ishÆ u ysñlï weúßKs /All Rights Reserved]

Ravindu Bandaranayake - ictfromabc

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 377 MARKING

2025/12/22
ictfromabc
Ravindu Bandaranayake

1. The invention of the "3 pin vacuum tube", laid the foundation for the modern computer technology. The person who is honored for this is?
"අග්‍ර තුනේ රික්තක නළය" සොයාගැනීම, නවීන පරිගණක තාක්ෂණය සඳහා අඩිතාලම දැමීය. මේ සඳහා ගෞරවයට පාත්‍ර වූ පුද්ගලයා කවුද?
   1) John Presper Eckert / ජෝන් ප්‍රෙස්පර් එකර්ට්
   2) Lee De Forest / ලී ද ෆොරස්ට්
   3) John V Atanasoff / ජෝන් වී ඇටනසොෆ්
   4) Gottfried Wilhelm / ගොට්ෆ්‍රිඩ් විල්හෙල්ම්
   5) Howard Aiken / හෝවාර්ඩ් අයිකන්

Answer : 2)

Explanation. / පැහැදිලි කිරීම.

Lee De Forest is credited with the invention of the vacuum tube (also known as the electronic valve). His work on the vacuum tube was crucial in the development of early electronic computers, as it allowed for the amplification and switching of electronic signals, which was essential for the functioning of these machines.
Lee De Forest රික්තක නළය (ඉලෙක්ට්‍රොනික කපාටය ලෙසද හැඳින්වේ) සොයාගැනීමේ ගෞරවය හිමි වේ. මෙම යන්ත්‍රවල ක්‍රියාකාරිත්වය සඳහා අත්‍යාවශ්‍ය වූ විද්‍යුත් සංඥා විස්තාරණය කිරීමට සහ මාරු කිරීමට එය ඉඩ සලසා දුන් බැවින්, රික්ත නළය පිළිබඳ ඔහුගේ කාර්යය මුල් කාලීන ඉලෙක්ට්‍රොනික පරිගණක සංවර්ධනය කිරීමේදී තීරණාත්මක විය.

Why the other options are incorrect.
අනෙක් විකල්ප වැරදි ඇයි.

1) John Presper Eckert
Along with John Mauchly, Eckert is known for designing the ENIAC, one of the earliest electronic general-purpose computers, but he did not invent the electronic valve.
John Mauchly සමඟ එක්ව Eckert පැරණිතම ඉලෙක්ට්‍රොනික පොදු කාර්ය පරිගණක වලින් එකක් වන ENIAC නිර්මාණය කිරීම සම්බන්ධයෙන් ප්‍රසිද්ධියක් උසුලන නමුත් ඔහු ඉලෙක්ට්‍රොනික කපාටය සොයා ගත්තේ නැත.

3) John V. Atanasoff
Known for creating the Atanasoff-Berry Computer (ABC), which was an early electronic computer, but he did not invent the electronic valve.
මුල් කාලයේ ඉලෙක්ට්‍රොනික පරිගණකයක් වූ Atanasoff-Berry Computer (ABC) නිර්මාණය කිරීම සඳහා ප්‍රසිද්ධියට පත් වූ නමුත් ඔහු ඉලෙක්ට්‍රොනික කපාටය සොයා ගත්තේ නැත.

4) Gottfried Wilhelm Von Leibniz
A philosopher and mathematician known for his work in calculus and binary number systems, not for the invention of the electronic valve.
දාර්ශනිකයෙක් සහ ගණිතඥයෙක් ඉලෙක්ට්‍රොනික කපාටය සොයා ගැනීම සඳහා නොව කලනය සහ ද්විමය සංඛ්‍යා පද්ධති පිළිබඳ ඔහුගේ වැඩ සඳහා ප්‍රසිද්ධය.

5) Howard Aiken
An American physicist and computer pioneer known for the Harvard Mark I computer, but he did not invent the electronic valve.
Harvard Mark I පරිගණකය සඳහා ප්‍රසිද්ධ ඇමරිකානු භෞතික විද්‍යාඥයෙක් සහ පරිගණක පුරෝගාමියෙක්, නමුත් ඔහු ඉලෙක්ට්‍රොනික කපාටය සොයා ගත්තේ නැත.

