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 387 MARKING

2026/01/01
ictfromabc
Ravindu Bandaranayake

1. Consider the following statements.
පහත ප්‍රකාශ සලකන්න.

A. Integrated circuits were used to develop the third generation of computers.
තුන්වන පරම්පරාවේ පරිගණක නිපදවීමට අනුකලිත පරිපථ යොදාගන්නා ලදී.

B. First generation computers used vacuum tubes and magnetic drums for main memory.
පළමු පරම්පරාවේ පරිගණක වල රික්තක නළ හා ප්‍රධාන මතකය සඳහා චුම්භක බෙර භාවිත කරන ලදී.

C. The world's first general purpose digital computer was the IBM 704.
ලොව ප්‍රථම පොදු කාර්යය අංකිත පරිගණකය IBM 704 වේ.

Which statement(s) are true?
සත්‍ය ප්‍රකාශ(ය) කුමක් ද?

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

- **Statement / ප්‍රකාශය A)**
This statement is true. The third generation of computers, developed during the 1960s, saw the use of integrated circuits (ICs). These circuits replaced transistors, which were used in the second generation, allowing computers to become more efficient, reliable, and compact.
මෙම ප්‍රකාශය සත්‍ය වේ. 1960 දී නිපදවන ලද තුන්වන පරම්පරාවේ පරිගණක, ඒකාබද්ධ පරිපථ (ICs) භාවිතා කරන ලදී. මෙම පරිපථ දෙවන පරම්පරාවේ භාවිතා කරන ලද ට්‍රාන්සිස්ටර ප්‍රතිස්ථාපනය කරන ලද අතර, පරිගණක වඩාත් කාර්යක්ෂම, විශ්වාසනීය සහ සංයුක්ත වීමට ඉඩ සලසයි.

- **Statement / ප්‍රකාශය B)**
This statement is true. First-generation computers, developed in the late 1940s to early 1950s, relied on vacuum tubes for processing and magnetic drums as a form of main memory. These machines were large, slow, and generated a lot of heat.
මේ ප්‍රකාශය සත්‍ය වේ. පළමු පරම්පරාවේ පරිගණක, 1940 අග භාගයේ සිට 1950 මුල් භාගයේ දී සංවර්ධනය කරන ලද අතර, ප්‍රධාන මතකයේ ආකාරයක් ලෙස සැකසීම සඳහා රික්තක නල සහ චුම්බක බෙර මත රඳා පැවතුනි. මෙම යන්ත්‍ර විශාල, මන්දගාමී වූ අතර විශාල තාපයක් ජනනය විය.

- **Statement / ප්‍රකාශය C)**
This statement is false. The IBM 704 was not the first general-purpose digital computer. The title of the first general-purpose digital computer is often attributed to the ENIAC (Electronic Numerical Integrator and Computer), developed in the 1940s, not the IBM 704.
මෙම ප්‍රකාශය අසත්‍ය වේ. IBM 704 යනු පළමු පොදු කාර්ය ඩිජිටල් පරිගණකය නොවේ. පළමු පොදු කාර්ය ඩිජිටල් පරිගණකයේ නාමය බොහෝ විට ENIAC (Electronic Numerical Integrator and Computer) වෙත ආරෝපණය කර ඇත, එය IBM 704 නොවේ.

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

2. Given the 8-bit 2's complement binary number 1110 1101, what is its decimal equivalent?
8 bit 2 හි අනුපූරක ද්විමය අංකය 1110 1101 ලබා දී ඇති අතර, එහි දශම සමානය කුමක්ද?

1) -19 2) -35 3) -29 4) -27 5) -39

Answer : 1)

To find the decimal equivalent of the 8-bit 2's complement binary number 11101101, follow these steps:
8-bit 2 හි අනුපූරක ද්විමය අංක 11101101 හි දශම තුල්‍ය සොයා ගැනීමට, මෙම පියවර අනුගමනය කරන්න.

1 Identify if it's a negative number / එය සෘණ අංකයක් දැයි හඳුනා ගැනීම.

The leftmost bit (1) indicates that it's a negative number since it's 1 in 2's complement.
වම්පස ඇති බිටුව (1) පෙන්නුම් කරන්නේ එය 2 හි අනුපූරකයේ 1 වන බැවින් එය සෘණ අංකයක් බවයි.

