කරයි.
2) It links an external CSS file to the HTML document.
එය HTML ලේඛනයට බාහිර CSS ගොනුවක් සම්බන්ධ කරයි.
3) It defines an inline CSS rule for a single HTML element.
එය තනි HTML මුලාංගයක් සඳහා පේළිගත CSS රීතියක් නිර්වචනය කරයි.
4) It enables JavaScript to manipulate CSS properties.
එය JavaScript හට CSS ගුණාංග හැසිරවීමට හැකියාව ලබා දෙයි.
5) It creates a default style sheet for the browser.
එය අතිරික්සුව සඳහා පෙරනිමි විලාස පත්‍රයක් නිර්මාණය කරයි.

Answer: 1)

Explanation / පැහැදිලි කිරීම:

The <style> tag in HTML is used to embed CSS styles directly within an HTML document. This means that the CSS rules written inside the <style> tag apply to the HTML elements within the same document. It is typically placed within the <head> section of the HTML document, but it can also be used in the <body> section if needed.
HTML හි <style> උසුලනය HTML ලේඛනයක් තුළ සෘජුවම CSS විලාස ඇතුළත් කිරීමට භාවිතා කරයි. මෙයින් අදහස් කරන්නේ <style> උසුලනය තුළ ලියා ඇති CSS නීති එකම ලේඛනය තුළ ඇති HTML මුලාංගවලට අදාළ වන බවයි. එය සාමාන්‍යයෙන් HTML ලේඛනයේ <head> කොටස තුළ තබා ඇත, නමුත් අවශ්‍ය නම් එය <body> කොටසේ ද භාවිතා කළ හැකිය.

The correct answer is option 1), It embeds CSS styles directly within an HTML document.
නිවැරදි පිළිතුර විකල්ප 1) වේ, එය HTML ලේඛනයක් තුළ සෘජුවම CSS විලාස ඇතුළත් කරයි.

This is the primary purpose of the <style> tag. It allows you to write CSS rules directly in the HTML file, making it convenient for small-scale styling or quick prototyping.
<style> උසුලනයේ ප්‍රධාන අරමුණ මෙයයි. එය ඔබට HTML ගොනුවේ සෘජුවම CSS නීති ලිවීමට ඉඩ සලසයි, එය කුඩා පරිමාණයේ විලාස හෝ ඉක්මන් මූලාකෘති සඳහා පහසු කරයි.

10. Which HTML attribute is used to define Inline styles?
පේළිගත විලාස නිර්වචනය කිරීමට භාවිතා කරන්නේ කුමන HTML ගුණාංගයද?

1) Class 2) Style 3) Id 4) Font 5) Group

Answer: 2)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

1) Class
Used to specify a class for an element, which can be styled using CSS, but it is not used for inline styles.
මූලාංගයක් සඳහා පන්තියක් නියම කිරීමට භාවිතා කරයි, එය CSS භාවිතයෙන් මෝස්තර කළ හැකි නමුත් එය පේළිගත මෝස්තර සඳහා භාවිතා නොවේ.

2) Style
This attribute is used to apply CSS styles directly to an HTML element. Inline styles are written directly within the style attribute of the HTML tag, allowing for immediate application of CSS properties to that element.
මෙම ගුණාංගය CSS මෝස්තර සෘජුවම HTML මූලාංගයකට යෙදීමට භාවිතා කරයි. පේළිගත විලාස HTML උසුලනයේ විලාස ගුණාංගය තුළ කෙලින්ම ලියා ඇති අතර, එම මූලාංගයට CSS ගුණාංග ක්ෂණිකව යෙදීමට ඉඩ සලසයි.

3) Id
Used to specify a unique identifier for an element, which can be used to apply CSS styles or target the element with JavaScript, but it is not used for inline styles.
මූලාංගයක් සඳහා අනන්‍ය හඳුනාගැනීමක් නියම කිරීමට භාවිතා කරයි, එය CSS මෝස්තර යෙදීමට හෝ JavaScript සමඟ මූලාංගය ඉලක්ක කිරීමට භාවිත කළ හැකි නමුත් එය පේළිගත විලාස සඳහා භාවිත නොවේ.

4) Font
This is not an HTML attribute. Font styles are applied using CSS properties rather than an attribute.
මෙය HTML ගුණාංගයක් නොවේ. Font මෝස්තර යොදනු ලබන්නේ උපලක්ෂණයකට වඩා CSS ගුණ භාවිතා කරමිනි.

5) Group
This is not an HTML attribute related to styling.
මෙය මෝස්තරයට අදාළ HTML උපලක්ෂණයක් නොවේ.

The HTML attribute used to define inline styles is 2) `Style`.
පේළිගත විලාසයන් නිර්වචනය කිරීමට භාවිතා කරන HTML ගුණාංගය වන්නේ 2) `Style` ය.