Therefore the correct answer is **2) Lee De Forest**
එබැවින් නිවැරදි පිළිතුර **2) Lee De Forest**

2. BCD equivalents of 10_{10} and 89_{10},
10_{10} ට සහ 89_{10} ට සමාන BCD අගයන් වන්නේ,

1) 1001 0000 , 0010 1001
2) 1010 1011 , 0011 1000
3) 0000 1010 , 1000 1001
4) 0001 0000 , 1000 1001
5) 1001 0001 , 1001 1001

Answer : 4)

In Binary Coded Decimal (BCD), each digit of a decimal number is represented separately by its 4-bit binary equivalent.
Binary Coded Decimal (BCD) හි, දශම සංඛ්‍යාවක සෑම ඉලක්කමක්ම එහි 4-bit ද්විමය සමානතාවයෙන් වෙන වෙනම නිරූපණය කෙරේ.

For 10, / 10 සඳහා,

The digits are 1 and 0. / ඉලක්කම් 1 සහ 0 වේ.

- 1 in BCD is 0001. / BCD හි 1 යනු 0001 කි.
- 0 in BCD is 0000. / BCD හි 0 යනු 0000 කි.

So, 10 in BCD is 0001 0000. / BCD හි 10 යනු 0001 0000 වේ.

For 89, / 89 සඳහා,

The digits are 8 and 9. / ඉලක්කම් 8 සහ 9 වේ.

- 8 in BCD is 1000. / BCD හි 8 යනු 1000 කි.
- 9 in BCD is 1001. / BCD හි 9 යනු 1001 කි.

So, 89 in BCD is 1000 1001. / ඉතින්, BCD හි 89 යනු 1000 1001 වේ.

So, the BCD equivalent of 10 and 89 is 0001 0000 and 1000 1001.
එබැවින්, 10 සහ 89 හි BCD සමාන වන්නේ 0001 0000 සහ 1000 1001 වේ.

3. The correct statement(s) about a Karnaugh map is/are,
කානෝ සිතියමක් සම්බන්ධයෙන් නිවැරදි ප්‍රකාශ(ය) වන්නේ,

A. Gray code is used for a Karnaugh map.
කානෝ සිතියමක් සඳහා Gray code භාවිතා කරයි.

B. When trivializing a Karnaugh map, an expression may not be trivial until the end.
කානෝ සිතියමක් මගින් සුළු කිරීමේදී ප්‍රකාශනයක් අවසානය දක්වා සුළු නොවීමට ඉඩ ඇත.

C. A Karnaugh map is created by customizing a truth table.
සත්‍යතා වගුවක් විශේෂයෙන් සකස්කිරීමෙන් කානෝ සිතියමක් නිර්මාණය කරයි.

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

Answer : 5)

Let's break down each statement,
එක් එක් ප්‍රකාශය බිඳ දමමු,

- **Statement / ප්‍රකාශය A)**
True, Gray code is indeed used to organize the Karnaugh map, which helps in identifying adjacent groups of cells that differ by only one bit.
සත්‍ය වේ, Gray code ඇත්ත වශයෙන්ම කානෝ සිතියම සංවිධානය කිරීමට භාවිතා කරයි, එය එක් බිට් එකකින් පමණක් වෙනස් වන යාබද සෛල කාණ්ඩ හඳුනා ගැනීමට උපකාරී වේ.

- **Statement / ප්‍රකාශය B)**
True, This means that during the process of simplifying the Karnaugh map, the expression may not appear simple until the entire process is complete.
සත්‍ය වේ, මෙයින් අදහස් කරන්නේ කානෝ සිතියම සරල කිරීමේ ක්‍රියාවලියේදී, සම්පූර්ණ ක්‍රියාවලිය සම්පූර්ණ වන තෙක් ප්‍රකාශනය සරල ලෙස නොපෙන්විය හැකි බවයි.

- **Statement / ප්‍රකාශය C)**
  True, A Karnaugh map is derived from a truth table and is used to simplify Boolean expressions.
  සත්‍ය වේ, කානෝ සිතියමක් සත්‍යතා වගුවකින් ව්‍යුත්පන්න වී ඇති අතර එය බූලියානු ප්‍රකාශන සරල කිරීමට භාවිතා කරයි.