2 Find the 2's complement / 2 හි අනුපූරකය සොයන්න.

- Invert the bits: 11101101 becomes 00010010.
  බිටු ප්‍රතිලෝමය : 11101101 = 00010010 බවට පත් වේ.
- Add 1 to this inverted result: 00010010 + 1 = 00010011 (which is 19 in decimal).
  මෙම ප්‍රතිලෝම ප්‍රතිඵලයට 1 ක් එකතු කරන්න: 00010010 + 1 = 00010011 (එය දශමයෙන් 19 කි).

3 Make it negative / එය සෘණ අගයක් කරන්න

Since it was a negative number originally, the decimal equivalent is -19.
එය මුලින් සෘණ අංකයක් වූ බැවින්, දශම තුල්‍ය වන්නේ -19 වේ.

So, 11101101 in 2's complement is -19 in decimal.
එබැවින්, 11101101 හි 2 හි අනුපූරකය දශමයෙන් -19 වේ.

3. Simplify the given logic statement.
ලබා දී ඇති තාර්කික ප්‍රකාශනය සුළු කරන්න.

$B+0(\overline{CC} + CD)$

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

Answer : 1)

$1B + 0(\overline{CC} + CD)$	
$1B + 0(\overline{C} + \overline{C} + CD)$	Demorgen theorem ද මුවාර් ප්‍රමේය
$1B + 0$	Identity law සර්වසාම්‍ය න්‍යාය
$1B$	Identity law සර්වසාම්‍ය න්‍යාය
B	Identity law සර්වසාම්‍ය න්‍යාය

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

4. Which type of database is designed to handle large volumes of structured and unstructured data, often used for applications like data warehousing and big data analytics?
දත්ත ගබඩා කිරීම සහ විශාල දත්ත විශ්ලේෂණ වැනි යෙදුම් සඳහා බොහෝ විට භාවිතා කරන ව්‍යුහගත සහ ව්‍යුහගත නොවන දත්ත විශාල පරිමාවක් හැසිරවීමට සැලසුම් කර ඇත්තේ කුමන ආකාරයේ දත්ත සමුදායක්ද?

1) Relational database
සම්බන්ධතා දත්ත සමුදාය
2) NoSQL database
NoSQL දත්ත සමුදාය
3) Object-oriented database
වස්තු-නැඹුරු දත්ත සමුදාය
4) Hierarchical database
ධුරාවලි දත්ත සමුදාය
5) Network database
ජාල දත්ත සමුදාය

Answer : 2)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
Designed for structured data and uses a schema with rows and tables. It is not ideal for handling large volumes of unstructured or semi-structured data.
ව්‍යුහගත දත්ත සඳහා නිර්මාණය කර ඇති අතර පේළි සහ වගු සහිත යෝජනා ක්‍රමයක් භාවිතා කරයි. එය ව්‍යුහගත නොකළ හෝ අර්ධ ව්‍යුහගත දත්ත විශාල පරිමාවක් හැසිරවීමට සුදුසු නොවේ.

Option / විකල්ප 2)
Designed to handle large volumes of structured, semi-structured, and unstructured data. It is commonly used in applications like big data analytics, data warehousing, and scalable web applications. Examples include MongoDB, Cassandra, and Hadoop.
ව්‍යුහගත, අර්ධ ව්‍යුහගත සහ ව්‍යුහගත නොවන දත්ත විශාල ප්‍රමාණයක් හැසිරවීමට නිර්මාණය කර ඇත. විශාල දත්ත විශ්ලේෂණ, දත්ත ගබඩා කිරීම සහ පරිමාණය කළ හැකි වෙබ් යෙදුම් වැනි යෙදුම්වල එය බහුලව භාවිතා වේ. උදාහරණ ලෙස MongoDB, Cassandra, and Hadoop ඇතුළත් වේ.

Option / විකල්ප 3)
Integrates object-oriented programming concepts, storing data as objects, but is not optimized for handling large volumes of diverse data types.
වස්තු-නැඹුරු ක්‍රමලේඛන සංකල්ප ඒකාබද්ධ කරයි, වස්තු ලෙස දත්ත ගබඩා කරයි, නමුත් විවිධ දත්ත වර්ගවල විශාල පරිමාවන් හැසිරවීමට ප්‍රශස්ත නොවේ.

Option / විකල්ප 4)
Uses a tree-like structure to represent data relationships. It is rigid and not suitable for big data or unstructured data.
දත්ත සම්බන්ධතා නියෝජනය කිරීමට ගසක් වැනි ව්‍යුහයක් භාවිතා කරයි. එය දැඩි වන අතර විශාල දත්ත හෝ ව්‍යුහගත නොවන දත්ත සඳහා සුදුසු නොවේ.

