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 360 MARKING

2025/12/05
ictfromabc
Ravindu Bandaranayake

1. In which generation were Expert systems introduced?
විශේෂ පද්ධති හඳුන්වා දෙන ලද්දේ කුමන පරම්පරාවේදීද?

1) First Generation පළමු පරම්පරාව
2) Second Generation දෙවන පරම්පරාව
3) 13th Generation දහතුන්වන පරම්පරාව
4) Third Generation තුන්වන පරම්පරාව
5) <u>Fourth Generation</u> හතරවන පරම්පරාව

Answer : 5)

Special systems are introduced in the 4th generation. Also, parallel processing, robotics and artificial intelligence have been introduced in the 4th generation. And in the 4th generation, microprocessors were used. In this generation, everyone is used to using personal computers. This is due to reduced processing time and made smaller computers. During this time, palm computers and smart phones were produced. We are in the period between the 4th generation and the 5th generation.
විශේෂ පද්ධති 4 වන පරම්පරාවේදී හඳුන්වා දී ඇත. එමෙන්ම සමාන්තර සැකසුම් , රොබෝ යන්ත්‍ර සහ කෘතිම බුද්ධිය 4 වන පරම්පරාවේදී හඳුන්වාදී ඇත. එමෙන්ම 4 වන පරම්පරාවේදී ක්ෂුද්‍ර සකසන භාවිත විය. මෙම පරම්පරාවේදී සියලුම දෙනා පුද්ගලික පරිගණක භාවිතයට හුරුවිය. එයට හේතුව වන්නේ සැකසීම් කාලය අඩු වීම හා පරිගණක ප්‍රමාණයෙන් කුඩා වීමයි. මේ කාලයේදී අත්ල පරිගණක , ස්මාර්ට් දුරකථන නිපදවුණි. අප මේ සිටින්නේද 4 වන පරම්පරාව හා 5 වන පරම්පරාව අතර කාල පරාසයේය.

The first generation computers emerged in the 1940s and lasted until the mid-1950s. These computers were characterized by their use of vacuum tubes for circuitry and relied heavily on punched cards and paper tape for input and output.
පළමු පරම්පරාවේ පරිගණක 1940 දී පමණ මතු වූ අතර 1950 මැද භාගය දක්වා පැවතුණි. මෙම පරිගණක පරිපථ සඳහා රික්තක නළ භාවිතා කිරීම මගින් සංලක්ෂිත වූ අතර ආදානය සහ ප්‍රතිදානය සඳහා සිදුරු කරන ලද කාඩ්පත් සහ කඩදාසි පටි මත දැඩි ලෙස රඳා පැවතුනි.

The second generation of computers emerged in the late 1950s and lasted until the mid-1960s. These computers were characterized by the use of transistors instead of vacuum tubes, which made them smaller, faster, cheaper, and more reliable than their predecessors. Additionally, second-generation computers used magnetic core memory for storage, which was more efficient and reliable than the earlier drum memory.
දෙවන පරම්පරාවේ පරිගණක 1950 අග භාගයේදී මතු වූ අතර 1960 මැද භාගය දක්වා පැවතුනි. මෙම පරිගණක රික්තක නළ වෙනුවට ට්‍රාන්සිස්ටර භාවිතා කිරීම මගින් සංලක්ෂිත වූ අතර එමගින් ඒවා ඒවායේ පූර්වගාමීන්ට වඩා කුඩා, වේගවත්, ලාභදායී සහ විශ්වාසදායක විය. මීට අමතරව, දෙවන පරම්පරාවේ පරිගණක ගබඩා කිරීම සඳහා චුම්බක මූලික මතකය භාවිතා කරන ලද අතර එය පෙර පැවති බෙර මතකයට වඩා කාර්යක්ෂම සහ විශ්වාසදායක විය.

The third generation of computers, which emerged in the 1960s, introduced significant advancements over their predecessors. These computers were characterized by the use of integrated circuits (ICs), which combined multiple transistors and other components onto a single semiconductor chip. A distinctive feature of third-generation computers is the use of high-level programming languages and software.
1960 ගණන්වල බිහි වූ තුන්වන පරම්පරාවේ පරිගණක, ඒවායේ පූර්වගාමීන්ට වඩා සැලකිය යුතු දියුණුවක් හඳුන්වා දුන්නේය. බහු ට්‍රාන්සිස්ටර සහ අනෙකුත් සංරචක එක් අර්ධ සන්නායක චිපයක් මත ඒකාබද්ධ කරන ලද අනුකලිත පරිපථ (IC) භාවිතයෙන් මෙම පරිගණක සංලක්ෂිත විය. තුන්වන පරම්පරාවේ පරිගණක ඉහළ මට්ටමේ ක්‍රමලේඛන භාෂා සහ මෘදුකාංග භාවිතා කිරීම කැපීපෙනෙන ලක්ෂණයකි.