Since all three statements are correct, the correct answer is indeed 5) All of the above.
ප්‍රකාශ තුනම නිවැරදි බැවින්, නිවැරදි පිළිතුර ඇත්තෙන්ම 5) ඉහත සියල්ලම වේ.

4. Attributes with values that must be requested and remembered with effort are called.......................
ආයාසයකින් ඉල්ලාගෙන මතක තබා ගත යුතු අගයන් සහිත උපලක්ෂණ ………………… නමින් හඳුන්වයි.
Choose the answer that applies to the blank.
හිස්තැනට අදාළ පිළිතුර තෝරන්න.
   1) Derived Attributes
   2) Descriptive Attributes
   3) Key Attributes
   4) <u>Stored Attributes</u>
   5) Composite Attributes

Answer : 4)

Stored Attributes ගබඩා කළ උපලක්ෂණ
Attributes with values that must be retrieved and remembered with effort are called stored attributes.
ආයාසයකින් ඉල්ලාගෙන මතක තබාගත යුතු අගයන් සහිත උපලක්ෂණ ගබඩා කල උපලක්ෂණ ලෙස හැඳින්වේ.

Derived Attributes ව්‍යුත්පන්න කළ උපලක්ෂණ
Attributes that can be derived from already stored attributes are called derived attributes.
දැනටමත් ගබඩා කරගෙන ඇති උපලක්ෂණ වලින් ව්‍යුත්පන්න කරගත හැකි උපලක්ෂණ ව්‍යුත්පන්න කල උපලක්ෂණ ලෙස හැඳින්වේ.

Descriptive Attributes විස්තරාත්මක උපලක්ෂණ
Attributes used to describe the relationship between two physical instances are called descriptive attributes.
භූතාර්ථ අවස්ථා දෙකක් අතර පවතින සම්බන්ධතාවය විස්තර කිරීමට භාවිත කරන උපලක්ෂණ විස්තරාත්මක උපලක්ෂණ ලෙස හැඳින්වේ.

Key Attributes යතුරු උපලක්ෂණ
Attributes used to distinguish physical instances are called key attributes.
භූතාර්ථ අවස්ථා වෙන් කර හඳුනා ගැනීම සඳහා භාවිත කරන උපලක්ෂණ යතුරු උපලක්ෂණ ලෙස හැඳින්වේ.

Composite Attributes සංයුක්ත උපලක්ෂණ
Attributes that can be further divided into parts are called composite attributes.
තවදුරටත් කොටස් වලට බෙදා වෙන්කළ හැකි උපලක්ෂණ සංයුක්ත උපලක්ෂණ ලෙස හැඳින්වේ.

5. In a database, what is denormalization?
දත්ත සමුදායක් තුළ, ප්‍රති-ප්‍රමතකරණය යනු කුමක්ද?
   1) The process of dividing a table into multiple smaller tables.
      වගුවක් කුඩා වගු කිහිපයකට බෙදීමේ ක්‍රියාවලිය.
   2) The process of merging tables into one table.
      වගු එක වගුවකට ඒකාබද්ධ කිරීමේ ක්‍රියාවලිය.
   3) The process of adding more indexes to speed up queries.
      විමසුම් වේගවත් කිරීම සඳහා තවත් දර්ශක එකතු කිරීමේ ක්‍රියාවලිය.
   4) <u>The process of converting a normalized schema into a less normalized one for optimization.</u>
      ප්‍රශස්තකරණය සඳහා ප්‍රමතකරණය වූ සටහනක් අඩු ප්‍රමතකරණයක් බවට පරිවර්තනය කිරීමේ ක්‍රියාවලිය.
   5) The process of creating new foreign key constraints between tables.
      වගු අතර නව ආගන්තුක යතුරු සම්බන්ධතා නිර්මාණය කිරීමේ ක්‍රියාවලිය.

Answer: 4)

Denormalization is considered the opposite of normalization, but they serve different purposes in database design.
ප්‍රමතකරණයේ ප්‍රතිවිරුද්ධ දෙය ලෙස ප්‍රති-ප්‍රමතකරණය සැලකේ, නමුත් ඒවා දත්ත සමුදාය නිර්මාණයේදී විවිධ අරමුණු ඉටු කරයි.

