සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved]

Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 340 MARKING

2025/11/15
ictfromabc
Ravindu Bandaranayake

1. Who invented the logarithmic theories that used to easily multiply and divide numbers by doing addition and subtraction in 1617?
   1617 දී එකතු කිරීම සහ අඩු කිරීම මගින් පහසුවෙන් සංඛ්‍යා ගුණ කිරීමට සහ බෙදීමට භාවිතා කළ ලඝුගණක න්‍යායන් සොයාගත්තේ කවුරුන් විසින්ද?

   1) John Napier. / ජෝන් නේපියර්.
   2) William Oughtred. / විලියම් අවුට්‍රඩ්.
   3) Blaise Pascal. / බ්ලේස් පැස්කල්.
   4) Gottfried Wilhelm Leibniz. / ගොට්ෆ්‍රිඩ් විල්හෙල්ම් ලිබ්නිස්.
   5) Joseph Jacquard. / ජෝසප් ජැක්වාඩ්.

Answer : 1)

John Napier ජෝන් නේපියර්
In 1617 John Napier introduced logarithmic theory. Logarithms can easily add and subtract numbers, multiply and divide. Logarithmic tables should be used for that. Later he introduced an alternative method by recording the logarithms of ivory strips. These are called Napier's bones.
1617 දී ජෝන් නේපියර් විසින් ලඝුගණක සිද්ධාන්ත හඳුන්වා දෙන ලදී. ලඝුගණක මගින් සංඛ්‍යා පහසුවෙන් එකතුකිරීම සහ අඩුකිරීම මගින් ගුණ කිරීම සහ බෙදීම සිදුකළ හැකිය. ඒ සඳහා ලඝුගණක වගු භාවිතා කළ යුතුය. පසුව ඔහු විසින් ඇත්දළ තිරු වල ලඝුගණක සටහන් කරමින් විකල්ප ක්‍රමයක් හඳුන්වාදෙන ලදී. මේවා නේපියර්ගේ ඇත්දළ තිරු ලෙස හඳුන්වයි.

William Oughtred විලියම් අවුට්‍රඩ්
In 1632, William Audrey used his logarithmic concepts to create the slide rule in England.
1632 දී විලියම් අවුට්‍රඩ් විසින් ඔහුගේ ලඝුගණක සංකල්ප භාවිතා කරමින් එංගලන්තයේදී සර්පණ රූල නිර්මාණය කළේය.

Blaise Pascal බ්ලේස් පැස්කල්
In 1642, French mathematician Blaise Pascal invented the adding machine. It made addition and subtraction easy. This technology is still used in water meters, electricity meters and speedometers. This is known as the world's first mechanical calculator.
1642 දී ප්‍රංශ ගණිතඥයෙකු වූ බ්ලේස් පැස්කල් විසින් ආකලන යන්ත්‍රය නිර්මාණය කරන ලදී. එමගින් පහසුවෙන් එකතු කිරීම සහ අඩු කිරීම සිදුකරගත හැකි විය. ජල, විදුලි සහ වේග මාපකවල මෙම තාක්ෂණය තවමත් භාවිතා වේ. මෙය ලොව ප්‍රථම යාන්ත්‍රික ගණක යන්ත්‍රය ලෙස හඳුන්වයි.

Gottfried Wilhelm Leibniz ගොට්ෆ්‍රිඩ් විල්හෙල්ම් ලිබ්නිස්
In 1674, a German mathematician, Godfried Wilhelm Leibniz, improved the adding machine and made the stepped reckoner so that it could perform multiplication and division.
1674 දී ජර්මානු ජාතික ගණිතඥයෙක් වූ ගොඩ්ෆ්‍රිඩ් විල්හෙල්ම් ලිබ්නිස් විසින් ආකලන යන්ත්‍රය වැඩිදියුණු කර ගුණ කිරීම සහ බෙදීම සිදුකළ හැකි පරිදි පියවර ගණකය සැදුවේය.

Joseph Jacquard ජෝසප් ජැක්වාඩ්
Around 1800, Frenchman Joseph Jacquard invented the mechanical loom. Here he used punched cards to create different patterns on the fabric. These are called punch cards. This concept was later used as a secondary storage device in first generation computers.
1800 දී පමණ ප්‍රංශ ජාතික ජෝශප් ජැක්වාඩ් විසින් යාන්ත්‍රික රෙදි වියන යන්ත්‍රය නිර්මාණය කරන ලදී. මෙහිදී රෙදිවල විවිධ රටා නිර්මාණයට ඔහු සිදුරු කරන ලද කාඩ්පත් භාවිතා කරන ලදී. මේවා සිදුරුපත් ලෙස හැඳින්වේ. පසුකලෙක පළමු පරම්පරාවේ පරිගණකවල ද්විතියික ආචයන උපක්‍රමය ලෙස භාවිතා වුයේද මෙම සංකල්පයයි.

