සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved ]

Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu

Information and Communication Technology

තොරතුරු හා සන්නිවේදන තාක්ෂණය

# 2026 QUIZ NO 310 MARKING

2025/10/16
ictfromabc
Ravindu Bandaranayake

1. Exascale computing refers to the ability to perform at least $10^{18}$ operations per second. ................................ is the world's first exascale supercomputer.
Exascale පරිගණනය යනු තත්පරයකට අවම වශයෙන් මෙහෙයුම් $10^{18}$ ක් සිදු කිරීමේ හැකියාවයි.
................................යනු ලොව ප්‍රථම Exascale සුපිරි පරිගණකයයි.

1) Frontier 2) Titan 3) Summit 4) Fugaku 5) Aurora

**Answer: 1)**

Let's explain the statements in the above question.
ඉහත ප්‍රශ්නයේ ඇති ප්‍රකාශ මෙසේ පැහැදිලි කරමු.

**Frontier**
Frontier, installed at the Oak Ridge National Laboratory, holds the distinction of being the first supercomputer to achieve exascale performance, thus meeting the $10^{18}$ operations per second benchmark.
Oak Ridge ජාතික රසායනාගාරයේ ස්ථාපනය කර ඇති Frontier Exascale කාර්ය සාධනය ලබා ගත් ප්‍රථම සුපිරි පරිගණකය ලෙස විශිෂ්ඨත්වය දරයි, එමඟින් තත්පරයට $10^{18}$ මෙහෙයුම් මිණුම් ලකුණක් සපුරාලයි.

**Titan**
Titan, previously among the world's fastest supercomputers, was operated by the Oak Ridge National Laboratory (ORNL). It was designed to perform at petascale levels, handling $10^{15}$ operations per second.
මීට පෙර ලොව වේගවත්ම සුපිරි පරිගණක අතර වූ Titan Oak Ridge National Laboratory මගින් ක්‍රියාත්මක විය. එය තත්පරයකට මෙහෙයුම් $10^{15}$ ක් හසුරුවන පෙටස්කේල් මට්ටමින් ක්‍රියා කිරීමට සැලසුම් කර ඇත.

**Summit**
Summit, another supercomputer developed by ORNL, succeeded Titan. Summit has been recognized for its exceptional performance, but it operates at a petascale level, achieving a peak performance of approximately 200 petaflops.
ORNL විසින් නිපදවන ලද තවත් සුපිරි පරිගණකයක් වන Summit Titan ගෙන් පසු පත් විය. සමුලුව එහි සුවිශේෂී කාර්ය සාධනය සඳහා පිළිගෙන ඇත, නමුත් එය දළ වශයෙන් petaflops 200 ක උපරිම කාර්ය සාධනයක් සාක්ෂාත් කර ගනිමින් petascale මට්ටමකින් ක්‍රියාත්මක වේ.

**Fugaku**
Fugaku, a supercomputer developed by RIKEN and Fujitsu in Japan, was the world's fastest supercomputer as of 2020. Although it showcased impressive computational capabilities, it did not reach the exascale threshold.
ජපානයේ RIKEN සහ Fujitsu විසින් නිපදවන ලද සුපිරි පරිගණකයක් වන Fugaku 2020 වන විට ලොව වේගවත්ම සුපිරි පරිගණකය විය. එය සිත් ඇදගන්නාසුළු ගණනය කිරීමේ හැකියාවන් පෙන්නුම් කළද, එය exascale සීමාවට ළඟා වූයේ නැත.

**Aurora**
Aurora, an anticipated supercomputer, is being developed by Intel and Cray for the Argonne National Laboratory. It is expected to achieve exascale performance, but it has not yet reached operational status.
Aurora සුපිරි පරිගණකයක්, Intel සහ Cray විසින් Argone ජාතික රසායනාගාරය සඳහා සංවර්ධනය කරමින් පවතී. එය විශිෂ්ට කාර්ය සාධනයක් ලබා ගැනීමට අපේක්ෂා කරන නමුත් එය තවමත් මෙහෙයුම් තත්ත්වයට පැමිණ නැත.

So, the answer is 1).
එබැවින්, පිළිතුර 1) වේ.

2. What can be considered as data validation method/methods from followings?
පහත දක්වා ඇති දත්ත වලංගු කිරීමේ ක්‍රමය/ක්‍රම ලෙස සැලකිය හැක්කේ මොනවාද?