Option / විකල්ප 5)
Uses a graph structure to show many-to-many relationships. While powerful for certain applications, it is not typically used for big data analytics.
බොහෝ සම්බන්ධතා පෙන්වීමට ප්‍රස්ථාර ව්‍යුහයක් භාවිතා කරයි. ඇතැම් යෙදුම් සඳහා බලවත් වුවද, එය සාමාන්‍යයෙන් විශාල දත්ත විශ්ලේෂණ සඳහා භාවිතා නොවේ.

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

5. Which of the following statement is true about following SQL query?
පහත SQL විමසුම සම්බන්ධයෙන් පහත සඳහන් කුමන ප්‍රකාශය සත්‍යද?

```
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100) NOT NULL,
    Price DECIMAL CHECK (Price > 0),
    Stock INT DEFAULT 0 );
```

1) Ensures Price can be negative or null.
Price සෘණ හෝ ශූන්‍ය විය හැකි බව සහතික කරයි.
2) Ensures ProductName can be null.
ProductName ශූන්‍ය විය හැකි බව සහතික කරයි.
3) Ensures Price must always be greater than zero.
Price සෑම විටම බිංදුවට වඩා වැඩි විය යුතු බව සහතික කරයි.
4) Ensures Stock must always have a value.
Stock වලට සැමවිටම අගයක් තිබිය යුතු බව සහතික කරයි.
5) Ensures ProductID can have duplicate values.
ProductID ට අනුපිටපත් අගයන් තිබිය හැකි බව සහතික කරයි.

Answer: 3)

Explanation / පැහැදිලි කිරීම:

The given SQL query creates a table Products where ProductID is unique and cannot have duplicates, ProductName must have a value and cannot be null, Price is restricted to values greater than zero with a CHECK constraint and Stock has a default value of 0 if no value is provided.
ලබා දී ඇති SQL විමසුම මඟින් ProductID අනන්‍ය සහ අනුපිටපත් තිබිය නොහැකි Products වගුවක් නිර්මාණය කරයි, ProductName ට අගයක් තිබිය යුතු අතර එය ශූන්‍ය විය නොහැක, CHECK සම්බාධකයක් සමඟින් Price බිංදුවට වඩා වැඩි අගයන්ට සීමා වේ, සහ Stock සඳහා අගයක් දී නොමැති නම්, පෙරනිමි අගය 0 වේ.

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
Incorrect. The CHECK (Price > 0) constraint ensures Price must always be greater than zero, so it cannot be negative or null.
වැරදියි. CHECK (Price > 0) සම්බාධකය මිල සෑම විටම බිංදුවට වඩා වැඩි විය යුතු බව සහතික කරයි, එබැවින් එය සෘණ හෝ ශූන්‍ය විය නොහැක.

Option / විකල්ප 2)
Incorrect. The NOT NULL constraint ensures ProductName cannot be null.
වැරදියි. NOT NULL සම්බාධකය මඟින් නිෂ්පාදන නාමය ශූන්‍ය විය නොහැකි බව සහතික කරයි.

Option / විකල්ප 3)
Correct. The CHECK (Price > 0) constraint enforces this rule.
නිවැරදියි. CHECK (Price > 0) සම්බාධකය මෙම රීතිය බලාත්මක කරයි.

Option / විකල්ප 4)
Incorrect. The Stock column has a default value of 0, but this does not enforce a requirement to explicitly provide a value.
වැරදියි. Stock තීරුවේ පෙරනිමි අගය 0 ඇත, නමුත් මෙය පැහැදිලිවම අගයක් සැපයීමේ අවශ්‍යතාවයක් බලාත්මක නොකරයි.

Option / විකල්ප 5)
Incorrect. The PRIMARY KEY constraint ensures ProductID must be unique and cannot have duplicates.
වැරදියි. PRIMARY KEY සම්බාධකය ProductID අනන්‍ය විය යුතු අතර අනුපිටපත් තිබිය නොහැකි බව සහතික කරයි.

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

6. Which combination of constraints ensures that a column in a table can never be null and must always contain a unique value?
වගුවක ඇති තිරුවක් කිසිවිටෙක ශුන්‍ය විය නොහැකි බවත් සෑම විටම අනන්‍ය අගයක් අඩංගු විය යුතු බවත් සහතික කරන්නේ කුමන සම්බාධක සංයෝජනයද?