So the correct answer is John Napier.
ඒ අනුව නිවැරදි පිළිතුර වන්නේ ජෝන් නේපියර් වේ.

2. Which of the following cannot be considered as an advantage of artificial intelligence?
කෘත්‍රිම බුද්ධියේ වාසියක් ලෙස සැලකිය නොහැකි පහත සඳහන් මොනවාද?

1) Fast responses. / වේගවත් ප්‍රතිචාර.
2) Ability to mimic human behavior.
මිනිස් හැසිරීම් අනුකරණය කිරීමේ හැකියාව.
3) Ability to quickly analyze and comprehend large amounts of data.
විශාල දත්ත ප්‍රමාණයක් ඉක්මනින් විශ්ලේෂණය කර අවබෝධ කර ගැනීමේ හැකියාව.
4) Ability to follow the most appropriate course of action.
වඩාත් සුදුසු ක්‍රියාමාර්ග අනුගමනය කිරීමේ හැකියාව.
5) Raising legal and ethical issues.
නීතිමය හා සදාචාරාත්මක ගැටළු මතු කිරීම.

Answer : 5)

- Advantages of Artificial Intelligence \ කෘතිම බුද්ධියේ වාසි

Fast responses\ වේගවත් ප්‍රතිචාර:
AI can process information and respond to it much faster than a human can. This makes it particularly useful in situations where time is of the essence, such as in emergency response or financial trading.
AI හට තොරතුරු සැකසීමට සහ මිනිසාට වඩා වේගයෙන් ඒවාට ප්‍රතිචාර දැක්විය හැක. හදිසි ප්‍රතිචාරයක් හෝ මූල්‍ය වෙළඳාමක් වැනි කාලය ඉතා වැදගත් වන අවස්ථාවන්හිදී මෙය විශේෂයෙන් ප්‍රයෝජනවත් වේ.

Ability to mimic human behavior\මිනිස් හැසිරීම් අනුකරණය කිරීමේ හැකියාව:
Through machine learning and other techniques, AI can learn to mimic human behavior. This can be useful in a variety of contexts, from customer service (where AI can interact with customers in a human-like way) to healthcare (where AI can help to diagnose diseases by mimicking the diagnostic process of a doctor).
යන්ත්‍ර ඉගෙනීම සහ වෙනත් ශිල්පීය ක්‍රම හරහා, AI හට මිනිස් හැසිරීම් අනුකරණය කිරීමට ඉගෙන ගත හැක. මෙය පාරිභෝගික සේවයේ සිට AI හට පාරිභෝගිකයන් සමඟ මිනිසාට සමාන ආකාරයෙන් අන්තර් ක්‍රියා කළ හැකි) සෞඛ්‍ය සේවා දක්වා (වෛද්‍යවරයෙකුගේ රෝග විනිශ්චය ක්‍රියාවලිය අනුකරණය කිරීමෙන් රෝග හඳුනා ගැනීමට AI හට උපකාර කළ හැකි) විවිධ සන්දර්භයන් තුළ ප්‍රයෝජනවත් විය හැක.

Ability to quickly analyze and comprehend large amounts of data
විශාල දත්ත ප්‍රමාණයක් ඉක්මනින් විශ්ලේෂණය කිරීමට සහ ග්‍රහණය කර ගැනීමේ හැකියාව
AI can process and analyze large amounts of data much faster and more accurately than a human can. This can be particularly useful in fields like data analysis, where large amounts of data need to be processed and understood quickly.
AI හට මිනිසාට කළ හැකි ප්‍රමාණයට වඩා ඉතා වේගයෙන් සහ නිවැරදිව විශාල දත්ත ප්‍රමාණයක් සැකසීමට සහ විශ්ලේෂණය කිරීමට හැකිය. දත්ත විශ්ලේෂණය වැනි ක්ෂේත්‍රවල මෙය විශේෂයෙන් ප්‍රයෝජනවත් විය හැකි අතර, විශාල දත්ත ප්‍රමාණයක් ඉක්මනින් සැකසීමට සහ තේරුම් ගැනීමට අවශ්‍ය වේ.