A. Range check. පරාස පරීක්ෂාව.
B. Type check වර්ග පරීක්ෂාව
C. Direct Check සෘජු පරීක්ෂාව
D. Presence check ඇති/නැති බව පරීක්ෂාව

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

**Answer: 5)**

The question is asking which of the provided options are considered valid methods of data validation. Data validation is a process used to ensure that the data entered into a system is accurate, complete, and reasonable. It checks if the input data conforms to predefined rules or constraints.
ප්‍රශ්නය වන්නේ දත්ත වලංගු කිරීමේ වලංගු ක්‍රම ලෙස සලකනු ලබන්නේ සපයන ලද විකල්පයන් මොනවාද යන්නයි. දත්ත වලංගුකරණය යනු පද්ධතියකට ඇතුළත් කළ දත්ත නිවැරදි, සම්පූර්ණ සහ සාධාරණ බව සහතික කිරීම සඳහා භාවිතා කරන ක්‍රියාවලියකි. ආදාන දත්ත පූර්ව නිශ්චිත රීති හෝ සීමාවන්ට අනුකූලදැයි එය පරීක්ෂා කරයි.

A. **Range check / පරාස පරීක්ෂාව**
Ensures the data is within a specific range. For example, if age is required to be between 18 and 60, the range check will verify if the input age falls within this range.
දත්ත නිශ්චිත පරාසයක් තුළ ඇති බව සහතික කරයි. උදාහරණයක් ලෙස, වයස අවුරුදු 18 සහ 60 අතර වීමට අවශ්‍ය නම්, ආදාන වයස මෙම පරාසය තුළට වැටේ දැයි පරාස පරීක්ෂාව මගින් සත්‍යාපනය කරනු ඇත.

B. **Type check / වර්ග පරීක්ෂාව**
Verifies that the data is of the correct data type (ex: numeric, string, or date). For example, a type check ensures that a field for a phone number contains only numbers.
දත්ත නිවැරදි දත්ත වර්ගයට අයත් බව තහවුරු කරයි (උදා: සංඛ්‍යාත්මක, තන්තු, හෝ දිනය). උදාහරණයක් ලෙස, දුරකථන අංකයක් සඳහා වන ක්ෂේත්‍රයක ඇත්තේ අංක පමණක් බව ටයිප් චෙක්පතක් සහතික කරයි.

C. **Direct check / සෘජු පරීක්ෂාව**
This is not a standard term used in data validation, which is why it is not considered a correct answer.
මෙය දත්ත වලංගුකරණයේදී භාවිතා වන සම්මත පදයක් නොවේ, එය නිවැරදි පිළිතුරක් ලෙස නොසැලකේ.

D. **Presence check / ඇති/නැති බව පරීක්ෂාව**
Ensures that a required field has not been left empty. For example, if a form requires an email address, a presence check ensures that the field is filled before submission.
අවශ්‍ය ක්ෂේත්‍රයක් හිස්ව තබා නොමැති බව සහතික කරයි. උදාහරණයක් ලෙස, පෝරමයකට විද්‍යුත් තැපැල් ලිපිනයක් අවශ්‍ය නම්, ඉදිරිපත් කිරීමට පෙර ක්ෂේත්‍රය පුරවා ඇති බවට ඇති/නැත පරීක්ෂාව සහතික කරයි.

The correct answer is 5) A, B, and D, meaning Range check, Type check, and Presence check are valid methods of data validation.
නිවැරදි පිළිතුර 5) A, B සහ D එනම් පරාසය පරීක්ෂා කිරීම, වර්ගය පරීක්ෂා කිරීම සහ ඇති/නැත පරීක්ෂා කිරීම දත්ත වලංගු කිරීමේ වලංගු ක්‍රම වේ.

3. “A” is represented by 1000001 in ASCII. What is the correct ASCII value for “COD” ?
ASCII කේත ක්‍රමයේදී “A” නියෝජනය කරනු ලබන්නේ 1000001 මගිනි. “COD” සඳහා නිවැරදි කේතය වනුයේ කුමක්ද?

