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 290 MARKING

2025/09/26
ictfromabc
Ravindu Bandaranayake

1. What is the primary role of a device driver in a computer system?
පරිගණක පද්ධතියක උපාංග ධාවකයක මූලික කාර්යභාරය කුමක්ද?

1) To facilitate communication between the operating system and hardware devices.
මෙහෙයුම් පද්ධතිය සහ දෘඪාංග උපාංග අතර සන්නිවේදනය පහසු කිරීම සඳහා.

2) To manage and organize files and directories on a hard drive.
දෘඪ තැටියක ගොනු සහ නාමාවලි කළමනාකරණය කිරීමට සහ සංවිධානය කිරීමට.

3) To control network traffic and manage data packets.
ජාල තදබදය පාලනය කිරීමට සහ දත්ත පැකට් කළමනාකරණය කිරීමට.

4) To provide an interface for user applications to interact with the operating system.
මෙහෙයුම් පද්ධතිය සමඟ අන්තර්ක්‍රියා කිරීමට පරිශීලක යෙදුම් සඳහා අතුරු මුහුණතක් සැපයීම.

5) To optimize the performance of the central processing unit (CPU).
මධ්‍ය සැකසුම් ඒකකයේ (CPU) කාර්ය සාධනය ප්‍රශස්ත කිරීම සඳහා.

Answer: 1)

Explanation / පැහැදිලි කිරීම

The primary role of a device driver in a computer system is to facilitate communication between the operating system and hardware devices. Device drivers act as a bridge, allowing the operating system to interact with and control hardware components such as printers, keyboards, or graphics cards, ensuring that they function properly with the system.
පරිගණක පද්ධතියක උපාංග ධාවකයක මූලික කාර්යභාරය වන්නේ මෙහෙයුම් පද්ධතිය සහ දෘඪාංග උපාංග අතර සන්නිවේදනය පහසු කිරීමයි. උපාංග ධාවක පාලමක් ලෙස ක්‍රියා කරයි, මෙහෙයුම් පද්ධතියට මුද්‍රණ යන්ත්‍ර, යතුරුපුවරු හෝ ග්‍රැෆික් කාඩ්පත් වැනි දෘඪාංග සංරචක සමඟ අන්තර් ක්‍රියා කිරීමට සහ පාලනය කිරීමට ඉඩ සලසයි, ඒවා පද්ධතිය සමඟ නිසි ලෙස ක්‍රියා කරන බව සහතික කරයි.

Therefore, the correct answer is 1) To facilitate communication between the operating system and hardware devices.
එබැවින්, නිවැරදි පිළිතුර වන්නේ 1) මෙහෙයුම් පද්ධතිය සහ දෘඪාංග උපාංග අතර සන්නිවේදනය පහසු කිරීම සඳහාය.

2. Who presented the stored program concept ?
ගබඩා කර ඇති වැඩසටහන් සංකල්පය ඉදිරිපත් කළේ කවුද?

1) John Eckert. / ජෝන් එකර්ට්.
2) Blaise Pascal. / බ්ලේස් පැස්කල්.
3) Ada Lovelace. / ඒඩා ලව්ලේස්.
4) John Von Neuman. / ජෝන් වොන් නියුමාන්.
5) Charles Babbage. / චාල්ස් බැබේජ්.

Answer: 4)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

1) **John Eckert**
Co-inventor of the ENIAC, one of the earliest general-purpose electronic digital computers, but not directly associated with the stored-program concept.
ENIAC හි සම-නිපදවුම්කරු, පැරණිතම පොදු කාර්ය ඉලෙක්ට්‍රොනික ඩිජිටල් පරිගණක වලින් එකක් වන නමුත් ගබඩා කරන ලද වැඩසටහන් සංකල්පය සමඟ සෘජුව සම්බන්ධ නොවේ.

2) **Blaise Pascal**
Invented one of the earliest mechanica l calculators, the Pascaline, but did not contribute to the concept of stored programs in computing.
පැරණිතම යාන්ත්‍රික ගණක යන්ත්‍රයක් වන Pascaline සොයා ගන්නා ලද නමුත් පරිගණකකරණයේ ගබඩා කර ඇති වැඩසටහන් සංකල්පයට දායක නොවීය.

3) **Ada Lovelace**
Recognized as the world's first computer programmer for her work on Charles Babbage's Analytical Engine, she conceptualized algorithms to be executed by a machine.
Charles Babbage ගේ විශ්ලේෂණ එන්ජිම පිළිබඳ ඇයගේ කාර්යය සඳහා ලොව ප්‍රථම පරිගණක ක්‍රමලේඛිකාව ලෙස පිළිගත් ඇය යන්ත්‍රයක් මගින් ක්‍රියාත්මක කළ යුතු ඇල්ගොරිතම සංකල්පගත කළාය.