Ability to follow the most appropriate course of action\ වඩාත්ම සුදුසු ක්‍රියාමාර්ග අනුගමනය කිරීමේ හැකියාව:
AI can be programmed to follow a specific set of rules or guidelines, ensuring that it always follows the most appropriate course of action. This can be particularly useful in fields like manufacturing, where consistency and precision are key.
AI නිශ්චිත රීති හෝ මාර්ගෝපදේශ මාලාවක් අනුගමනය කිරීමට වැඩසටහන්ගත කළ හැක, එය සැම විටම වඩාත් සුදුසු ක්‍රියාමාර්ග අනුගමනය කරන බව සහතික කරයි. අනුකූලතාව සහ නිරවද්‍යතාවය ප්‍රධාන වන නිෂ්පාදන වැනි ක්ෂේත්‍රවල මෙය විශේෂයෙන් ප්‍රයෝජනවත් විය හැකිය.

- Disadvantages of Artificial Intelligence \ කෘතිම බුද්ධියේ අවාසි

Raising legal and ethical issues\ නීතිමය සහ සදාචාරාත්මක ගැටළු මතු කිරීම:
While AI has many advantages, it also raises several legal and ethical issues. These can include concerns about privacy (as AI often involves the collection and use of personal data), job displacement (as AI can often perform tasks more efficiently than humans, potentially leading to job losses), and decision-making transparency (as it can be difficult to understand how an AI is making its decisions). These issues are not advantages of AI, but rather challenges that need to be addressed as we continue to develop and use this technology.

AI හි බොහෝ වාසි ඇති අතර, එය නීතිමය සහ සදාචාරාත්මක ගැටළු ගණනාවක් ද මතු කරයි. පුද්ගලිකත්වය (AI බොහෝ විට පුද්ගලික දත්ත එකතු කිරීම සහ භාවිතා කිරීම ඇතුළත් වන බැවින්), රැකියා විස්ථාපනය (AI හට බොහෝ විට මිනිසුන්ට වඩා කාර්යක්ෂම ලෙස කාර්යයන් ඉටු කළ හැකි බැවින්, රැකියා අහිමිවීම් වලට තුඩු දිය හැකි බැවින්) සහ තීරණ ගැනීමේ විනිවිදභාවය (එය හැකි පරිදි) පිළිබඳ උත්සුකයන් මේවාට ඇතුළත් විය හැකිය. AI එහි තීරණ ගන්නා ආකාරය තේරුම් ගැනීමට අපහසු වේ). මෙම ගැටළු AI හි වාසි නොව, අප මෙම තාක්ෂණය දියුණු කිරීම සහ භාවිතා කිරීම දිගටම කරගෙන යාමේදී විසඳිය යුතු අභියෝග වේ.

3. Consider the following statements about number system.
සංඛ්‍යා පද්ධතිය පිළිබඳ පහත ප්‍රකාශ සලකා බලන්න.

A. Binary Coded Decimal(BCD) uses 4bits to represent a character and some devices such as Digital voltmeter, frequency translator and Digital watch uses BCD to output information in decimal.
Binary Coded Decimal(BCD) අක්ෂරයක් නිරූපණය කිරීමට බිටු 4ක් භාවිතා කරන අතර ඩිජිටල් වෝල්ට්මීටරය, සංඛ්‍යාත පරිවර්තකය සහ ඩිජිටල් ඔරලෝසුව වැනි සමහර උපාංග දශමයෙන් තොරතුරු ප්‍රතිදානය කිරීමට BCD භාවිතා කරයි.

B. ASCII uses 8 bits to represent a character but doesn’t synchronize with digital encoding.
ASCII අක්ෂරයක් නිරූපණය කිරීමට බිටු 8ක් භාවිතා කරන නමුත් ඩිජිටල් කේතනය සමඟ සමමුහුර්ත නොවේ.

C. Unicode is used to represent different languages along the world and uses 16bits to represent a character.
යුනිකෝඩ් ලොව පුරා විවිධ භාෂා නියෝජනය කිරීමට භාවිතා කරන අතර අනුලක්ෂණයක් නියෝජනය කිරීමට බිටු 16ක් භාවිතා කරයි.

Which of the above is/are correct?
ඉහත ඒවායින් කුමක් නිවැරදිද?