2. Both .... and .... can also be referred to as server computers.
.... සහ .... යන දෙකම සේවාදායක පරිගණක ලෙසද හැඳින්විය හැක.

1) Mini computers, Mainframe computers.
කුඩා පරිගණක, ප්‍රධාන රාමුවේ පරිගණක
2) Super computers, Mainframe computers.
සුපිරි පරිගණක, ප්‍රධාන රාමුවේ පරිගණක
3) Mini computers, Micro computers.
කුඩා පරිගණක, ක්ෂුද්‍ර පරිගණක
4) Super computers, Mini computers.
සුපිරි පරිගණක, කුඩා පරිගණක
5) Mainframe computers, Micro computers.
ප්‍රධාන රාමුවේ පරිගණක, ක්ෂුද්‍ර පරිගණක

Answer : 1)

Both mini computers and mainframe computers can also be referred to as server computers. Mini computers, also known as mid-range computers, are designed to serve multiple users simultaneously, making them suitable for tasks such as managing databases and running enterprise applications. These computers were commonly used in the past for scientific and engineering computations, and they have evolved to serve as servers in various business environments.
කුඩා පරිගණක සහ ප්‍රධාන රාමුවේ පරිගණක යන දෙකම සේවාදායක පරිගණක ලෙසද හැඳින්විය හැක. කුඩා පරිගණක, මධ්‍යම පරාසයේ පරිගණක ලෙසද හඳුන්වනු ලබන අතර, ඒවා දත්ත සමුදායන් කළමනාකරණය කිරීම සහ ව්‍යවසාය යෙදුම් ක්‍රියාත්මක කිරීම වැනි කාර්යයන් සඳහා යෝග්‍ය කරමින් බහු පරිශීලකයින්ට එකවර සේවය කිරීමට සැලසුම් කර ඇත. මෙම පරිගණක අතීතයේ විද්‍යාත්මක හා ඉංජිනේරු ගණනය කිරීම් සඳහා බහුලව භාවිතා වූ අතර ඒවා විවිධ ව්‍යාපාරික පරිසරයන්හි සේවාදායකයන් ලෙස පරිණාමය වී ඇත.

Mainframe computers / ප්‍රධාන රාමුවේ පරිගණක
Mainframe computers, on the other hand, are powerful machines capable of handling and processing very large amounts of data quickly. They are often employed by large organizations for critical applications, such as bulk data processing, transaction processing, and resource planning. Mainframe computers are designed to support thousands of users simultaneously and are known for their reliability and scalability, making them ideal candidates for server roles.
අනෙක් අතට, ප්‍රධාන රාමුවේ පරිගණක ඉතා විශාල දත්ත ප්‍රමාණයක් ඉක්මනින් හැසිරවීමට සහ සැකසීමට හැකියාව ඇති බලවත් යන්ත්‍ර වේ. තොග දත්ත සැකසීම, ගනුදෙනු සැකසීම සහ සම්පත් සැලසුම් කිරීම වැනි තීරණාත්මක යෙදුම් සඳහා ඒවා බොහෝ විට විශාල සංවිධාන විසින් යොදවනු ලැබේ. ප්‍රධාන රාමුවේ පරිගණක නිර්මාණය කර ඇත්තේ දහස් ගණන් පරිශීලකයින්ට එකවර සහාය දැක්වීම සඳහා වන අතර ඒවායේ විශ්වසනීයත්වය සහ පරිමාණය සඳහා ප්‍රසිද්ධ වී ඇති අතර, ඒවා සේවාදායක භූමිකාවන් සඳහා සුදුසු අපේක්ෂකයින් බවට පත් කරයි.

Super computers / සුපිරි පරිගණක
Super computers, while extremely powerful and used for complex computations and simulations, are typically not referred to as server computers. These machines are primarily used for scientific research, weather forecasting, and other tasks that require immense processing power. Therefore, option 2) Super computers, Mainframe computers is incorrect.
සුපිරි පරිගණක, අතිශයින් බලවත් සහ සංකීර්ණ ගණනය කිරීම් සහ සමාකරණ සඳහා භාවිතා කරන අතර, සාමාන්‍යයෙන් සේවාදායක පරිගණක ලෙස හඳුන්වනු නොලැබේ. මෙම යන්ත්‍ර මූලික වශයෙන් විද්‍යාත්මක පර්යේෂණ, කාලගුණ අනාවැකි සහ අතිවිශාල සැකසුම් බලයක් අවශ්‍ය කරන වෙනත් කාර්යයන් සඳහා යොදා ගැනේ. එබැවින්, විකල්ප 2) සුපිරි පරිගණක, ප්‍රධාන රාමුවේ පරිගණක වැරදියි.