1) 01000011 01001111 01000100
2) 01000010 01001110 01000100
3) 01001111 01001111 01000101
4) 01000011 01001111 01000110
5) 01000001 01001101 01000100

**Answer: 1)**

ASCII (American Standard Code for Information Interchange) assigns a binary value to each character. We need to find the correct binary (ASCII) values for the string **"COD"**.
ASCII (තොරතුරු හුවමාරුව සඳහා ඇමරිකානු සම්මත කේතය) එක් එක් අක්ෂර සඳහා ද්විමය අගයක් පවරයි. **"COD"**තන්තුව සඳහා නිවැරදි ද්විමය (ASCII) අගයන් සොයා ගත යුතුය.

- **C**
  The ASCII value for "C" is **67**. In binary, 67 is represented as **01000011**.
  "C" සඳහා ASCII අගය **67** වේ. ද්විමය වලදී, 67 **01000011** ලෙස නිරූපණය කෙරේ.

- **O**
  The ASCII value for "O" is **79**. In binary, 79 is represented as **01001111**.
  **"O"** සඳහා ASCII අගය **79** වේ. ද්විමය වලදී, 79 **01001111** ලෙස නිරූපණය කෙරේ.

- **D**
  The ASCII value for **"D"** is **68**. In binary, 68 is represented as **01000100**.
  **"D"** සඳහා **ASCII** අගය **68** වේ. ද්විමය වලදී, 68 **01000100** ලෙස නිරූපණය කෙරේ.

Therefore, the correct ASCII representation for **"COD"** is 01000011 01001111 01000100.
එබැවින්, **"COD"** සඳහා නිවැරදි ASCII නියෝජනය 01000011 01001111 01000100 වේ.

So, the correct answer is 1).
එබැවින්, නිවැරදි පිළිතුර 1).

4. $(\bar{A} + B).(\bar{B} + C).(A + \bar{C})$

Convert the above POS statement into a SOP statement.
ඉහත POS ප්‍රකාශය SOP ප්‍රකාශයක් බවට පරිවර්තනය කරන්න.

1) $(\bar{A}.B) + (\bar{B}.C) + (A.\bar{C})$
2) $(A.B) + (B.C) + (A.C)$
3) $(A + B)(B + C)(A + C)$
4) $(\overline{A + B})(\overline{B + C})(\overline{A + C})$
5) $\underline{(A.\bar{B}) + (B.\bar{C}) + (\bar{A}.C)}$

**Answer: 5)**

Let's construct the answer step by step.
පියවරෙන් පියවර පිළිතුර ගොඩනගමු.

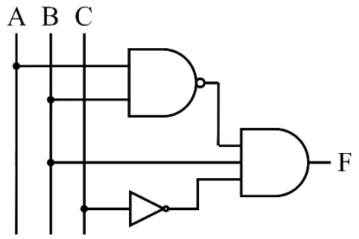
Make the whole function as NOT
මුළු ශ්‍රිතයේම NOT අර්ථය ලබා ගනී.

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

Let's use the rules of Boolean algebra and trivialize it.
බූලිය වීජ ගණිතමය නීති භාවිත කර සුළු කරමු.

| | |
|---|---|
| $\overline{(\bar{A} + B).(\bar{B} + C).(A + \bar{C})}$ | |
| $\overline{(\bar{A} + B)} + \overline{(\bar{B} + C)} + \overline{(A + \bar{C})}$ | De Morgan’s law / ද මුවාර් ප්‍රමේයය |
| $(\bar{\bar{A}}.\bar{B}) + (\bar{\bar{B}}.\bar{C}) + (\bar{A}.\bar{\bar{C}})$ | De Morgan’s law / ද මුවාර් ප්‍රමේයය |
| $A.\bar{B} + B.\bar{C} + \bar{A}.C$ | Double inverse law / ද්විත්ව ප්‍රතිලෝම න්‍යාය |

5. What is the Karnaugh map of the output F of the following logic circuit?
පහත දැක්වෙන තාර්කික පරිපථයේ F ප්‍රතිදානයට අදාළ කානෝ සිතියම කුමක්ද?




1)

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |

2)

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |

3)

| C \ AB | 10 | 00 | 01 | 11 |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |

4)

| C \ AB | 10 | 00 | 01 | 11 |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |

5)

| C \ AB | 11 | 10 | 00 | 01 |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |

**Answer: 3)**

First let's get the truth table for this logic circuit.
පළමුව මෙම තාර්කික පරිපථයට අදාළ සත්‍යතා වගුව ලබාගනිමු.

| A | B | C | $\bar{C}$ | A.B | $\overline{A.B}$ | F |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |

Accordingly, the following Karnaugh map can be obtained.
ඒ අනුව පහත කානෝ සිතියම ලබාගත හැකිය.

| C \ AB | 10 | 00 | 01 | 11 |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |

So, the correct answer number is 3.
එමනිසා, නිවැරදි පිළිතුරු අංකය 3 වේ.