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

Answer : 4)

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

- **Option / විකල්පය A)**
This is correct. BCD uses 4 bits to represent each decimal digit (0-9). It is commonly used in digital devices like digital voltmeters, frequency translators, and digital watches that need to display decimal numbers.
මෙය නිවැරදියි. එක් එක් දශම ඉලක්කම් (0-9) නියෝජනය කිරීමට BCD බිටු 4ක් භාවිතා කරයි. එය සාමාන්‍යයෙන් සංඛ්‍යාංක වෝල්ට්මීටර, සංඛ්‍යාත පරිවර්තක සහ දශම සංඛ්‍යා පෙන්වීමට අවශ්‍ය සංඛ්‍යාංක ඔරලෝසු වැනි ඩිජිටල් උපාංගවල භාවිතා වේ.

- **Option / විකල්පය B)**
This statement is partially incorrect. ASCII (American Standard Code for Information Interchange) uses 7 bits to represent a character, but it is often stored in 8 bits for simplicity. While ASCII doesn't directly synchronize with digital encoding standards like Unicode, the statement's claim about synchronization is vague and misleading.
මෙම ප්‍රකාශය අර්ධ වශයෙන් වැරදිය. ASCII (තොරතුරු හුවමාරුව සඳහා වූ ඇමරිකානු සම්මත කේතය) අක්ෂරයක් නිරූපණය කිරීමට බිටු 7ක් භාවිතා කරයි, නමුත් එය බොහෝ විට සරල බව සඳහා බිටු 8කින් ගබඩා කර ඇත. ASCII යුනිකෝඩ් වැනි ඩිජිටල් කේතිකරණ ප්‍රමිතීන් සමඟ සෘජුවම සමමුහුර්ත නොකරන අතර, සමමුහුර්තකරණය පිළිබඳ ප්‍රකාශයේ ප්‍රකාශය නොපැහැදිලි සහ නොමඟ යවන සුළුය.

- **Option / විකල්පය C)**
This is correct. Unicode is a character encoding standard that supports multiple languages and scripts around the world. It uses 16 bits (and sometimes more in the form of UTF-32) to represent a wide range of characters.
මෙය නිවැරදියි. යුනිකෝඩ් යනු ලොව පුරා විවිධ භාෂා සහ ස්ක්‍රිප්ට් සඳහා සහය දක්වන අක්ෂර කේතන සම්මතයකි. එය පුළුල් පරාසයක අක්ෂර නියෝජනය කිරීමට බිටු 16ක් (සහ සමහර විට UTF-32 ආකාරයෙන් වැඩි) භාවිතා කරයි.

Therefore the correct answer is **4)** A, C only / එබැවින් නිවැරදි පිළිතුර **4)** A,C පමණි

4. Consider the following boolean equations.
පහත බූලීය සමීකරණ සලකන්නග

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

Which of the above is/are correct ?
ඉහත කවරක් නිවැරදි වේද?

1) A only. A පමණි.	2) B only. B පමණි.	3) <u>A and B only.</u> A හා B පමණි.	4) All of the above. ඉහත සියල්ලම.	5) None of the above. ඉහත කිසිවක් නොවේ.

Answer : 3)

A. $A(A + B)(A + A.B) + (A + B)$

$(A.A + A.B)(A + A.B) + (A + B)$	Distributive law විඝටන න්‍යාය
$(A + A.B)(A + A.B) + (A + B)$	Idempotent law තද්වභාවී න්‍යාය
$(A + A.B) + (A + B)$	Idempotent law තද්වභාවී න්‍යාය
$A + A + A.B + B$	Commutative law න්‍යාදේශ න්‍යාය
$A + A.B + B$	Idempotent law තද්වභාවී න්‍යාය
$A + B$	Redundancy law සමතිරික්ත න්‍යාය

B. $\overline{AB + \overline{B}}$

$\overline{AB}.\overline{\overline{B}}$	De Morgan's law ද මුවාර් ප්‍රමේයය
$\overline{AB}.B$	Double inverse law ද්විත්ව ප්‍රතිලෝම න්‍යාය
$(\overline{A} + \overline{B}).B$	De Morgan's law ද මුවාර් ප්‍රමේයය
$\overline{A}.B + \overline{B}.B$	Distributive law විඝටන න්‍යාය
$\overline{A} + 0$	Inverse law ප්‍රතිලෝම න්‍යාය
$\overline{A}$	Identity law සර්වසාම්‍ය න්‍යාය

C. $AB = B+C$ Cannot simplify. සුළුකළ නොහැක.

Therefore, the answer is 3.
එබැවින් පිළිතුර 3 වේ.