While normalization focuses on reducing redundancy and improving data integrity, denormalization reintroduces redundancy to improve query performance and simplify database design for specific use cases.
ප්‍රමතකරණය අතිරික්තය අඩු කිරීම සහ දත්ත අඛණ්ඩතාව වැඩි දියුණු කිරීම කෙරෙහි අවධානය යොමු කරන අතර, විමසුම් කාර්ය සාධනය වැඩි දියුණු කිරීමට සහ විශේෂිත භාවිත අවස්ථා සඳහා දත්ත සමුදා නිර්මාණය සරල කිරීමට ප්‍රති-ප්‍රමතකරණය අතිරික්තතාව නැවත හඳුන්වා දෙයි.

While it can have some advantages, the main result of denormalization is an increase in data redundancy.
එය යම් වාසි තිබිය හැකි වුවද, ප්‍රති-ප්‍රමතකරණයෙහි ප්‍රධාන ප්‍රතිඵලය වන්නේ දත්ත අතිරික්තය වැඩි වීමයි.

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
Incorrect. This describes normalization, not denormalization.
වැරදියි. මෙය ප්‍රමතකරණය විස්තර කරයි, ප්‍රති-ප්‍රමතකරණය නොවේ.

Option / විකල්ප 2)
Incorrect. Although denormalization may involve combining tables, it is done for performance optimization, not just merging. This option is too simplistic.
වැරදියි. ප්‍රති-ප්‍රමතකරණයට වගු ඒකාබද්ධ කිරීම ඇතුළත් විය හැකි වුවද, එය ඒකාබද්ධ කිරීම පමණක් නොව කාර්ය සාධන ප්‍රශස්තිකරණය සඳහා සිදු කෙරේ. මෙම විකල්පය ඉතා සරල ය.

Option / විකල්ප 3)
Incorrect. Adding indexes improves query performance but is unrelated to denormalization.
වැරදියි. දර්ශක එකතු කිරීම විමසුම් කාර්ය සාධනය වැඩි දියුණු කරයි නමුත් ප්‍රති-ප්‍රමතකරණයට සම්බන්ධ නොවේ.

Option / විකල්ප 4)
Correct. This is the correct definition of denormalization.
නිවැරදියි. මෙය ප්‍රති-ප්‍රමතකරණයේ නිවැරදි අර්ථ දැක්වීමයි.

Option / විකල්ප 5)
Incorrect. Foreign key constraints are part of normalization and ensuring referential integrity, not denormalization.
වැරදියි. ආගන්තුක යතුරු සම්බාධක ප්‍රමතකරණයේ සහ යොමු අඛණ්ඩතාව සහතික කිරීමේ කොටසකි, ප්‍රති-ප්‍රමතකරණය නොවේ.

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

6. Which SQL query will return the names of employees whose age is between 25 and 30 inclusive?
වයස අවුරුදු 25 සහ 30 අතර සේවකයින්ගේ නම් ලබා දෙන්නේ කුමන SQL විමසුම මගින් ද?
   1) `SELECT EmployeeName FROM Employees WHERE Age >= 25 OR Age <= 30;`
   2) `SELECT EmployeeName FROM Employees WHERE Age BETWEEN 25 AND 30;`
   3) `SELECT EmployeeName FROM Employees WHERE Age > 25 OR Age < 30;`
   4) `SELECT EmployeeName FROM Employees WHERE Age IN (25, 30);`
   5) `SELECT EmployeeName FROM Employees WHERE Age BETWEEN 25 TO 30;`

Answer : 2)

Explanation / පැහැදිලි කිරීම

The BETWEEN 25 AND 30 condition includes both boundary values. Option 1) doesn't work. Option 3) excludes ages 25 and 30. Option 4) only matches ages exactly equal to 25 or 30, and option 5) has incorrect syntax.

BETWEEN 25 AND 30 අතර කොන්දේසියට මායිම් අගයන් දෙකම ඇතුළත් වේ. විකල්ප 1) ක්‍රියා නොකරයි. විකල්ප 3) වයස අවුරුදු 25 සහ 30 බැහැර කරයි (excludes). විකල්ප 4) ගැලපෙන්නේ හරියටම වයස 25 හෝ 30 ට සමාන වන අතර, විකල්ප 5) වැරදි වාක්‍ය ඛණ්ඩයක් වේ.