Micro computers / ක්ෂුද්‍ර පරිගණක
Micro computers, also known as personal computers or PCs, are designed for individual use. While they can function as servers in small-scale applications or home networks, they are not typically referred to as server computers in the context of enterprise-level tasks. Thus, options 3) Mini computers, Micro computers and 5) Mainframe computers, Micro computers do not fit the description.
ක්ෂුද්‍ර පරිගණක, පුද්ගලික පරිගණක හෝ PC ලෙසද හැඳින්වේ, තනි භාවිතය සඳහා නිර්මාණය කර ඇත. කුඩා පරිමාණ යෙදුම්වල හෝ ගෘහ ජාලවල සේවාදායකයන් ලෙස ක්‍රියා කළ හැකි නමුත්, ව්‍යවසාය මට්ටමේ කාර්යයන්හි සන්දර්භය තුළ ඒවා සාමාන්‍යයෙන් සේවාදායක පරිගණක ලෙස හඳුන්වනු නොලැබේ.මේ අනුව, විකල්ප 3) කුඩා පරිගණක, ක්ෂුද්‍ර පරිගණක සහ 5) ප්‍රධාන රාමුවේ පරිගණක, ක්ෂුද්‍ර පරිගණක විස්තරයට නොගැළපේ.

Super computers and mini computers serve different purposes and are not typically grouped together as server computers. Super computers are specialized for high-performance tasks, whereas mini computers are more versatile and commonly used as servers. Therefore, option 4) Super computers, Mini computers is also incorrect.
සුපිරි පරිගණක සහ කුඩා පරිගණක විවිධ අරමුණු සඳහා සේවය කරන අතර සාමාන්‍යයෙන් සේවාදායක පරිගණක ලෙස එකට එකතු වී නොමැත. සුපිරි පරිගණක ඉහළ කාර්ය සාධන කාර්යයන් සඳහා විශේෂිත වන අතර කුඩා පරිගණක වඩාත් බහුකාර්ය වන අතර සේවාදායක ලෙස බහුලව භාවිතා වේ. එබැවින්, විකල්ප 4) සුපිරි පරිගණක, කුඩා පරිගණක ද වැරදියි.

So, both mini computers and mainframe computers are appropriately referred to as server computers due to their ability to manage multiple users and handle extensive data processing tasks. Therefore, the correct answer is **1) Mini computers, Mainframe computers.**
එබැවින්, බහු පරිශීලකයින් කළමනාකරණය කිරීමට සහ පුළුල් දත්ත සැකසුම් කාර්යයන් හැසිරවීමට ඇති හැකියාව හේතුවෙන් කුඩා පරිගණක සහ ප්‍රධාන රාමුවේ පරිගණක යන දෙකම සුදුසු ලෙස සේවාදායක පරිගණක ලෙස හැඳින්වේ. එබැවින්, නිවැරදි පිළිතුර වන්නේ **1) කුඩා පරිගණක, ප්‍රධාන රාමුවේ පරිගණක.**

3. $10110011_2 + 00111100_2 =$ ….A….

What is the answer when the value obtained for A is subjected to bitwise AND operation with 11100011_2?
මෙහි A සඳහා ලැබෙන අගය 11100011_2 සමග බිටු අනුසාරිත AND කර්මයට ලක් කළ විට ලැබෙන පිළිතුර වන්නේ කුමක් ද?

1) 11100010_2 2) $\underline{227}_{10}$ 3) 77_8 4) 256_{10} 5) 11100110_2

Answer : 2)

First Step :
පළමු පියවර

$$
\begin{array}{r}
10110011_2 \\
+00111100_2 \\
\hline
11101111_2 \\
\hline
\end{array}
$$