1) `NOT NULL and UNIQUE` 2) `UNIQUE and CHECK` 3) `NOT NULL and DEFAULT`
4) `PRIMARY KEY and CHECK` 5) `FOREIGN KEY and UNIQUE`

Answer: 1)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
This is correct. The NOT NULL constraint ensures that the column cannot have null (empty) values. The UNIQUE constraint ensures that all values in the column are distinct. Together, these constraints ensure that the column always contains a non-null and unique value.
මෙය නිවැරදිය. NOT NULL සීමාව මඟින් තිරුවට ශුන්‍ය (හිස්) අගයන් තිබිය නොහැකි බව සහතික කරයි. UNIQUE සීමාව තිරුවේ ඇති සියලුම අගයන් වෙනස් බව සහතික කරයි. මෙම සීමාවන් එක්ව, තිරුවේ සෑම විටම ශුන්‍ය නොවන සහ අනන්‍ය අගයක් අඩංගු බව සහතික කරයි.

Option / විකල්ප 2)
This is incorrect. The UNIQUE constraint ensures distinct values, but the CHECK constraint enforces a logical condition (ex: CHECK (Value > 0)). While this combination can restrict values, it does not guarantee non-null values.
මෙය වැරදිය. UNIQUE සීමාව වෙනස් අගයන් සහතික කරයි, නමුත් CHECK සීමාව තාර්කික කොන්දේසියක් බලාත්මක කරයි (උදා: CHECK (Value > 0) මෙම සංයෝජනයට අගයන් සීමා කළ හැකි අතර, එය ශුන්‍ය නොවන අගයන් සහතික නොකරයි.

Option / විකල්ප 3)
This is incorrect. The NOT NULL constraint prevents null values, and the DEFAULT constraint provides a fallback value if none is provided during insertion. This combination does not ensure uniqueness.
මෙය වැරදියි. NOT NULL සීමාව ශුන්‍ය අගයන් වළක්වයි, සහ ඇතුල් කිරීමේදී කිසිවක් ලබා නොදෙන්නේ නම් DEFAULT සීමාව පසුබැසීමේ අගයක් සපයයි. මෙම සංයෝජනය සුවිශේෂත්වය සහතික නොකරයි.

Option / විකල්ප 4)
This is incorrect. The PRIMARY KEY constraint already enforces both NOT NULL and UNIQUE implicitly. Adding a CHECK constraint may add additional logical conditions but does not contribute to ensuring non-null and unique values independently.
මෙය වැරදිය. PRIMARY KEY සීමාව දැනටමත් NOT NULL සහ UNIQUE යන දෙකම ව්‍යංගයෙන් බලාත්මක කරයි. CHECK සීමාවක් එකතු කිරීම අමතර තාර්කික කොන්දේසි එකතු කළ හැකි නමුත් ස්වාධීනව ශුන්‍ය නොවන සහ අද්විතීය අගයන් සහතික කිරීමට දායක නොවේ.

Option / විකල්ප 5)
This is incorrect. A FOREIGN KEY ensures referential integrity, meaning the column values must match primary key values in another table. Adding UNIQUE ensures no duplicates in the column but does not prevent null values.
මෙය වැරදියි. FOREIGN KEY යොමු කිරීමේ අඛණ්ඩතාව සහතික කරයි, එනම් තිරු අගයන් වෙනත් වගුවක ප්‍රාථමික යතුරු අගයන් සමඟ ගැළපිය යුතුය. UNIQUE එකතු කිරීම තිරුවේ අනුපිටපත් නොමැති බව සහතික කරන නමුත් ශුන්‍ය අගයන් වලක්වන්නේ නැත.

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

7. What will the output be?
ප්‍රතිදානය කුමක් වේද?

```
WM  = "WWE Best Player"
Player = " John Cena"
Shot = "U Can't See Me"
print(Player); print(Shot*2)
```

1)
```
 John Cena
U Can't See MeU Can't See Me
```

2)
```
 John Cena
U Can't See Me U Can't See Me
```

3)
```
JohnCena
U Can't See MeU Can't See Me
```

4)
```
John Cena
U Can't See Me
U Can't See Me
```

5)
```
John Cena
U Can't See Me   U Can't See Me
```

Answer : 1)

In this Python code, there are two `print` statements.
මෙම පයිතන් කේතයේ, `print` ප්‍රකාශ දෙකක් ඇත.