4) **John von Neumann**
Introduced the stored-program concept, where both data and programs are stored in a computer's memory, forming the basis of most modern computer architectures.
බොහෝ නවීන පරිගණක නිර්මාණ ශිල්පයේ පදනම වන දත්ත සහ වැඩසටහන් දෙකම පරිගණකයක මතකයේ ගබඩා කර ඇති ගබඩා කරන ලද වැඩසටහන් සංකල්පය හඳුන්වා දුන්නේය.

5) **Charles Babbage**
Known as the "father of the computer", he designed the Analytical Engine, a mechanical general-purpose computer, but did not develop the stored-program concept.
"පරිගණකයේ පියා" ලෙස හැඳින්වෙන ඔහු යාන්ත්‍රික පොදු කාර්ය පරිගණකයක් වන විශ්ලේෂණ එන්ජිම නිර්මාණය කළ නමුත් ගබඩා කරන ලද වැඩසටහන් සංකල්පය වර්ධනය කළේ නැත.

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

3. BCD equivalent of 9310 and 3910?
BCD 9310 සහ 3910 ට සමාන වන්නේ?

<u>1) 1001 0011, 0011 1001</u>
2) 1011 0011, 0011 1001
3) 1001 1010, 1001 0001
4) 1110 0001, 0001 1100
5) 1001 0011, 1001 0111

Answer: 1)

Explanation / පැහැදිලි කිරීම

Convert 93 and 39 to BCD (Binary-Coded Decimal).
93 සහ 39 BCD (Binary-Coded Decimal) බවට පරිවර්තනය.

Convert / පරිවර්තනය:

93_{10}	
9	3
1001	0011
10010011_2	

39_{10}	
3	9
0011	1001
00111001_2	

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

4. A digital circuit lights up an LED when a combination of three switches (A, B, C) is OFF. Which form would conveniently express this logic?
ස්විච තුනක (A, B, C) සංයෝජනයක් අක්‍රිය වූ විට ඩිජිටල් පරිපථයක් LED එකක් දැල්වෙයි. මෙම තර්කනය පහසුවෙන් ප්‍රකාශ කළ හැකි ආකාරය කුමක්ද?

   1) SOP using minterms / minterms භාවිතා කරමින් SOP
   2) POS using maxterms / maxterms භාවිතා කරමින් POS
   3) SOP using maxterms / maxterms භාවිතා කරමින් SOP
   4) XOR form / XOR ලෙසින්
   5) Karnaugh map only / Karnaugh සිතියම පමණි

   Answer: 2)

   The LED lights up when all switches A, B, and C are OFF. In Boolean logic, “OFF” is 0, so the output is 1 only for the combination A=0, B=0, C=0.
   A, B සහ C ස්විච සියල්ල අක්‍රිය වූ විට LED එක දැල්වෙයි. බූලියන් තර්කනයේ, “OFF” 0 වන බැවින් ප්‍රතිදානය 1 වන්නේ A=0, B=0, C=0 සංයෝජනය සඳහා පමණි.

   POS (Product of Sums) using maxterms is ideal for expressing logic that is true when inputs are mostly 0, because each maxterm corresponds to an input combination that makes the output 0.
   Maxterms භාවිතා කරන POS (එකතුන්හි ගුණිතය) ආදාන බොහෝ දුරට 0 වන විට සත්‍ය වන තර්කනය ප්‍රකාශ කිරීම සඳහා වඩාත් සුදුසු වේ, මන්ද, එක් එක් maxterm ප්‍රතිදානය 0 බවට පත් කරන ආදාන සංයෝජනයකට අනුරූප වේ.

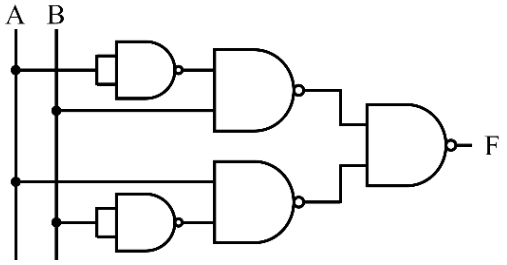
   SOP (Sum of Products) is convenient when the output is 1 for many combinations, which is not the case here.
   බොහෝ සංයෝජන සඳහා ප්‍රතිදානය 1 වන විට SOP (ගුණිතවල එකතුව) පහසු වේ, එය මෙහි එසේ නොවේ.