5. Which of the following statements is/are correct about half adder?
අර්ධ ආකලක සම්බන්ධ පහත ප්‍රකාශන අතරින් නිවැරදි ප්‍රකාශ(ය) මොනවාද?

A. It has 2 inputs and 2 outputs.
එයට ආදාන 2ක් හා ප්‍රතිදාන 2ක් පවතී.

B. It has 3 inputs and 2 outputs.
එයට ආදාන 2ක් හා ප්‍රතිදාන 3ක් පවතී.

C. A full adder can be created by adding two half adders.
අර්ධ ආකලක දෙකක් එකතු කිරීමෙන් පූර්ණ ආකලකයක් නිර්මාණය කළ හැකිය.

1) A only. A පමණි.	2) B only. B පමණි.	3) C only. C පමණි.	4) <u>A and C only.</u> A හා C පමණි.	5) B and C only. B හා C පමණි.

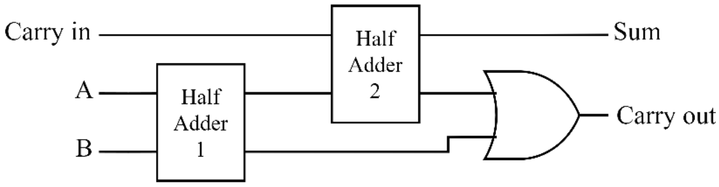
Answer : 4)

A half adder is a fundamental digital circuit that performs addition of two binary digits. It has two inputs. The outputs of a half adder are the sum and the carry out.
අර්ධ ආකලකය යනු ද්විමය ඉලක්කම් දෙකක් එකතු කිරීම සිදු කරන මූලික අංකිත පරිපථයකි. එයට ආදාන දෙකක් ඇත. අර්ධ ආකලකයක ප්‍රතිදානයන් sum සහ carry out ලෙස හඳුන්වයි.

A full adder is a digital circuit that performs addition of three binary digits. It has two inputs and a carry in from a previous stage. It produces two outputs. They are sum and carry out.
පූර්ණ ආකලකය යනු ද්විමය ඉලක්කම් තුනක් එකතු කිරීම සිදු කරන අංකිත පරිපථයකි. එයට සාමාන්‍ය ආදාන දෙකක් සහ carry in ආදානයක් ඇත. එය ප්‍රතිදානයන් දෙකක් නිපදවයි. ඒවා sum සහ carry out ලෙස හඳුන්වයි.

Adding two of these half adders can form a full adder. It can be done as follows.
මෙම අර්ධ ආකලක දෙකක් එකතු කිරීමෙන් පූර්ණ ආකලකයක් සෑදිය හැකිය. එය පහත පරිදි සිදුකළ හැකිය.

Note:
This circuit is out of syllabus. It is enough to remember that two half adders can be used to form a full adder.
මෙම පරිපථය විෂය නිර්දේශයෙන් බාහිර දෙයකි. අර්ධ ආකලක දෙකක් යොදාගෙන පූර්ණ ආකලකයක් සෑදිය හැකිබව පමණක් මතක තබා ගැනීම ප්‍රමාණවත් වේ.



Accordingly, only A and C statements are correct.
ඒ අනුව A හා C ප්‍රකාශ පමණක් නිවැරදිවේ.

6. Which of the following statements is correct about database models?
දත්ත සමුදා ආකෘතින් පිළිබඳ පහත සඳහන් ප්‍රකාශවලින් නිවැරදි වන්නේ කුමක්ද?

1) Relational model has data integrity issues.
සම්බන්ධතා ආකෘතියට දත්ත අඛණ්ඩතා ගැටළු ඇත.
2) Hierarchical model can handle any type of relationship.
ධූරාවලි ආකෘතියට ඕනෑම ආකාරයක සම්බන්ධතාවයක් හැසිරවිය හැක.
3) Network model provides data independence whereas hierarchical model does not.
ජාල ආකෘතිය දත්ත ස්වාධීනත්වය සපයන අතර ධූරාවලි ආකෘතිය එසේ නොවේ.
4) Network model allows to inherit objects such as tables.
ජාල ආකෘතිය වගු වැනි වස්තූන් උරුම කර ගැනීමට ඉඩ සලසයි.
5) Flat file systems allow to retrieve data efficiently.
ඒක ගොනු පද්ධති කාර්යක්ෂමව දත්ත ලබා ගැනීමට ඉඩ සලසයි.