6. What does it mean when an attribute is described as "multivalued"?
උපලක්ෂණයක් "බහු අගයක්" ලෙස විස්තර කරන විට එයින් අදහස් කරන්නේ කුමක්ද?

1) It can take on only one value at any time.
එය ඕනෑම අවස්ථාවක එක් අගයක් පමණක් ගත හැක.
2) It can take on multiple values for each instance of an entity.
එය භූතාර්තයක එක් එක් අවස්ථාව සඳහා බහු අගයන් ගත හැක.
3) It is dependent on another attribute for its value.
එය එහි අගය සඳහා වෙනත් ගුණාංගයක් මත රඳා පවතී.
4) It is always null or empty.
එය සැමවිටම ශුන්‍ය හෝ හිස් ය.
5) It must be unique across all instances.
එය සෑම අවස්ථාවකදීම අනන්‍ය විය යුතුය.

**Answer: 2)**

A brief explanation of each option / එක් එක් විකල්පය පිළිබඳ කෙටි පැහැදිලි කිරීමක්:

**Option / විකල්ප 1)**
This describes a single-valued attribute, not a multivalued attribute. A single-valued attribute can have only one value per entity instance
මෙය බහු අගයක් සහිත උපලක්ෂණයක් නොව තනි අගයක් සහිත උපලක්ෂණයක් විස්තර කරයි. තනි අගයක් සහිත උපලක්ෂණයකට තිබිය හැක්කේ එක් එක් ස්ථිතික අවස්ථාවකට එක් අගයක් පමණි

**Option / විකල්ප 2)**
This is the correct definition of a multivalued attribute. It means that the attribute can store more than one value for each instance of an entity
බහු වටිනාකම් සහිත උපලක්ෂණයක නිවැරදි අර්ථ දැක්වීම මෙයයි. එහි අර්ථය වන්නේ එක් එක් භූතාර්ථයක එක් එක් අවස්ථාව සඳහා උපලක්ෂණ එකකට වඩා වැඩි අගයක් ගබඩා කළ හැකි බවයි

**Option / විකල්ප 3)**
This refers to a derived attribute or an attribute with dependency, where the value is determined based on the value of another attribute
මෙය ව්‍යුත්පන්න උපලක්ෂණයක් හෝ පරායත්තතාවයක් සහිත උපලක්ෂණයක් වෙත යොමු කරයි, එහිදී අගය වෙනත් උපලක්ෂණයක අගය මත පදනම්ව තීරණය වේ.

**Option / විකල්ප 4)**
This describes an attribute that has no value or is optional, but it does not necessarily imply that the attribute is multivalued. An attribute being null or empty simply means no value is assigned to it.
මෙමගින් කිසිදු අගයක් නොමැති හෝ වෛකල්පිත උපලක්ෂණයක් විස්තර කරයි, නමුත් එය අවශ්‍යයෙන්ම උපලක්ෂණය බහු අගයක් ඇති බව ඇඟවෙන්නේ නැත. උපලක්ෂණයක් ශුන්‍ය හෝ හිස් වීම යන්නෙන් අදහස් වන්නේ එයට කිසිදු අගයක් පවරන්නේ නැති බවයි.

**Option / විකල්ප 5)**
This describes a unique attribute, such as a primary key. It requires that each value of this attribute be distinct across all instances of the entity.
මෙය ප්‍රාථමික යතුරක් වැනි අනන්‍ය උපලක්ෂණයක් විස්තර කරයි. මෙම උපලක්ෂණයේ සෑම අගයක්ම භූතාර්ථයේ සියලුම අවස්ථා හරහා වෙනස් වීම අවශ්‍ය වේ.