   XOR form is for exclusive OR logic, not suitable for this condition.
   XOR ආකෘතිය සුවිශේෂී OR තර්කනය සඳහා වන අතර, මෙම තත්වයට සුදුසු නොවේ.

   Karnaugh maps are a tool, not a direct expression form.
   Karnaugh සිතියම් යනු මෙවලමක් වන අතර සෘජු ප්‍රකාශන ආකාරයක් නොවේ.

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

5. Which logic gate is equivalent to the above circuit?
ඉහත පරිපථයට සමාන වන්නේ කුමන තාර්කික ද්වාරයද?



   1) AND   2) OR   3) XOR   4) XNOR   5) NOT

   Answer: 3)

Explanation / පැහැදිලි කිරීම:

Let’s build up a Boolean expression for the given circuit.
දී ඇති පරිපථය සඳහා බූලියන් ප්‍රකාශනයක් ගොඩනඟමු.

$F = \overline{(\overline{\overline{A}B})(\overline{A\overline{B}})}$

Now, let’s simplify it.
දැන්, එය සරල කරමු.

$= \overline{(\overline{\overline{A}B})(\overline{A\overline{B}})}$

$= \overline{(\overline{A\overline{B}})} + \overline{(\overline{\overline{A}B})}$ De Morgan’s Law / ද මුවාර් ප්‍රමේයය

$= A\overline{B} + \overline{A}B$ Double complement Law / ද්විත්ව ප්‍රතිලෝම න්‍යාය

$= A \oplus B$ $A\overline{B} + \overline{A}B = A \oplus B$

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

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

```
Salary = [10000,12500,9000,15000,20000]
Salary[1:3] = [18000, 19000]
Salary[5]= 25000
print(Salary)
```

1) `[18000, 19000, 15000, 20000]`
2) `[10000, 18000, 19000, 9000, 15000, 20000]`
3) `[10000, [18000, 19000], 9000, 15000, 20000]`
4) `[10000, 18000, 19000, 15000, 20000,25000]`
<u>5) Error</u>

Answer: 5)

Let’s analyze each code lines one by one.
අපි එක් එක් කේත රේඛා එකින් එක විශ්ලේෂණය කරමු“

A. `Salary = [10000, 12500, 9000, 15000, 20000]`
The list `Salary` initially contains the values `[10000, 12500, 9000, 15000, 20000]`.
`Salary` ලැයිස්තුවේ මුලින් `[10000, 12500, 9000, 15000, 20000]` අගයන් අඩංගු වේ.

B. `Salary[1:3] = [18000, 19000]`
This line replaces the elements in positions 1 and 2 (which are 12500 and 9000) with the values `[18000, 19000]`. The list now becomes `[10000, 18000, 19000, 15000, 20000]`.
මෙම රේඛාව 1 සහ 2 පිහිටුම්වල ඇති මූලාංග (ඒවා 12500 සහ 9000) `[18000, 19000]` අගයන් සමග ප්‍රතිස්ථාපනය කරයි. ලැයිස්තුව දැන් `[10000, 18000, 19000, 15000, 20000]` බවට පත්වේ.

C. `Salary[5] = 25000`
This line attempts to assign the value 25000 to the element at index 5. However, the list currently only has 5 elements, so index 5 does not exist. This will cause an IndexError.
මෙම පේළිය 5 වන දර්ශකයේ ඇති මූලාංගයට 25000 අගය පැවරීමට උත්සාහ කරයි. කෙසේ වෙතත්, ලැයිස්තුවේ දැනට ඇත්තේ මූලාංග 5ක් පමණි, එබැවින් 5 දර්ශකය නොපවතී. මෙය IndexError එකක් ඇති කරයි.

Since an error occurs at this step, the correct answer is 5).
මෙම පියවරේදී දෝෂයක් සිදු වන බැවින්, නිවැරදි පිළිතුර වන්නේ 5) ය.

7. Consider the following python data.
පහත පයිතන් දත්ත සලකා බලන්න.

A. `IT = "2000"`
B. `AL = 'ictfromabc'`
C. `ict,from,abc = 120`
D. `ict = from = abc = 120,000`

Which of the above Python statement(s) is/are correct according to the syntax rules ?
ඉහත දී ඇති ඒවා අතුරින් කාරක රීති වලට අනුකූල වන නිවැරදි පයිතන් ප්‍රකාශ(ය) මොනවාද?

1) A only. / A පමණි.
2) B only. / B පමණි.
3) D only. / D පමණි.
4) D and C only. / D හා C පමණි.
5) A and B only. / A හා B පමණි.