Answer: 3)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
This is incorrect. The relational model is specifically designed to ensure data integrity through constraints like primary keys, foreign keys, and normalization.
මෙය වැරදියි. ප්‍රාථමික යතුරු, ආගන්තුක යතුරු සහ සාමාන්‍යකරණය වැනි සීමාවන් හරහා දත්ත අඛණ්ඩතාව සහතික කිරීම සඳහා සම්බන්ධතා ආකෘතිය විශේෂයෙන් නිර්මාණය කර ඇත.

Option / විකල්ප 2)
This is incorrect. The hierarchical model is limited to one-to-many relationships, making it inflexible for handling more complex relationships.
මෙය වැරදියි. ධූරාවලි ආකෘතිය ඒක-බහු සම්බන්ධතා වලට සීමා වී ඇති අතර, එය වඩාත් සංකීර්ණ සබඳතා හැසිරවීමට නම්‍යශීලී වේ.

Option / විකල්ප 3)
This is correct. The network model allows for greater data independence due to its ability to handle many-to-many relationships and a more flexible schema. The hierarchical model lacks this flexibility and ties data tightly to its structure.
මෙය නිවැරදියි. ජාල ආකෘතිය බහු-බහු සබඳතා හැසිරවීමේ හැකියාව සහ වඩාත් නම්‍යශීලී යෝජනා ක්‍රමයක් හේතුවෙන් වැඩි දත්ත ස්වාධීනත්වයකට ඉඩ සලසයි. ධූරාවලි ආකෘතියට මෙම නම්‍යශීලී බවක් නොමැති අතර දත්ත එහි ව්‍යුහයට තදින් සම්බන්ධ කරයි.

Option / විකල්ප 4)
This is incorrect. Inheritance is a feature of object-oriented or object-relational models, not the network model.
මෙය වැරදියි. උරුමය යනු වස්තු-නැඹුරු හෝ වස්තු-සම්බන්ධතා ආකෘතිවල ලක්ෂණයක් මිස ජාල ආකෘතිය නොවේ.

Option / විකල්ප 5)
This is incorrect. Flat file systems are not efficient for large-scale data retrieval since they lack indexing and querying capabilities found in database management systems.
මෙය වැරදියි. දත්ත සමුදා කළමනාකරණ පද්ධතිවල ඇති සුචිගත කිරීමේ සහ විමසුම් හැකියාවන් නොමැති බැවින් පැතලි ගොනු පද්ධති මහා පරිමාණ දත්ත ලබා ගැනීම සඳහා කාර්යක්ෂම නොවේ.

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

7. Which of the following correctly defines the cardinality and degree of a relation, respectively?
පහත සඳහන් දේවලින් පිළිවෙළින් සම්බන්ධතාවයක ගණනියතාව සහ මට්ටම (degree) නිවැරදිව නිර්වචනය කරන්නේ කුමක්ද?

1) Cardinality defines maximum participation of each entity, and Degree is the number of entities in a relationship.
ගණනියතාව එක් එක් භූතාර්තයේ උපරිම සහභාගීත්වය නිර්වචනය කරන අතර, මට්ටම යනු සම්බන්ධතාවයක ඇති භූතාර්ත ගණනයි.
2) Cardinality is the number of entity sets in a relationship, and Degree is the number of participating instances.
ගණනියතාව යනු සම්බන්ධතාවයක ඇති භූතාර්ත කට්ටල ගණන වන අතර, මට්ටම යනු සහභාගී වන අවස්ථා ගණනයි.
3) Cardinality is the number of attributes of each entity, and Degree is the number of entities.
ගණනියතාව යනු එක් එක් භූතාර්තයේ උපලක්ෂණ ගණන වන අතර, මට්ටම යනු භූතාර්ත ගණනයි.
4) Cardinality is the number of participating entity types, and Degree is the number of strong entities in a relationship.
ගණනියතාව යනු සහභාගී වන භූතාර්ත වර්ග ගණන වන අතර, මට්ටම යනු සම්බන්ධතාවයක ඇති ශක්තිමත් භූතාර්ත ගණන වේ.
5) Cardinality is the number of attributes, and Degree is the number of entity instances.
ගණනියතාව යනු උපලක්ෂණ ගණන වන අතර, මට්ටම යනු භූතාර්ත අවස්ථා ගණන වේ.

