සියලු ම හිමිකම් ඇවිරිණි / *All Rights Reserved* ]

Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu

**Information and Communication Technology**

**තොරතුරු හා සන්නිවේදන තාක්ෂණය**

# 2026 QUIZ NO 337 MARKING

2025/11/12
ictfromabc
Ravindu Bandaranayake

1. Which of the following statement is true about computers developed in the past?
අතීතයේ නිපදවන ලද පරිගණක සම්බන්ධයෙන් පහත සඳහන් ප්‍රකාශ වලින් සත්‍ය වන්නේ කුමක්ද?

   1) Transistors were used to build third generation computers.
   තුන්වන පරම්පරාවේ පරිගණක සෑදීමට ට්‍රාන්සිස්ටර භාවිතා කරන ලදී.

   2) Integrated circuits were used in the second generation of computers.
   දෙවන පරම්පරාවේ පරිගණකවල අනුකලිත පරිපථ භාවිතා කරන ලදී.

   3) Assembly language is used to program in third generation computers.
   තුන්වන පරම්පරාවේ පරිගණකවල ක්‍රමලේඛනය සඳහා ඇසෙම්බ්ලි භාෂාව භාවිතා කරන ලදී.

   4) Software industry was started after the second generation.
   දෙවන පරම්පරාවෙන් පසුව මෘදුකාංග කර්මාන්තය ආරම්භ විය.

   5) Microprocessors are used in third generation of computers.
   තෙවන පරම්පරාවේ පරිගණකවල ක්ෂුද්‍ර සකසන භාවිතා කරන ලදී.

   **Answer : 4)**

   Transistors were used to build second generation computers. Advantages of using transistors over than vacuum tubes are they are smaller in size and consumed less power. Assembly language was used to program in this generation.
   දෙවන පරම්පරාවේ පරිගණක තැනීම සඳහා ට්‍රාන්සිස්ටර භාවිතා කරන ලදී. රික්තක නළ වලට වඩා ට්‍රාන්සිස්ටර භාවිතා කිරීමේ වාසි වන්නේ ඒවා ප්‍රමාණයෙන් කුඩා වන අතර අඩු බලයක් පරිභෝජනය කිරීමයි. මේ පරම්පරාවේ පරිගණක වැඩසටහන්ගත කිරීම සඳහා ඇසෙම්බ්ලි භාෂාව භාවිතා විය.

   Integrated Circuits (IC) were used for build computers in third generation. IC's are built using transistors, capacitors and resistors.
   තෙවන පරම්පරාවේ පරිගණක තැනීම සඳහා අනුකලිත පරිපථ (IC) භාවිතා කරන ලදී. අනුකලිත පරිපථ නිපදවා ඇත්තේ ට්‍රාන්සිස්ටර, ධාරිත්‍රක සහ ප්‍රතිරෝධක භාවිතා කරමිනි.

   Software industry were started in the Third generation. Microprocessors are used to built computers in Fourth generation.
   මෘදුකාංග කර්මාන්තය ආරම්භ වූයේ තුන්වන පරම්පරාවේදීය. සිව්වන පරම්පරාවේ පරිගණක තැනීම සඳහා ක්ෂුද්‍ර සකසන භාවිතා කරන ලදී.

   Therefore, number 4 can be taken as the correct answer here.
   එමනිසා මෙහි නිවැරදි පිළිතුර ලෙස අංක 4 ලබාගත හැකිය.

2. Which of the following is not a step in 4-step fetch execute cycle.
පහත කවරක් පියවර හතරේ සෙවුම් ඉෂ්ට චක්‍රයෙහි පියවරක් නොවන්නේද?

   1) Decode instructions into commands / උපදෙස් විකේතනය කිරීම.
   2) Execute commands / උපදෙස් ක්‍රියාත්මක කිරීම.
   3) Fetch instructions from memory / ප්‍රධාන මතකයෙන් උපදෙසක් ලබා ගැනීම.
   4) Output the result / ප්‍රතිඵලය ප්‍රතිදානය කිරීම.
   5) Store results in memory / ප්‍රධාන මතකයේ ප්‍රතිඵලය තැන්පත් කිරීම.