7. What is the SQL statement to delete all records in the BOOK table as Author's data Williums?
BOOK වගුවේ Author දත්ත Williums ලෙස ඇති සියලුම වාර්තා මකා දැමීමට SQL ප්‍රකාශය කුමක්ද?
   1) `DELETE WHERE Book.Author = Williums;`
   2) `DELETE 'Williums' FROM Book;`
   3) `DELETE * FROM Book WHERE Author = 'Williums';`
   4) `DELETE FROM Book WHERE Author = 'Williums';`
   5) `DELETE * FROM Author WHERE Book = 'Williums';`

Answer: 4)

Let's examine each option / එක් එක් විකල්පය පරීක්ෂා කරමු.

1) **`DELETE WHERE Book.Author = Williums;`**
Incorrect. The DELETE statement in SQL needs a FROM clause to specify which table to delete from. Additionally, string literals like 'Williums' must be enclosed in single quotes.
වැරදියි. SQL හි DELETE ප්‍රකාශය, මකා දැමිය යුත්තේ කුමන වගුවෙන්ද යන්න සඳහන් කිරීමට FROM වගන්තියක් අවශ්‍ය වේ. මීට අමතරව, 'Williums' වැනි වචන වචන තනි උද්ධෘතවල අමුණා තිබිය යුතුය.

2) **`DELETE ‘Williums’ FROM Book;`**
The syntax is incorrect. The DELETE statement does not use the string literal directly. Instead, it specifies the table from which rows are deleted and uses the WHERE clause to filter which rows to delete.
වාක්‍ය Syntax වැරදියි. DELETE ප්‍රකාශය තන්තුව සෘජුවම භාවිතා නොකරයි. ඒ වෙනුවට, එය පේළි මකා දැමිය යුතු වගුව සඳහන් කරන අතර මකා දැමිය යුතු පේළි පෙරීමට WHERE වගන්තිය භාවිතා කරයි.

3) **`DELETE * FROM Book WHERE Author = ‘Williums’;`**
Incorrect. The DELETE statement does not use the * wildcard. The * is used in SELECT statements to indicate all columns but is not used with DELETE.
DELETE ප්‍රකාශය * Wildcard භාවිතා නොකරයි. වැරදියි. සියලුම තීරු දැක්වීමට SELECT ප්‍රකාශවල * භාවිතා වන නමුත් DELETE සමඟ භාවිතා නොවේ.

4) **`DELETE FROM Book WHERE Author = ‘Williums’;`**
Correct. This is the correct syntax for deleting rows from a table based on a condition. Here, DELETE FROM Book specifies that rows should be deleted from the BOOK table, and the WHERE Author = 'Williums' condition ensures that only rows where the Author column matches 'Williums' will be deleted.
නිවැරදියි. කොන්දේසියක් මත පදනම්ව වගුවකින් පේළි මකා දැමීම සඳහා නිවැරදි වාක්‍ය ඛණ්ඩය මෙයයි. මෙහිදී, DELETE FROM Book යන්නෙන් දැක්වෙන්නේ පොත් වගුවෙන් පේළි මකා දැමිය යුතු බවත්, WHERE Author = ‘Williums’ කොන්දේසිය මඟින් Author තීරුව Williums සමඟ ගැලපෙන පේළි පමණක් මකාදැමීම සහතික කරන බවත්ය.

5) **`DELETE * FROM Author WHERE Book = ‘Williums’;`**
This statement is incorrect because it mistakenly refers to an Author table and uses an invalid column reference. You need to delete from the BOOK table and check the Author column for 'Williums'.
මෙම ප්‍රකාශය වැරදි ලෙස කර්තෘ වගුවකට යොමු වීම සහ වලංගු නොවන තීරු යොමුවක් භාවිතා කිරීම නිසා එය වැරදිය. ඔබ පොත් වගුවෙන් මකා දැමිය යුතු අතර 'Williums' සඳහා කර්තෘ තීරුව පරීක්ෂා කල යුතුය.