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

7. What is the output of the below Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
x = [ 1 , [2, 3, 4] , 5, ( 6 , [7 , 8] ) ]
print( x [ 3 ] [ -1 ]  )
```

1) `6, [7, 8]`   2) `[7, 8]`   3) `1`   4) `[2, 3, 4]`   5) `6`

**Answer: 2)**

Let's break down above python code to understand how the indexing works.
indexing තේරුම් ගැනීමට පයිතන් කේතයට කාණ්ඩ වශයෙන් සලකා බලමු.

- `x[0] = 1`
- `x[1] = [2, 3, 4]`
- `x[2] = 5`
- `x[3] = (6, [7, 8])`

`x[3]` accesses the fourth element of the list x (which is the tuple `(6, [7, 8])`).
`x[3]` වන ලැයිස්තුවේ හතරවන අංගයට (tuple වන `(6, [7, 8])`) ප්‍රවේශ වේ.

`x[3][-1]` accesses the last element of the tuple `(6, [7, 8])`, which is the list `[7, 8]`.
`(6, [7, 8])` වන tuple හි අවසාන මූලාංගය වෙත `[7, 8]` ප්‍රවේශ වේ.

The code will print `[7, 8]`.
කේතය `[7, 8]` ලෙස මුද්‍රණය කරයි.

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

8. Which of the following code line will create a list in Python?
පයිතන් හි ලැයිස්තුවක් සාදනු ලබන්නේ පහත සඳහන් කුමන කේත පේළියද?

1) `list1 = [1, 2, 3]`   2) `list1 = (1, 2, 3)`   3) `list1 = {1, 2, 3}`
4) `list1 = <1, 2, 3>`   5) `list1 = 1, 2, 3`

**Answer: 1)**

Let's analyze the options / විකල්ප විශ්ලේෂණය කරමු.

1) **`list1 = [1, 2, 3]`:**
This is correct. In Python, lists are created using square brackets '`[]`'. This statement creates a list containing the elements 1, 2 and 3.
මෙය නිවැරදියි. Python හි, ලැයිස්තු සාදනු ලබන්නේ වර්ග වරහන් භාවිතා කර '`[]`'.මෙම ප්‍රකාශය 1, 2 සහ 3 මූලද්‍රව්‍ය අඩංගු ලැයිස්තුවක් නිර්මාණය කරයි.

Ex:
```
list1 = [1, 2, 3]
print(type(list1))  # Output: <class 'list'>
```

2) **`list1 = (1, 2, 3)`:**
This is incorrect. This statement creates a tuple, not a list. Tuples are created using parentheses '`()`'.
මෙය වැරදියි. මෙම ප්‍රකාශය නිර්මාණය කරන්නේ tuple එකක් මිස ලැයිස්තුවක් නොවේ. Tuples නිර්මාණය කර ඇත්තේ වරහන් '`()`' භාවිතා කරමිනි.

Ex:
```
list1 = (1, 2, 3)
print(type(list1))  # Output: <class 'tuple'>
```

3) **list1 = {1, 2, 3}:**
This is incorrect. This statement creates a set, not a list. Sets are created using curly braces '{}'.
මෙය වැරදියි. මෙම ප්‍රකාශය list එකක් නොව set එකක් නිර්මාණය කරයි. '{}' කැරලි වරහන් භාවිතයෙන් කට්ටල සාදනු ලැබේ.

Ex:
```
list1 = {1, 2, 3}
print(type(list1))  # Output: <class 'set'>
```

4) **list1 = <1, 2, 3>:**
This is incorrect. The syntax '<1, 2, 3>' is not valid in Python for creating any standard data structure.
මෙය වැරදියි. ඕනෑම සම්මත දත්ත ව්‍යුහයක් නිර්මාණය කිරීම සඳහා '<1, 2, 3>' වාක්‍ය ඛණ්ඩය Python තුළ වලංගු නොවේ.

5) **list1 = 1, 2, 3:**
This is incorrect. This statement creates a tuple, not a list. When multiple items are separated by commas without any enclosing brackets, Python interprets it as a tuple.
මෙය වැරදියි. මෙම ප්‍රකාශය නිර්මාණය කරන්නේ tuple එකක් මිස list එකක් නොවේ. බොහෝ අයිතම කොමාවකින් කිසිදු සංවෘත වරහන් නොමැතිව වෙන් කළ විට, Python එය tuple ලෙස අර්ථ දක්වයි.

The correct answer for the statement that will create a list in Python is 1) `list1 = [1, 2, 3]`.