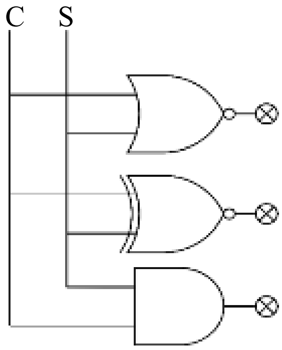
Second Step:
දෙවන පියවර

11101111_2 AND $11100011_2 = 11100011_2$

Third Step :
තෙවන පියවර

<table>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1 x 128</td><td>1 x 64</td><td>1 x 32</td><td>0 x 16</td><td>0 x 8</td><td>0 x 4</td><td>1 x 2</td><td>1 x 1</td></tr>
<tr><td colspan="8">128 + 64 + 32 + 2 + 1 = 227_{10}</td></tr>
<tr><td colspan="2">011</td><td colspan="3">100</td><td colspan="3">011</td></tr>
<tr><td>1 x 2</td><td>1 x 1</td><td>1 x 4</td><td>0 x 2</td><td>0 x 1</td><td>0 x 4</td><td>1 x 2</td><td>1 x 1</td></tr>
<tr><td colspan="2">2 + 1</td><td colspan="3">4</td><td colspan="3">2 + 1</td></tr>
<tr><td colspan="2">3</td><td colspan="3">4</td><td colspan="3">3</td></tr>
<tr><td colspan="8">343_8</td></tr>
<tr><td colspan="4">1110</td><td colspan="4">0011</td></tr>
<tr><td>1 x 8</td><td>1 x 4</td><td>1 x 2</td><td>0 x1</td><td>0 x 8</td><td>0 x 4</td><td>1 x 2</td><td>1 x 1</td></tr>
<tr><td colspan="4">8 + 4 + 2</td><td colspan="4">2 + 1</td></tr>
<tr><td colspan="4">14 = E</td><td colspan="4">3</td></tr>
<tr><td colspan="8">$E3_{16}$</td></tr>
</table>

4. In a rain prediction system there are 3 LEDs to say the predictions, it checks for clouds (C=1, when clouds appear) and sunlight (S=0, when sunlight is available)
වැසි පුරෝකථන පද්ධතියක අනාවැකි කීමට LED 3ක් ඇත, එය වලාකුළු (C-1, වලාකුළු දිස්වන විට) සහ හිරු එළිය (S=0, හිරු එළිය ඇති විට) පරීක්ෂා කරයි.



Mostly it won't rain
බොහෝ විට වැසි නොලැබේ

50%-50% or Night
50%-50% හෝ රාත්‍රි

Mostly it will rain
බොහෝ විට වැසි ඇතිවේ

1) AND gate
2) OR gate
3) XOR gate
4) XNOR gate
5) NAND gate

What should be the logic gate in X to get the above results
ඉහත ප්‍රතිඵල ලබා ගැනීමට X හි ඇති තාර්කික ද්වාරය කුමක් විය යුතුද?

Answer : 3)

Answer 3, 50-50 is when there's clouds and sunlight both together (C=1, S=0) and night occurs when no clouds and no sunlight (C=0, S=1) so to get a true output only in these two instances (10,01) we should use an XOR gate.
පිළිතුර 3, 50-50 යනු වලාකුළු සහ හිරු එළිය යන දෙකම එකට ඇති විට (C=1, S=0) සහ වලාකුළු නොමැති විට සහ සූර්යාලෝකය නොමැති විට රාත්‍රිය(C=0, S=1) වේ. මෙම අවස්ථා 2න්(10,01) පමණක් සත්‍ය ප්‍රතිදානයක් ලබා ගැනීම සඳහා XOR ද්වාරයක් භාවිතා කළ යුතුය.

5. What would be the result if the following Boolean expression is simplified?
පහත දැක්වෙන බූලියානු ප්‍රකාශනය සරල කළහොත් එහි ප්‍රතිඵලය කුමක් වනු ඇත්ද?

$F = \overline{PQ}(P + \overline{Q})(\overline{P} + Q)$

1) $\overline{P}$ 2) $\overline{Q}$ 3) $\overline{PQ}$ 4) $\overline{P}\,\overline{Q}$ 5) $(\overline{P} + \overline{Q})$

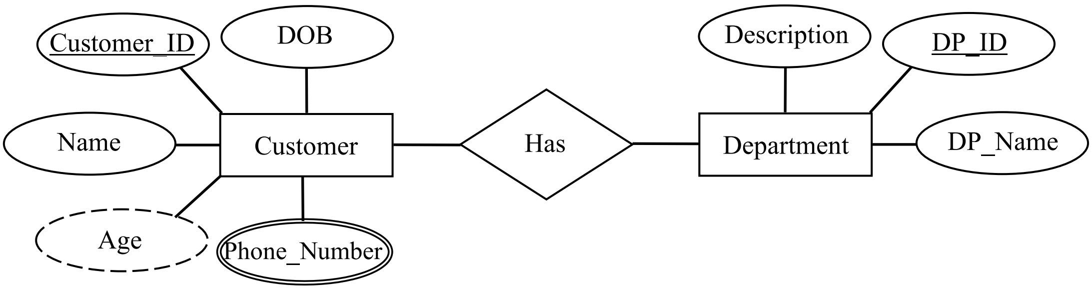
Answer : 4)

Let's simplify the following expression.
පහත ප්‍රකාශය සරල කරමු.