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

8.
```
List0 = [2,4,6,8,10]
List1 = List0
List0.append(List1.pop())
List1[-1] = 0
print(List1 + List0)
```

Imagine the 2nd code line of the above python code was replaced by `List1 = List0.copy()`. What will be the new output generated?
ඉහත පයිතන් කේතයේ 2 වැනි කේත රේඛාව `List1 = List0.copy()` මගින් ප්‍රතිස්ථාපනය කර ඇතැයි සිතන්න. උත්පාදනය වන නව ප්‍රතිදානය කුමක් වනු ඇත්ද?

1) `[2, 4, 6, 8, 0] [2, 4, 6, 8, 0]`
2) `[2, 4, 6, 0] [2, 4, 6, 8, 10, 10]`
3) `[2, 4, 6, 0, 2, 4, 6, 8, 10, 10]`
4) `[2, 4, 6, 8, 0, 2, 4, 6, 8, 0]`
5) `Error`

Answer: 3)

When the second line of the code is replaced with List1 = List0.copy(), the behavior of the program changes significantly because List1 now becomes a separate copy of List0. This means List1 and List0 will have the same initial values, but they are independent lists, and changes to one will not affect the other.
කේතයේ දෙවන පේළිය List1 = List0.copy() සමඟ ප්‍රතිස්ථාපනය කළ විට, List1 දැන් List0 හි වෙනම පිටපතක් බවට පත්වන බැවින් වැඩසටහනේ හැසිරීම සැලකිය යුතු ලෙස වෙනස් වේ. මෙයින් අදහස් කරන්නේ List1 සහ List0 එකම ආරම්භක අගයන් ඇත, නමුත් ඒවා ස්වාධීන ලැයිස්තු වන අතර එකකට වෙනස් කිරීම් අනෙකට බලපාන්නේ නැත.

Let's analyze step by step / පියවරෙන් පියවර විශ්ලේෂණය කරමු.

Line / පේළිය 1:
`List0` is initialized:
`List0` ආරම්භ කර ඇත:

```
List0 = [2, 4, 6, 8, 10]
```

Line / පේළිය 2:
Creates a separate list from List0 and assign it to List1:
List0 වෙතින් වෙනම ලැයිස්තුවක් සාදා එය List1 වෙත පවරයි:

```
List1 = [2, 4, 6, 8, 10]
```

Line / පේළිය 3:
List1.pop() removes the last element (10) from List1 and returns it. Then, List0.append(10) appends this value to List0.
List1.pop() විසින් List1 වෙතින් අවසාන මූලාංගය (10) ඉවත් කර එය ආපසු ලබා දෙයි. එවිට, List0.append(10) මෙම අගය List0 වෙත එකතු කරයි.

```
List0 = [2, 4, 6, 8, 10, 10]
List1 = [2, 4, 6, 8]
```

Line / පේළිය 4:
List1[-1] = 0 modifies the last element of List1 to 0. Since List1 is independent of List0, this does not affect List0.
List1[-1] = 0 විසින් List1 හි අවසාන මූලාංගය 0 ලෙස වෙනස් කරයි. List1, List0 වෙතින් ස්වාධීන බැවින්, මෙය List0 ට බලපාන්නේ නැත.

```
List0 = [2, 4, 6, 8, 10, 10]
List1 = [2, 4, 6, 0]
```

Line / පේළිය 5:
The statement `print(List1 + List0)` concatenates the two lists and prints the result.
ප්‍රකාශය `print(List1 + List0)` ලැයිස්තු දෙක ඒකාබද්ධ කර ප්‍රතිඵලය මුද්‍රණය කරයි.

```
[2, 4, 6, 0, 2, 4, 6, 8, 10, 10]
```

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

9.
```
List=[9,7,4,2,1]
for x in List:
    if ((13//6)**x+9/6 != 5.5):
        print(x)
    else:
        break
```

What will be the output of the above Python code?
ඉහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් වේද?

1)	2)	3)	4)	5)
9	9 7	9 7 <u>4</u>	9 7 4 2	9 7 4 2 1

Answer: 3)

`List = [9, 7, 4, 2, 1]`: Initializes a list named List with the elements 9, 7, 4, 2, and 1.
`List = [9, 7, 4, 2, 1]`: 9, 7, 4, 2, සහ `1` මූලාංගය සමඟ `List` නම් ලැයිස්තුවක් ආරම්භ කරයි.