   **Answer : 4 )**

   Steps of the 4-step fetch execute cycle are:
   සෙවුම් ඉෂ්ට චක්‍රයේ පියවර 4 වන්නේ:

Step 1: Fetch instructions from memory. මතකයෙන් උපදෙස් ලබා ගැනීම.
Step 2: Decode instructions into commands. උපදෙස් විධානවලට විකේතනය කිරීම.
Step 3: Execute commands. විධාන ක්‍රියාත්මක කිරීම.
Step 4: Store results in memory. මතකයේ ප්‍රතිඵල ගබඩා කිරීම.

1. Fetch instructions from memory: In this step, the CPU fetches the next instruction from the memory, typically from the main memory (RAM) or cache memory.
මතකයෙන් උපදෙස් ලබා ගැනීම: මෙම පියවරේදී, මධ්‍ය සැකසුම් ඒකකය විසින් ඊලඟ උපදෙස මතකයෙන්, සාමාන්‍යයෙන් ප්‍රධාන මතකයෙන් (RAM) හෝ වාරක මතකයෙන් ලබා ගනී.

2. Decode instructions into commands: Once the instruction is fetched, the CPU decodes it to determine what operation needs to be performed and what operands are involved.
උපදෙස් විධානවලට විකේතනය කිරීම: උපදෙස් ලබා ගත් පසු, මධ්‍ය සැකසුම් ඒකකය එය විකේතනය කරන්නේ කුමන මෙහෙයුම සිදු කළ යුතුද යන්න සහ කුමන මෙහෙයුම් (operands) මතද සම්බන්ධද යන්න තීරණය කිරීමටයි.

3. Execute commands: After the instruction is decoded, the CPU executes the command or operation specified by the instruction. This could involve arithmetic or logical operations, data movement, or control flow operations.
විධාන ක්‍රියාත්මක කිරීම: උපදෙස් විකේතනය කළ පසු, මධ්‍ය සැකසුම් ඒකකය මඟින් උපදෙස් මඟින් දක්වා ඇති විධානය හෝ ක්‍රියාව ක්‍රියාත්මක කරයි. මෙයට අංක ගණිතමය හෝ තාර්කික මෙහෙයුම්, දත්ත චලනය හෝ පාලන ප්‍රවාහ මෙහෙයුම් ඇතුළත් විය හැකිය.

4. Store results in memory: Depending on the instruction executed, there may be a need to store the results back in memory. For example, if the instruction involves performing a calculation, the result may need to be stored in memory for later use or for communication with other parts of the system.
මතකයේ ප්‍රතිඵල ගබඩා කිරීම: ක්‍රියාත්මක කරන ලද උපදෙස් මත පදනම්ව, ප්‍රතිඵල නැවත මතකයේ ගබඩා කිරීමට අවශ්‍ය විය හැක. උදාහරණයක් ලෙස, උපදෙස ගණනය කිරීමක් සිදු කරයි නම්, ප්‍රතිඵලය පසුකාලීන භාවිතය සඳහා හෝ පද්ධතියේ අනෙකුත් කොටස් සමඟ සන්නිවේදනය සඳහා මතකයේ ගබඩා කිරීමට අවශ්‍ය විය හැකිය.

3. The Decimal value of the ASCII code of ‘C’ is 67. Accordingly, which ASCII values represent the characters ‘B’ and ‘E’.
‘C’ හි ASCII කේතයේ දශම අගය 67 කි. ඒ අනුව, කුමන ASCII අගයන් ‘B’ සහ ‘E’ අක්ෂර නියෝජනය කරයි.

1) $1001111_2$, $1010001_2$
2) $1000011_2$, $1000011_2$
3) $\underline{1000010_2, 1000101_2}$
4) $1110011_2$, $1100001_2$
5) $1000011_2$, $1000001_2$

**Answer : 3)**

In the ASCII encoding scheme, each character is assigned a unique numerical value. The ASCII value for 'C' is 67 in decimal. Since 'B' comes before 'C' in the alphabet, its ASCII value is one less, which is 66. Similarly, 'E' comes after 'C' in the alphabet, so its ASCII value is two more, which is 69.
ASCII කේතිකරණ ක්‍රමයේදී, සෑම අක්ෂරයකටම අනන්‍ය සංඛ්‍යාත්මක අගයක් පවරනු ලැබේ. C සඳහා ASCII අගය දශමයෙන් 67 කි. english හෝඩියේ C ට පෙර B එන බැවින්, එහි ASCII අගය එකකින් අඩු වේ, එනම් 66 වේ. එලෙසම, හෝඩියේ Cට පසුව E පැමිණේ, එබැවින් එහි ASCII අගය තවත් දෙකක්, එනම් 69 වේ.