$F = \overline{PQ}(P + \overline{Q})(\overline{P} + Q)$	
$F = (\overline{P} + \overline{Q})(P + \overline{Q})(\overline{P} + Q)$	De Morgan's law ද මුවාර් ප්‍රමේයය
$F = (\overline{P} + \overline{Q})(P\overline{P} + PQ + \overline{P}\,\overline{Q} + Q\overline{Q})$	Distributive law විඝටන න්‍යාය
$F = (\overline{P} + \overline{Q})(0 + PQ + \overline{P}\,\overline{Q} + Q\overline{Q})$	Inverse law ප්‍රතිලෝම න්‍යාය
$F = (\overline{P} + \overline{Q})(PQ + \overline{P}\,\overline{Q} + Q\overline{Q})$	Identity law සර්වසාම්‍ය න්‍යාය
$F = (\overline{P} + \overline{Q})(PQ + \overline{P}\,\overline{Q} + 0)$	Inverse law ප්‍රතිලෝම න්‍යාය
$F = (\overline{P} + \overline{Q})(PQ + \overline{P}\,\overline{Q})$	Identity law සර්වසාම්‍ය න්‍යාය
$F = (\overline{P}PQ + \overline{PP}\,\overline{Q} + PQ\overline{Q} + \overline{P}\,\overline{Q}\,\overline{Q})$	Distributive law විඝටන න්‍යාය
$F = (0 + \overline{PP}\,\overline{Q} + 0 + \overline{P}\,\overline{Q}\,\overline{Q})$	Inverse law ප්‍රතිලෝම න්‍යාය
$F = \overline{PP}\,\overline{Q} + \overline{P}\,\overline{Q}\,\overline{Q}$	Identity law සර්වසාම්‍ය න්‍යාය
$F = \overline{P}\,\overline{Q} + \overline{P}\,\overline{Q}$	Idempotent law තදේවභාවී න්‍යාය
$F = \overline{P}\,\overline{Q}$	Idempotent law තදේවභාවී න්‍යාය

So, the correct answer is 4.
එබැවින්, නිවැරදි පිළිතුර 4 වේ.

6. Consider the below ER diagram.
පහත ER රූප සටහන සලකා බලන්න.



What is the type of attribute used for the age of the customer?
පාරිභෝගිකයාගේ වයස (age) සඳහා භාවිතා කරන උපලක්ෂණ වර්ගය කුමක්ද?

1) Simple attribute / සරල උපලක්ෂණය
2) Composite attribute / සංයුක්ත උපලක්ෂණය
3) Multivalued attribute / බහු අගයන් සහිත උපලක්ෂණය
4) Key attributes / යතුරු උපලක්ෂණය
5) Derived attribute / ව්‍යුත්පන්න උපලක්ෂණය

Answer: 5)

In an ER diagram / ER රූප සටහනක:

1. **Simple attribute / සරල උපලක්ෂණය**
   An attribute that cannot be divided further. For example, **"Name"** is a simple attribute because it stands alone.
   තවදුරටත් බෙදිය නොහැකි උපලක්ෂණයකි. උදාහරණයක් ලෙස,**"Name"** යනු එය තනිව පවතින නිසා සරල උපලක්ෂණයකි.

2. **Composite attribute / සංයුක්ත උපලක්ෂණය**
   An attribute that can be divided into smaller parts. For example, if we have an address, it could be divided into street, city, and postal code.
   කුඩා කොටස් වලට බෙදිය හැකි උපලක්ෂණයකි. උදාහරණයක් ලෙස, අපට ලිපිනයක් තිබේ නම්, එය වීදිය, නගරය සහ තැපැල් කේතය ලෙස බෙදිය හැකිය.

3. **Multivalued attribute / බහු අගයන් සහිත උපලක්ෂණය**
   An attribute that can have multiple values. For example, **"Phone_Number"** could have more than one value if a customer has multiple phone numbers.
   බහු අගයන් තිබිය හැකි උපලක්ෂණයකි. උදාහරණයක් ලෙස, පාරිභෝගිකයෙකුට දුරකථන අංක කිහිපයක් තිබේ නම්, **"Phone_Number"** හි අගය එකකට වඩා තිබිය හැක.