Answer: 5)

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

A. **`IT = "2000"`**
This statement is correct because Python allows variable names with letters, and the value "2000" is a valid string. No syntax errors here.
මෙම ප්‍රකාශය නිවැරදි වන්නේ Python අකුරු සහිත විචල්‍ය නාම වලට ඉඩ දෙන නිසාත්, "2000" අගය වලංගු string එකක් වන නිසාත් ය. මෙහි සින්ටැක්ස් දෝෂ නොමැත.

B. **`AL = 'ictfromabc'`**
This statement is also correct. It assigns the string `ictfromabc` to the variable `AL`, following the correct syntax for variable assignment in Python.
මේ ප්‍රකාශයත් නිවැරදි වේ. එය පයිතන් හි විචල්‍ය පැවරුම් සඳහා නිවැරදි වාක්‍ය ඛණ්ඩය අනුගමනය කරමින් `AL` විචල්‍යයට `ictfromabc` string එක පවරයි.

C. **`ict,from,abc = 120`**
This statement is incorrect because while tuple unpacking can be used to assign values to multiple variables, there should be a corresponding number of values on the right side. Here, there's only one value (`120`) assigned to three variables, which is not allowed without additional formatting.
මෙම ප්‍රකාශය නිවැරදි නොවේ මන්ද යත් බහු විචල්‍යයන් සඳහා අගයන් පැවරීමට tuple unpack කිරීම භාවිතා කළ හැකි අතර, දකුණු පැත්තේ ඊට අනුරූප අගයන් සංඛ්‍යාවක් තිබිය යුතුය. මෙහිදී, විචල්‍ය තුනකට පවරා ඇත්තේ එක් අගයක් (`120`) පමණි, අමතර හැඩතල ගැන්වීමකින් තොරව ඉඩ නොදේ.

D. **`ict = from = abc = 120,000`**
This statement is incorrect because `120,000` is not a valid numeric format in Python due to the comma. It should be written as `120,000` or use a decimal point if needed.
කොමාව හේතුවෙන් පයිතන් හි `120,000` වලංගු සංඛ්‍යාත්මක ආකෘතියක් නොවන නිසා මෙම ප්‍රකාශය වැරදිය. එය `120,000` ලෙස ලිවිය යුතුය හෝ අවශ්‍ය නම් දශම ලක්ෂයක් භාවිතා කරන්න.

Therefore, the correct answer is, 5).
එබැවින්, නිවැරදි පිළිතුර නම්, 5).

8. Is Python case sensitive when dealing with identifiers?
හඳුන්වන සමඟ ගනුදෙනු කිරීමේදී පයිතන් කැපිටල් සිම්පල් සංවේදීද?

1) No. / නැත.
2) Yes. / ඔව්.
3) Machine dependent / ක්‍රියාකරන යන්ත්‍රය මත රඳා පවතී.
4) Both 2 and 3. / 2 සහ 3 යන දෙකම.
5) None of the mentioned. / ඉහත කිසිවක් නොවේ.

Answer: 2)

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

Option / විකල්පය 1)
This option is incorrect because Python is case sensitive. Identifiers with different cases are treated as distinct. Python සිද්ධි සංවේදී බැවින් මෙම විකල්පය වැරදියි. විවිධ අවස්ථා සහිත හඳුනාගැනීම් වෙනස් ලෙස සලකනු ලැබේ.

Option / විකල්පය 2)
This is the correct answer. Python is case sensitive, so Variable, variable, and VARIABLE are considered different identifiers. මෙය නිවැරදි පිළිතුරයි. පයිතන් සිද්ධි සංවේදී වේ, එබැවින් විචල්‍ය, විචල්‍ය සහ විචල්‍ය විවිධ හඳුනාගැනීම් ලෙස සැලකේ.

Option / විකල්පය 3)
This is incorrect. Python's case sensitivity is not machine-dependent; it is a part of the language's design and applies universally regardless of the machine or operating system being used.
මෙය වැරදියි. Python case sensitivity යන්ත්‍ර මත රඳා නොපවතී එය භාෂාවේ සැලසුමේ කොටසක් වන අතර භාවිතා කරන යන්ත්‍රය හෝ මෙහෙයුම් පද්ධතිය කුමක් වුවත් විශ්වීය වශයෙන් අදාළ වේ.