1. `print(Player)`
This prints the value of the variable `Player`, which is "`John Cena`", so the first output is,
මෙය `Player` විචල්‍යයේ අගය මුද්‍රණය කරයි, එය "`John Cena`" වේ, එබැවින් පළමු ප්‍රතිදානය වන්නේ,

- `John Cena`

2. `print(Shot*2)`
The `` `*2` `` repeats the string "`U Can't See Me`" two times, so it prints,
`` `*2` `` මඟින් "`U Can't See Me`" string දෙවරක් පුනරාවර්තනය කරයි, එබැවින් එය මුද්‍රණය කරයි,

- `U Can't See MeU Can't See Me`

```
John Cena
U Can't See MeU Can't See Me
```

8. Consider the below python code.
පහත පයිතන් කේතය සලකා බලන්න.

```
Class= {'Physics':95,'ICT':85,'Maths':78}
total_students=list(Class.values())
print(total_students)
```

Which of the following answer contains the output of the above python code?
ඉහත පයිතන් කේතයේ ප්‍රතිදානය අඩංගු වන්නේ පහත කුමන පිළිතුරේද?

1) `['Physics', 'ICT', 'Maths']`
2) `[95, 85, 78]`
3) `['Physics', 'Maths', 'ICT']`
4) `'Physics', 'ICT', 'Maths'`
5) `95, 85, 78`

Answer : 2)

The given Python code creates a dictionary called **`Class`**, where the **keys** are subject names, and the values are student marks. The `list(Class.values())` function retrieves the **values (the marks)** from the dictionary and converts them into a list. Finally, the **`print(total_students)`** function outputs the list of values.
ලබා දී ඇති පයිතන් කේතය **`Class`** නමින් dictionary එකක් නිර්මාණය කරයි, එහි **keys** වනුයේ විෂය නම් වන අතර අගයන් ශිෂ්‍ය ලකුණු වේ. `list(Class.values())` ශ්‍රිතය dictionary එකේ ඇති අගයන් (ලකුණු) ලබාගෙන ඒවා ලැයිස්තුවක් බවට පරිවර්තනය කරයි. අවසාන වශයෙන්, **`print(total_students)`** ශ්‍රිතය අගයන් ලැයිස්තුව ප්‍රතිදානය කරයි.

Keys: `'Physics', 'ICT', 'Maths'`
Values: `95, 85, 78`

The code retrieves only the **values (the marks)** and prints them in a list format.
කේතය අගයන් (ලකුණු) පමණක් ලබාගෙන ඒවා ලැයිස්තු ආකෘතියකින් මුද්‍රණය කරයි.

- **Option / විකල්ප 1)**
  Incorrect statement. This contains the keys of the dictionary, but the code is retrieving the values.
  වැරදි ප්‍රකාශයකි. මෙහි dictionary එකක keys අඩංගු වේ, නමුත් කේතය අගයන් ලබා ගනී.

- **Option / විකල්ප 2)**
  Correct statement. This is the correct list of values [95, 85, 78] which the code outputs.
  නිවැරදි ප්‍රකාශයකි. කේතය ප්‍රතිදානය කරන [95, 85, 78] අගයන් වල නිවැරදි ලැයිස්තුව මෙයයි.

- **Option / විකල්ප 3)**
  Incorrect statement. This is also a list of keys, but the order is irrelevant because the code retrieves the values.
  වැරදි ප්‍රකාශයකි. මෙයද keys ලැයිස්තුවකි, නමුත් කේතය අගයන් (values) ලබා ගන්නා බැවින් අනුපිළිවෙල අදාල නොවේ.

- **Option / විකල්ප 4)**
  Incorrect statement. This looks like a tuple of keys, but the code is retrieving the values, not the keys.
  වැරදි ප්‍රකාශයකි. මෙය keys තොගයක් මෙන් පෙනේ, නමුත් කේතය නැවත ලබා ගන්නේ values මිස keys නොවේ.

- **Option / විකල්ප 5)**
  Incorrect statement. This is a tuple of values, but the correct output is a list.
  වැරදි ප්‍රකාශයකි. මෙය values කිහිපයක් සහිත tuple එකක් වේ. නමුත් නිවැරදි ප්‍රතිදානය ලැයිස්තුවකි.

Correct Answer is **2) [95, 85, 78].**
නිවැරදි පිළිතුර **2) [95, 85, 78]** වේ.

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

```
l = ['castle',['beach',2003],'umbrella']
l[1].append(2004)
print(l)
```

1) `['castle', 2004, 'umbrella']`
2) `['castle', ['beach', 2003, 2004], 'umbrella']`
3) `['castle', ['beach', 2003], 2004, 'umbrella']`
4) `['castle', [2004,'beach', 2003], 'umbrella']`
5) `['castle', ['beach', 2003, 2004]]`

Answer : 2)

Let’s consider each code line one by one.
අපි එක් එක් කේත පේළින් එකින් එක සලකා බලමු.