4. **Key attribute / යතුරු උපලක්ෂණය**
   An attribute that uniquely identifies an entity. In this case, **"Customer_ID"** is a key attribute for the customer.
   වස්තුවක් අනන්‍ය ලෙස හඳුනා ගන්නා උපලක්ෂණයකි. මෙම අවස්ථාවේදී, **"Customer_ID"** යනු පාරිභෝගිකයා සඳහා වන ප්‍රධාන ගුණාංගයකි.

5. **Derived attribute / ව්‍යුත්පන්න උපලක්ෂණය**
   An attribute that can be calculated from other data. For example, **"Age"** can be calculated from the **"DOB"** (Date of Birth) of the customer, so it is called a derived attribute.
   වෙනත් දත්ත වලින් ගණනය කළ හැකි උපලක්ෂණයකි. උදාහරණයක් ලෙස, **"Age"** පාරිභෝගිකයාගේ **"DOB"** (උපන් දිනය) වලින් ගණනය කළ හැක, එබැවින් එය ව්‍යුත්පන්න උපලක්ෂණයක් ලෙස හැඳින්වේ.

The correct answer is 5).
නිවැරදි පිළිතුර 5) වේ.

Age is not stored directly but can be calculated from DOB.
වයස සෘජුවම ගබඩා කර නැති නමුත් DOB වලින් ගණනය කළ හැක.

7. What is cardinality in database relationships?
දත්ත සමුදාය සම්බන්ධතාවල ගණනියතාව යනු කුමක්ද?

1) The number of rows in a table / වගුවක ඇති පේළි ගණන.
2) The numerical relationship between entities / ඒකක අතර සංඛ්‍යාත්මක සම්බන්ධතාවය.
3) The number of columns in a table / වගුවක ඇති තීරු ගණන.
4) The size of the database / දත්ත සමුදායේ ප්‍රමාණය.
5) The number of indexes / දර්ශක ගණන.

Answer : 2)

Explanation / පැහැදිලි කිරීම

Cardinality in database relationships refers to the numerical relationship between two entities in an Entity-Relationship (ER) model. It describes how many instances of one entity are associated with instances of another entity. For example, in a one-to-many relationship, one instance of Entity A may relate to many instances of Entity B. Cardinality is crucial in designing databases as it determines the constraints and structure of relationships, ensuring accurate data representation.
දත්ත සමුදා සම්බන්ධතා වල ගණනියතාව යනු භූතාර්ථ-සම්බන්ධතා (ER) ආකෘතියක භූතාර්ථ දෙකක් අතර සංඛ්‍යාත්මක සම්බන්ධතාවයයි. එක් භූතාර්ථයක අවස්ථා කීයක් තවත් භූතාර්ථයක අවස්ථා සමඟ සම්බන්ධ වී ඇත්ද යන්න එය විස්තර කරයි. උදාහරණයක් වශයෙන්, ඒක-බහු සම්බන්ධතාවක් තුළ, භූතාර්ථ A හි එක් අවස්ථාවක් B භූතාර්ථයේ බොහෝ අවස්ථාවන්ට සම්බන්ධ විය හැක. දත්ත සමුදායන් සැලසුම් කිරීමේදී ගණනියතාව තීරනාත්මක වන්නේ එය සම්බන්ධතා වල බාධාවන් සහ ව්‍යුහය තීරණය කරන අතර නිවැරදි දත්ත නිරූපණය සහතික කරයි.

This concept helps establish clear rules for data interaction, such as whether an entity must be associated with one, many, or no instances of another entity. Understanding cardinality ensures that database relationships reflect real-world scenarios and support efficient data management.
මෙම සංකල්පය දත්ත අන්තර්ක්‍රියා සඳහා පැහැදිලි රීති පිහිටුවීමට උපකාර කරයි, එනම් භූතාර්ථයක් වෙනත් භූතාර්ථයක ඒක-බහු හෝ කිසිදු අවස්ථාවක් සමඟ සම්බන්ධ විය යුතුද යන්න වැනිය. ගණනියතාව අවබෝධ කර ගැනීමෙන් දත්ත සමුදා සම්බන්ධතා සැබෑ ලෝක අවස්ථා පිළිබිඹු කරන අතර කාර්යක්ෂම දත්ත කළමනාකරණයට සහාය වේ.

So, the correct answer number is 2).
එමනිසා, නිවැරදි පිළිතුරු අංකය 2) වේ.