Option / විකල්පය 4)
This option is incorrect because option 3 is not accurate. Python's case sensitivity is not machine dependent, so combining the correct answer 2) with the incorrect one 3) is not valid.
විකල්පය 3) නිවැරදි නොවන නිසා මෙම විකල්පය වැරදියි. Python සිද්ධි සංවේදීතාව යන්ත්‍ර මත රඳා නොපවතී, එබැවින් නිවැරදි පිළිතුර 2) වේ. 3) සමඟ ඒකාබද්ධ කිරීම වලංගු නොවේ.

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

9. What is the output of the following Python program when executed?
පහත දක්වා ඇති පයිතන් ක්‍රමලේඛය ක්‍රියාත්මක කළ විට ප්‍රතිදානය කුමක්ද?

```
mylist=("ictfromabc,",["a1","a2"],"only",(2024,2025,2026))
print(mylist[3][1:])
```

1) `(2024, 2025, 2026)`
2) `only`
3) <u>`(2025, 2026)`</u>
4) `2025`
5) Syntax error

Answer: 3)

Explanation / පැහැදිලි කිරීම:

`mylist` is a tuple that contains four elements,
`mylist` යනු මූලාංග හතරක් අඩංගු ටපල් එකකි,

- `"ictfromabc,"` - a string
- `["a1", "a2"]` - a list
- `"only"` - a string
- `(2024, 2025, 2026)` - a tuple

`mylist[3]` accesses the fourth element of the tuple, which is `(2024, 2025, 2026)`.
`mylist[3]` tuple හි සිව්වන මූලාංග වෙත ප්‍රවේශ වේ, එනම් `(2024, 2025, 2026)`.

`mylist[3][1:]` slices the tuple `(2024, 2025, 2026)` starting from the second element (index 1) to the end. This gives the tuple `(2025, 2026)`.
`mylist[3][1:]` tuple `(2024, 2025, 2026)` දෙවන මූලාංගයේ (index 1) සිට අවසානය දක්වා කොටස් කරයි. මෙය tuple `(2025, 2026)` ලබා දෙයි.

So, the output of the program is `(2025, 2026)`.
එබැවින්, වැඩසටහනේ ප්‍රතිදානය `(2025, 2026)` වේ.

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

```
def outer():
    def inner():
        return 10
    return inner

func = outer()
print(func())
```

1) 10 2) 20 3) 100 4) syntax error 5) no output

Answer: 1)

Explanation / පැහැදිලි කිරීම

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

`def outer():`
Here, we define a function called `outer`. This function contains another function inside it, which is called `inner`.
මෙහිදී, අපි `outer` නම් ශ්‍රිතයක් නිර්වචනය කරමු. මෙම ශ්‍රිතය තුළ තවත් ශ්‍රිතයක් අඩංගු වන අතර එය `inner` ලෙස හැඳින්වේ.

```
def inner():
      return 10
```

Inside `outer`, there's another function defined called `inner`. The `inner` function simply returns the value 10 when called.
`outer` තුළ, `inner` ලෙස අර්ථ දක්වා ඇති තවත් ශ්‍රිතයක් ඇත. `inner` ශ්‍රිතය ඇමතූ විට 10 අගය ලබා දෙයි.

`return inner`
The `outer` function returns the `inner` function itself, not its result. So when `outer()` is called, it returns the `inner` function.
`outer` ශ්‍රිතය `inner` ශ්‍රිතයම ලබා දෙයි, එහි ප්‍රතිඵලය නොවේ. එබැවින් `outer()` ලෙස හැඳින්වූ විට, එය `inner` ශ්‍රිතය ආපසු ලබා දෙයි.

`func = outer()`
When `outer()` is called, it returns the `inner` function, and this function is assigned to the variable `func`. This means `func` now holds a reference to the `inner` function.
`outer()` කැඳවූ විට, එය `inner` ශ්‍රිතය ආපසු ලබා දෙන අතර, මෙම ශ්‍රිතය `func` විචල්‍ය වෙත පවරා ඇත. මෙයින් අදහස් කරන්නේ `func` දැන් `inner` ශ්‍රිතයට සඳහනක් දරයි.

`print(func())`
When `func()` is called, it's equivalent to calling `inner()`. Since `inner` returns 10, `func()` will return 10, which is then printed.
`func()` කැඳවූ විට, එය `inner()` ඇමතීමට සමාන වේ. `inner` මගින් 10 ලබාදෙන බැවින්, `func()` විසින් 10 ලබා දෙනු ඇත, එය ප්‍රතිදානය කෙරේ.

The output will be `10`. So, the correct answer number is 1.
ප්‍රතිදානය `10` වනු ඇත. එමනිසා, නිවැරදි පිළිතුරු අංකය 1 වේ.

❋ ❋ ❋