The loop iterates over each element in the list `List`.
ලූපය `List` ලැයිස්තුවේ එක් එක් මූලාංගය මත පුනරාවර්තනය වේ.

In each iteration / එක් එක් පුනරාවර්තනය තුළ:

- It calculates the expression `((13 // 6) ** x + 9 / 6)`.
  එය `((13 // 6) ** x + 9 / 6)` ප්‍රකාශනය ගණනය කරයි.
- If this expression is not equal to `5.5`, it prints the value of `x`.
  මෙම ප්‍රකාශනය `5.5`ට සමාන නොවේ නම්, එය `x` හි අගය මුද්‍රණය කරයි.
- If the expression is equal to `5.5`, it breaks out of the loop.
  ප්‍රකාශනය `5.5`ට සමාන නම්, එය ලූපයෙන් පිටතට යයි.

When `x=9`, `x=7` and `x=4` ; `((13 // 6) ** x + 9 / 6) != 5.5` becomes `True`. But when `x=2` it becomes `False`. Then the code segment under `else`, `break`, is executed. So, the program comes out of the loop.
`x=9`, `x=7` සහ `x=4` ; `((13 // 6) ** x + 9 / 6) != 5.5` යන්න `True` වෙයි. නමුත් `x=2` වූ විට එය `False` වෙයි. එවිට `else` යටතේ ඇති කේත කොටස, `break`, ක්‍රියාත්මක වේ. එබැවින්, වැඩසටහන ලූපයෙන් පිටතට පැමිණේ.

So, the output of the above Python code is:
එබැවින්, ඉහත Python කේතයේ ප්‍රතිදානය වන්නේ:
```
9
7
4
```

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

10. A teacher is preparing a webpage with examples of numbers in exponential form, using chemical notation for molecules, and emphasizing key points.
ගුරුවරයෙක් බලය ලබා දිමේ ආකාරයෙන් සංඛ්‍යා සඳහා උදාහරණ සහිත වෙබ් පිටුවක් සකස් කරමින්, අණු සඳහා රසායනික අංකනය භාවිතා කරමින් සහ ප්‍රධාන කරුණු අවධාරණය කරමින් සිටියි.

What HTML tags will help in writing "H_2O", "x^2" and **`highlighting key phrases`**?
"H_2O", "x^2" ලිවීමට සහ **ප්‍රධාන වාක්‍ය ඛණ්ඩ ඉස්මතු කිරීමට** උපකාරී වන HTML උසුලන මොනවාද?

1) <u>`<sub>, <sup>, <strong>`</u>
2) `<underline>, <small>, <bold>`
3) `<i>, <subscript>, <strong>`
4) `<sup>, <sub>, <b>`
5) `<strong>, <italic>, <sup>`

Answer: 1)

- `<sub>` is used for subscript, placing text slightly below the baseline (e.g., the "2" in H_2O).
`<sub>` subscript එකක් සඳහා භාවිතා කරයි, පාඨ මූලික රේඛාවට තරමක් පහළින් තබයි.
(උදා: H_2O හි "2").

- `<sup>` is used for superscript, placing text slightly above the baseline (e.g., the "2" in x^2).
`<sup>` superscript එකක් සඳහා භාවිතා කරයි, පාඨ මූලික රේඛාවට තරමක් ඉහළින් තබයි.
(උදා: x^2 හි "2").

- `<strong>` is used for emphasizing important text, making it bold and adding semantic importance to it.
`<strong>` වැදගත් පාඨ අවධාරණය කිරීමට, එය තද කිරීමට සහ එයට අර්ථකථන වැදගත්කමක් එක් කිරීමට භාවිතා කරයි.

Option 1) is correct because it matches the required formatting for chemical notation, exponential forms, and highlighted phrases.
රසායනික අංකනය, බලය ලබා දිමේ ආකෘති සහ උද්දීපනය කළ වාක්‍ය ඛණ්ඩ සඳහා අවශ්‍ය හැඩතල ගැන්වීමට එය ගැලපෙන බැවින් විකල්ප 1) නිවැරදි වේ.