8. What are the values of the following Python expressions?
පහත පයිතන් ප්‍රකාශනවල අගයන් මොනවාද?

```
= 2**(3**2)
= (2**3)**2
= 2**3**2
```

1) `512, 64, 512`   2) `512, 64, 64`   3) `512, 512, 512`
4) `64, 512, 64`   5) `64, 64, 64`

Answer : 1)

• Expression 1 is evaluated as: `2**9`, which is equal to 512.
1 වන ප්‍රකාශනය 512 ට සමාන වන `2**9` ලෙස ඇගයීමට ලක් කෙරේ.

• Expression 2 is evaluated as `8**2`, which is equal to 64.
2 වන ප්‍රකාශනය 64 ට සමාන වන `8**2` ලෙස ඇගයීමට ලක් කෙරේ.

• The last expression is evaluated as `2**(3**2)`.
අවසාන ප්‍රකාශනය `2**(3**2)` ලෙස ඇගයීමට ලක් කෙරේ.

This is because the associativity of ** operator is from right to left. Hence the result of the third expression is 512. මෙයට හේතුව ** මෙහෙයවනයේ ආශ්‍රය දකුණේ සිට වමට වන බැවිනි. එබැවින් තුන්වන ප්‍රකාශනයේ ප්‍රතිඵලය 512 වේ.

The correct answer is 1).
නිවැරදි පිළිතුර 1) වේ.

9. Which of the following is true about compilers?
සම්පාදකයන් සම්බන්ධයෙන් පහත සඳහන් දේවලින් සත්‍ය කුමක්ද?

1) <u>Converts the entire program into machine code before execution.</u>
සම්පූර්ණ වැඩසටහන ක්‍රියාත්මක කිරීමට පෙර යන්ත්‍ර කේතය බවට පරිවර්තනය කරයි.

2) Converts the program to machine code line by line.
වැඩසටහන යන්ත්‍ර කේතයට පේළියෙන් පේළියට පරිවර්තනය කරයි.

3) Converts the program into bytecode for execution.
ක්‍රියාත්මක කිරීම සඳහා වැඩසටහන බයිට්කේත බවට පරිවර්තනය කරයි.

4) Provides immediate feedback on syntax errors.
වාක්‍ය ඛණ්ඩ දෝෂ පිළිබඳ ක්ෂණික ප්‍රතිපෝෂණ සපයයි.

5) Interprets the program directly without generating an intermediate representation.
අතරමැදි නියෝජනයක් ජනනය නොකර වැඩසටහන සෘජුව අර්ථකථනය කරයි.

Answer : 1)

Explanation / පැහැදිලි කිරීම

1) True: A compiler translates the entire source code of a program into machine code (or an intermediate representation, such as assembly language) before execution begins. This means that the whole program is processed and converted into executable code in one go, and the resulting machine code can then be run independently of the source code.
නිවැරදියි : සම්පාදකයක් ක්‍රියාත්මක කිරීම ආරම්භ කිරීමට පෙර වැඩසටහනක සම්පූර්ණ මූල කේතය යන්ත්‍ර කේතය (හෝ එකලස් කිරීමේ භාෂාව වැනි අතරමැදි නිරූපණයක්) බවට පරිවර්තනය කරයි. මෙයින් අදහස් කරන්නේ සම්පූර්ණ ක්‍රමලේඛය එකවර ක්‍රියාත්මක කර ක්‍රියාත්මක කළ හැකි කේතයක් බවට පරිවර්තනය වන අතර, ප්‍රතිඵලයක් ලෙස ලැබෙන යන්ත්‍ර කේතය මූල කේතයෙන් ස්වාධීනව ධාවනය කළ හැකි බවයි.

2)False: This describes the behavior of an interpreter, not a compiler. An interpreter translates and executes code line by line, rather than converting the entire program into machine code at once.
වැරදියි: මෙය පරිවර්තකයෙකුගේ හැසිරීම විස්තර කරයි, සම්පාදකයෙකු නොවේ. පරිවර්තකයෙකු විසින් සම්පූර්ණ වැඩසටහන එකවර යන්ත්‍ර කේත බවට පරිවර්තනය කරනවාට වඩා පේළියෙන් කේත පරිවර්තනය කර ක්‍රියාත්මක කරයි.