- `l = ['castle', ['beach', 2003],'umbrella']`

  This creates a list named `l` with three elements.
  මෙය මුලාංග තුනක් සහිත `l` නම් ලැයිස්තුවක් නිර්මාණය කරයි.

  (i) The first element is a string `'castle'`.
  පළමු මුලාංගය 'castle' නම් string එකයි.

  (ii) The second element is another list `['beach', 2003]`.
  දෙවන මුලාංගය `['beach', 2003]` නම් තවත් ලැයිස්තුවකි.

(iii) The third element is a string `'umbrella'`.
තුන්වන මූලාංගය `'umbrella'` string එකක් වේ.

- **`l[1].append(2004)`**
  This accesses the second element of the list `l`, which is the `sublist ['beach', 2003]`, and appends the number 2004 to it. Now, the `sublist` becomes `['beach', 2003, 2004].`
  මෙය `l` ලැයිස්තුවේ දෙවන මූලාංගයට ප්‍රවේශ වන අතර එය `sublist ['beach', 2003]` වන අතර, එයට 2004 අංකය එකතු කරයි. දැන්, `sublist` එක `sublist ['beach', 2003]` බවට පත් වේ.

- **`print(l)`**
  This prints the entire list `l`, which now looks like this.
  මෙය මුළු `l` ලැයිස්තුවම මුද්‍රණය කරයි, එය දැන් මේ ආකාරයට පෙනේ.

```
['castle', ['beach', 2003, 2004], 'umbrella']
```

So, the correct answer is 2).
එබැවින්, නිවැරදි පිළිතුර 2).

10. Which HTML tag from following does not require an end tag?
පහත දැක්වෙන කුමන HTML උසුලනයට අවසන් උසුලනයක් අවශ්‍ය නොවේද?

1) `<font>` 2) `<title>` 3) `<ol>` 4) `<br>` 5) `<b>`

Answer : 4)

This question asks which HTML tag does not require an end tag. In HTML, most tags have opening and closing tags, but some are self-closing or void elements, meaning they do not require a separate closing tag.
මෙම ප්‍රශ්නය අසන්නේ කුමන HTML උසුලනයද අවසන් උසුලනයක් අවශ්‍ය නොවන බවයි. HTML හි, බොහෝ උසුලන වලට විවෘත සහ වසා දැමීමේ උසුලනයක් ඇත, නමුත් සමහරක් ස්වයං-වසා දැමීමේ හෝ හිස් මූලාංග වේ, එනම් ඒවාට වෙනම වසා දැමීමේ උසුලනයක් අවශ්‍ය නොවේ.

- **Option / විකල්ප 1)**

`<font>` requires an end tag (`</font>`) to define the end of the font style.
`<font>` අකුරු විලාසයේ අවසානය අර්ථ දැක්වීමට අවසන් උසුලනයක් (`</font>`) අවශ්‍ය වේ.

- **Option / විකල්ප 2)**

`<title>` requires an end tag (`</title>`) to indicate the title of the document.
`<title>` ලේඛනයේ මාතෘකාව දැක්වීමට අවසන් උසුලනයක් (`</title>`) අවශ්‍ය වේ.

- **Option / විකල්ප 3)**

`<ol>` requires an end tag (`</ol>`) to close an ordered list.
`<ol>` අනුපිළිවෙළක් සහිත ලැයිස්තුවක් වැසීමට අවසන් උසුලනයක් (`</ol>`) අවශ්‍ය වේ.

- **Option / විකල්ප 4)**

`<br>` is a line break tag and does not require a closing tag because it is a void element.
`<br>`යනු රේඛා බිඳීමේ උසුලනයක් වන අතර එය හිස් මූලාංගයක් බැවින් වසා දැමීමේ උසුලනයක් අවශ්‍ය නොවේ.

- **Option / විකල්ප 5)**

`<b>` requires an end tag (`</b>`) to stop bold formatting.
`<b>` තද හැඩතල ගැන්වීම නැවැත්වීමට අවසන් උසුලනයක් (`</b>`) අවශ්‍ය වේ.

So, the correct answer is Option 4, as `<br>` does not require an end tag.
මේ අනුව, නිවැරදි පිළිතුර වන්නේ විකල්ප 4 වන අතර, `<br>` අවසන් උසුලනයක් අවශ්‍ය නොවන බැවිනි.