Answer: 1)

Explanation / පැහැදිලි කිරීම:

Cardinality refers to the maximum participation of each entity in the relationship, defined in terms like "one-to-one", "one-to-many" or "many-to-many."
ගණනියතාවය යන්නෙන් අදහස් කරන්නේ සම්බන්ධතාවයේ එක් එක් භූතාර්ථයේ උපරිම සහභාගීත්වයයි, එය "ඒක-ඒක", "ඒක-බහු" හෝ "බහු-බහු" වැනි වචන වලින් අර්ථ දක්වා ඇත.

Degree refers to the number of entities or entity sets involved in the relationship. For example, a relationship involving two entity sets has a degree of 2.
Degree යනු සම්බන්ධතාවයට සම්බන්ධ භූතාර්ථ ගණනයි. උදාහරණයක් ලෙස, භූතාර්ථ දෙකක් සම්බන්ධ සම්බන්ධතාවයකට degree 2ක් ඇත.

8. Which of the following is used to define a block of code in Python?
පයිතන් හි කේත බ්ලොක් එකක් නිර්වචනය කිරීමට පහත සඳහන් දේවලින් කුමක් භාවිතා කරයිද?

1) Curly braces. සඟල වරහන් {}
2) Parentheses. වරහන් ()
3) <u>Indentation.</u> ඉන්ඩෙන්ටේෂන්
4) Semicolon. ;
5) Square brackets. කොටු වරහන් []

Answer : 3)

The correct answer is 3) Indentation.
නිවැරදි පිළිතුර 3) ඉන්ඩෙන්ටේෂන්.

In Python, indentation is used to define the structure of a block of code. Unlike other programming languages that use curly braces `{}` or keywords to signify the start and end of code blocks, Python relies on consistent indentation (usually four spaces) to determine which statements are part of the same block. For example, in an `if` statement or a loop, all lines that are indented under the statement belong to the same block. If the indentation is inconsistent, Python will raise an error. This makes indentation a critical part of Python's syntax for defining blocks of code.
Python හි, කේත කොටසක ව්‍යුහය නිර්වචනය කිරීමට ඉන්ඩෙන්ටේෂන් භාවිතා කරයි. කේත බ්ලොක් වල ආරම්භය සහ අවසානය සනිටුහන් කිරීමට curly braces `{}` හෝ keywords භාවිතා කරන අනෙකුත් ක්‍රමලේඛන භාෂා මෙන් නොව, Python එකම වාරණයේ කොටසක් වන්නේ කුමන ප්‍රකාශයන්ද යන්න තීරණය කිරීම සඳහා ස්ථාවර ඉන්ඩෙන්ටේෂන් (සාමාන්‍යයෙන් හිස් හතරක්) මත රඳා පවතී. උදාහරණයක් ලෙස, if ප්‍රකාශයක හෝ ලූපයක, ප්‍රකාශය යටතේ ඇති සියලුම රේඛා එකම කොටසට අයත් වේ. ඉන්ඩෙන්ටේෂන් එක නොගැලපේ නම්, පයිතන් දෝෂයක් මතු කරයි. මෙය කේත කොටස් නිර්වචනය කිරීම සඳහා පයිතන්ගේ වාක්‍ය ඛණ්ඩයේ ඉන්ඩෙන්ටේෂන් තීරණාත්මක කොටසක් කරයි.

9. What is the output of the following python program?
පහත පයිතන් වැඩසටහනේ ප්‍රතිදානය කුමක්ද?

```
x = 10
def modify():
      x = 20
      print(x - 5, end=',')
      x += 5

print(x, end=',')
modify()
print(x)
```

1) `10,15,25`
2) <u>`10,15,10`</u>
3) `10,15,1`
4) `10,15,18`
5) `10,15,30`

Answer : 2)

Explanation / පැහැදිලි කිරීම

In this code, the variable `x` is initially set to `10` in the global scope. The `print(x, end=',')` statement prints the value of `x`, which is `10`, followed by a comma.
මෙම කේතයෙහි, `x` විචල්‍යය මුලින් ගෝලීය විෂය පථයේ `10` ලෙස සකසා ඇත. `print(x, end=',')` ප්‍රකාශය `x` හි අගය මුද්‍රණය කරයි, එය `10` වන අතර පසුව කොමාවකින්.

Next, the `modify()` function is called. Inside the function, a new local variable `x` is created and assigned the value `20`.
මීළඟට, `modify()` ශ්‍රිතය ලෙස හැඳින්වේ. ශ්‍රිතය තුළ `x` නව ස්ථානීය විචල්‍යයක් සාදා `20` අගය පවරනු ලැබේ.