Python හි ලැයිස්තුවක් සාදන ප්‍රකාශය සඳහා නිවැරදි පිළිතුර වන්නේ 1) `list1 = [1, 2, 3]`.

9. How do you insert comments in Python code?
පයිතන් කේතයට විවරණ ඇතුළත් කරන්නේ කෙසේද?

1) <u>`#This is a comment`</u>
2) `//This is a comment`
3) `/*This is a comment*/`
4) All of the Above. ඉහත සියල්ලම.
5) None of the above. ඉහත කිසිවක් නොවේ.

**Answer: 1)**

In Python, the correct way to insert comments is 1) # This is a comment
පයිතන් හි, අදහස් ඇතුළත් කිරීමට නිවැරදි ක්‍රමය වන්නේ 1) # This is a comment

This is the correct syntax for comments in Python. The # symbol indicates that everything following it on that line is a comment, which will be ignored by the Python interpreter. This is used for single-line comments.
පයිතන් හි අදහස් සඳහා නිවැරදි වාක්‍ය ඛණ්ඩය මෙයයි. # සංකේතය පෙන්නුම් කරන්නේ එම රේඛාවේ එය අනුගමනය කරන සෑම දෙයක්ම අදහස් දැක්වීමක් වන අතර එය පයිතන් පරිවර්තකයා විසින් නොසලකා හරිනු ඇත. මෙය තනි පේළියේ අදහස් සඳහා භාවිතා වේ.

In Python, you can only use # for comments. If you want to create multi-line comments, you can use multiple # symbols or string literals (enclosed in triple quotes) that are not assigned to any variable, although the latter is more of a documentation string than a traditional comment.
පයිතන් හි, ඔබට අදහස් සඳහා # පමණක් භාවිතා කළ හැක. ඔබට බහු-රේඛා අදහස් නිර්මාණය කිරීමට අවශ්‍ය නම්, ඔබට සම්ප්‍රදායික අදහස් දැක්වීමකට වඩා ලේඛන තන්තුවක් වුවද, කිසිදු විචල්‍යයකට පවරා නැති බහු # සංකේත හෝ තන්තු වචන (ත්‍රිත්ව උද්ධෘත වලින් කොටා ඇත) භාවිතා කළ හැක.

The other options are incorrect for Python / අනෙකුත් විකල්ප පයිතන් සඳහා වැරදියි.

- **Option / විකල්ප 2)**
This syntax is used for comments in languages like C, C++, and Java, but it is not valid in Python.
මෙම වාක්‍ය ඛණ්ඩය C, C++ සහ Java වැනි භාෂා වල අදහස් දැක්වීම් සඳහා භාවිතා කරයි, නමුත් එය පයිතන් වල වලංගු නොවේ.

- **Option / විකල්ප 3)**
  This syntax is also from C-style languages,used for multi-line comments,but it does'nt work in Python.
  මෙම වාක්‍ය ඛණ්ඩය බහු-රේඛා විවරණ සඳහා භාවිතා කරන C-style භාෂා වලින් ද වේ, නමුත් එය පයිතන් හි ක්‍රියා නොකරයි.

Therefore the correct answer is 1) # This is a comment
එබැවින් නිවැරදි පිළිතුර 1) # This is a comment

10. The output of the following Python calculation is,
පහත පයිතන් ගණනය කිරීමේ ප්‍රතිදානය වනුයේ,

```
(False+True)+(True+True)**True-False
```

1) `True` 2) `False` 3) `1` 4) `2` 5) <u>`3`</u>

**Answer: 5)**

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

```
(False+True)+(True+True)**True-False
```

In Python, `1` is represented by `True` and `0` is represented by `False`. Accordingly, these values can be substituted for this expression.
පයිතන් වල `True` යන්නෙන් `1` ත්, `False` යන්නෙන් `0` ත් නිරූපණය වේ. ඒ අනුව මෙම ප්‍රකාශනයට එම අගයන් ආදේශ කරගත හැකිය.

```
(0+1)+(1+1)**1-0
```

Now let's trivialize this expression considering the order of priority.
දැන් ප්‍රමුඛතා අනුපිළිවෙළ සලකමින් මෙම ප්‍රකාශනය සුළු කරගනිමු.

```
(0+1)+(1+1)**1-0
1+(1+1)**1-0
1+2**1-0
1+2-0
3-0
3
```

So, the correct answer number is 5.
එමනිසා නිවැරදි පිළිතුරු අංකය 5 වේ.