3) False: This describes the behavior of some other programming tools, such as Java compilers, which translate source code into bytecode for execution on the Java Virtual Machine (JVM). This is not a universal feature of all compilers, as many compilers produce native machine code directly.
වැරදියි: මෙය Java Virtual Machine (JVM) මත ක්‍රියාත්මක කිරීම සඳහා ප්‍රභව කේතය බයිට්කේත බවට පරිවර්තනය කරන ජාවා සම්පාදක වැනි වෙනත් ක්‍රමලේඛන මෙවලම්වල හැසිරීම විස්තර කරයි. බොහෝ සම්පාදකයින් ස්වදේශීය යන්ත්‍ර කේතය සෘජුවම නිපදවන බැවින් මෙය සියලුම සම්පාදකයන්ගේ විශ්වීය ලක්ෂණයක් නොවේ.

4) False: This is more characteristic of interpreters or interactive environments (like some scripting languages). While some modern compilers can provide reasonably fast feedback during the compilation process, they typically compile the whole program and then report errors, rather than providing immediate feedback as the program is being written.
වැරදියි: මෙය පරිවර්තකයන්ට හෝ අන්තර්ක්‍රියාකාරී පරිසරයන්ට (සමහර ස්ක්‍රිප්ටින් භාෂා වැනි) වඩාත් ලක්ෂණයකි. සමහර නවීන සම්පාදකයින්ට සම්පාදනය කිරීමේ ක්‍රියාවලියේදී සාධාරණ ලෙස වේගවත් ප්‍රතිපෝෂණ ලබා දිය හැකි අතර, ඔවුන් සාමාන්‍යයෙන් සම්පූර්ණ වැඩසටහන සම්පාදනය කර පසුව දෝෂ වාර්තා කරයි, වැඩසටහන ලියන විට ක්ෂණික ප්‍රතිපෝෂණ ලබා දෙනවා වෙනුවට.

5) False: This describes the behavior of an interpreter, not a compiler. A compiler usually generates an intermediate representation (like assembly code or intermediate code) before producing the final machine code.
වැරදියි: මෙය පරිවර්තකයෙකුගේ හැසිරීම විස්තර කරයි, සම්පාදකයෙකු නොවේ. සම්පාදකයක් සාමාන්‍යයෙන් අවසාන යන්ත්‍ර කේතය නිෂ්පාදනය කිරීමට පෙර අතරමැදි නිරූපණයක් (එකලස් කිරීමේ කේතය හෝ අතරමැදි කේතය වැනි) ජනනය කරයි.

Therefore, the correct statement about compilers is answer 1).
එබැවින් සම්පාදක පිළිබඳ නිවැරදි ප්‍රකාශය පිළිතුර 1) වේ.

10. Consider the following Python code snippet.
පහත පයිතන් කේත කොටස සලකා බලන්න.

```
string = "hello world how are you"
result = []
for word in string.split():
    result.append(word[::-1])
print(" ".join(result))
```

What will be the output of this code?
පහත කේතයේ ප්‍රතිදානය කුමක්දැයි?

1) `olleh dlrow woh era uoy` 2) `hello world how are you`
3) `olleh world how are you` 4) `hello world woh era uoy`
5) None of the above

Answer : 1)

• **string = "hello world how are you":**
A string is defined with the value `"hello world how are you"`.
string එකක් `"hello world how are you"` ලෙස අර්ථ දක්වයි.

• **for word in string.split()::**
The string is split into individual words: `["hello", "world", "how", "are", "you"]`.
string එකක් තනි වචන වලට බෙදී ඇත: `["hello", "world", "how", "are", "you"]`.

• **`word[::-1]`:**
This reverses each word in the list `(e.g., "hello" becomes "olleh", "world" becomes "dlrow")`.
මෙය ලැයිස්තුවේ ඇති සෑම වචනයක්ම ආපසු හරවයි (උදා:`"hello", "olleh"` බවට පත් වේ,`"world", "dlrow"` බවට පත් වේ).

• **`result.append(word[::-1])`:**
Each reversed word is added to the result list.
සෑම ප්‍රතිලෝම වචනයක්ම ප්‍රතිඵල ලැයිස්තුවට එක් කෙරේ.

• **print(" ".join(result)):**
After all the words are reversed and added to list, the words are joined together with spaces and printed.
සියලුම වචන ආපසු හරවා ලැයිස්තුවට එකතු කළ පසු, වචන spaces සමඟ එකතු කර මුද්‍රණය කරයි.

Step-by-step processing of the words / වචනවල පියවරෙන් පියවර සැකසීම
`"hello" → "olleh"`
`"world" → "dlrow"`
`"how" → "woh"`
`"are" → "era"`
`"you" → "uoy"`

So, the output is: / මේ අනුව, ප්‍රතිදානය වන්නේ:
`olleh dlrow woh era uoy`