This local x only exists within the function and does not affect the global x. The first `print(x - 5, end=',')` statement inside the function prints 20 - 5, which is 15, followed by a comma.
මෙම ස්ථානීය `x` ශ්‍රිතය තුළ පමණක් පවතින අතර ගෝලීය `x` ට බලපාන්නේ නැත. ශ්‍රිතය තුළ ඇති පළමු `print(x - 5, end=',')` ප්‍රකාශය `20 – 5` ලෙස මුද්‍රණය කරයි, එනම් `15` පසුව කොමාවක්.

Then, x is incremented by 5, making it 25, but this change remains local to the function.
එවිට, x 5 කින් වැඩි කර එය 25 බවට පත් කරයි, නමුත් මෙම වෙනස ශ්‍රිතයට ස්ථානීය පවතී.

Finally, after the `modify()` function call, the last `print(x)` prints the global variable x, which is still 10, because the modification inside the function did not affect the global value. So the final output will be **10,15,10.**
අවසාන වශයෙන්, `modify()` ශ්‍රිත ඇමතුමෙන් පසුව, අවසාන `print(x)` මඟින් ගෝලීය විචල්‍යය x මුද්‍රණය කරයි, එය තවමත් 10 වේ, මන්ද ශ්‍රිතය තුළ ඇති වෙනස් කිරීම ගෝලීය අගයට බල නොපායි. එබැවින් අවසාන ප්‍රතිදානය **10,15,10** වනු ඇත

Therefore the correct answer is 2)
එබැවින් නිවැරදි පිළිතුර 2)

10. What pattern will this Python code generate?
මෙම පයිතන් කේතය කුමන රටාවක් ජනනය කරයිද?

```
n = 5
i = 0
while i < n:
    print("*" * (n - i))
    i += 1
```

1)
```
*****
****
***
**
*
```
2)
```
*
**
***
****
*****
```
3)
```
* * * * *
```
4)
```
*
*
*
*
```
5) No output

Answer : 1)

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

`n = 5`
This sets n to 5, which will determine the height of the triangle (the number of lines).
මෙය n හි අගය 5 ට සකසයි, එය ත්‍රිකෝණයේ උස (පේළි ගණන) තීරණය කරනු ඇත.

`i = 0`
This initializes a variable i, which will be used to track the current line number.
මෙය i විචල්‍යයක් ආරම්භ කරයි, එය වත්මන් පේළි අංකය නිරීක්ෂණය කිරීමට භාවිතා කරනු ඇත.

`while i < n:`
The loop will run as long as i is less than n (i.e., while i is less than 5).
i, n ට වඩා අඩුවන තාක් (එනම්, i, 5 ට අඩු වන විට) ලූපය ක්‍රියාත්මක වේ.

`print("*" * (n - i))`
This line prints a string of asterisks. The number of asterisks printed is n - i. On each iteration, as i increases, the number of asterisks decreases.
මෙම පේළිය තරු ලකුණු මාලාවක් මුද්‍රණය කරයි. මුද්‍රණය කර ඇති තරු ලකුණු ගණන n - i වේ. එක් එක් පුනරාවර්තනයකදී, i වැඩි වන විට, තරු ලකුණු ගණන අඩු වේ.

`i += 1`
This increases the value of i by 1 after each iteration, so the loop eventually ends.
මෙය එක් එක් පුනරාවර්තනයට පසුව i හි අගය 1 කින් වැඩි කරයි, එබැවින් ලූපය අවසානයේ අවසන් වේ.

In the first iteration (i = 0), the code prints * repeated n - i = 5 - 0 = 5 times: *****.
In the second iteration (i = 1), it prints * repeated n - i = 5 - 1 = 4 times: ****.
This continues until i becomes 5, at which point the loop ends.
පළමු පුනරාවර්තනයේදී (i = 0), කේතය * නැවත නැවත n - i = 5 - 0 = 5 වතාවක් මුද්‍රණය කරයි: *****.
දෙවන පුනරාවර්තනයේ (i = 1), එය * නැවත නැවත n - i = 5 - 1 = 4 වතාවක් මුද්‍රණය කරයි: ****.
මෙය i, 5 බවට පත් වන තෙක් දිගටම පවතිනු ඇත, එම අවස්ථාවේදී ලූපය අවසන් වේ.

So, the correct answer number is 1. / එමනිසා, නිවැරදි පිළිතුරු අංකය 1 වේ.