Converting these decimal ASCII values into binary representations:
මෙම දශම ASCII අගයන් ද්විමය නිරූපණයන් බවට පරිවර්තනය කිරීම:

- The ASCII value of 'B' (66) in binary is 1000010.
ද්විමය තුළ (B) 66 හි ASCII අගය 1000010 වේ.

- The ASCII value of 'E' (69) in binary is 1000101.
ද්විමය තුළ (E) 69 හි ASCII අගය 1000101 වේ.

4. What is the appropriate boolean expression for the output F of the given truth table with inputs A and B?
ඳ ඇති සත්‍යතා වගුවෙහි A සහ B ආදාන අනුව ප්‍රතිදානය F සඳහා සුදුසු බූලිය ප්‍රකාශනය කුමක්ද?

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

1) A AND B
2) A NAND B
3) A XOR B
4) A XNOR B
5) A OR B

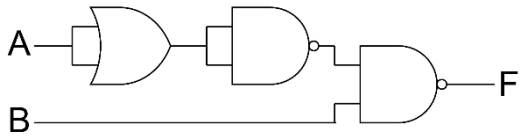
**Answer:3)**

The XOR operation outputs false (0) if both inputs are the same (both 0 or both 1) and true (1) if the inputs are different.
XOR ද්වාරය ප්‍රතිදානය අසත්‍ය ලෙස ඉඳිරිපත් කරන්නේ ආදාන දෙකම 1 හෝ 0 වන විටයි. එමෙන්ම ආදාන දෙකම සමාන නොවී වෙනස් වන්නේ නම් ප්‍රතිදානය සත්‍ය වේ.

- When A and B are both 0, F should be 0 (because they are the same).
A සහ B දෙකම 0 වන විට F ප්‍රතිදානය 0 විය යුතුය (ඒවා සමාන බැවින්).
- When A is 0 and B is 1, F should be 1 (because they are different).
A=0 සහ B=1 වන විට F ප්‍රතිදානය 1 විය යුතුය (ඒවා වෙනස් බැවින්).
- When A is 1 and B is 0, F should be 1 (because they are different).
A=1 සහ B=0 වන විට F ප්‍රතිදානය 1 විය යුතුය (ඒවා වෙනස් බැවින්).
- When A and B are both 1, F should be 0 (because they are the same).
A සහ B දෙකම 1 වන විට F ප්‍රතිදානය 0 විය යුතුය (ඒවා සමාන බැවින්).

As you can see, the output matches the behavior of the XOR gate. When A and B are the same, the output is 0, and when they are different, the output is 1. Therefore, the appropriate boolean expression for the output F of the given truth table with inputs A and B is indeed:
F = A XOR B
ඔබට පෙනෙන පරිදි, මෙම ප්‍රතිදානය XOR ද්වාරයේ හැසිරීමට ගැලපේ. A සහ B සමාන වන විට, ප්‍රතිදානය 0 වන අතර ඒවා වෙනස් වූ විට ප්‍රතිදානය 1 වේ. එබැවින් සත්‍යතා වගුවෙහි ප්‍රතිදානය සඳහා වඩාත් සුදුසු පිළිතුර වනුයේ:
F = A XOR B

5. Select the answer which contains the simplified expression of the below circuit.
පහත තාර්කික පරිපථයේ සරල ප්‍රකාශනය අඩංගු පිළිතුර තෝරන්න.




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

**Answer:2)**

When the inputs of the OR gate are connected together, the same input value is obtained as its output.
OR ද්වාරයේ ආදාන එකට සම්බන්ධ කරඇති විට එහි ප්‍රතිදානය ලෙස එම අදානය කල අගයම ලැබේ.

$A + A = A$ (idempotent law)

When the inputs of an NAND gate are connected together, the inverse of that input is obtained as its output.
NAND ද්වාරයේ ආදාන එකට සම්බන්ධ කරඇති විට එහි ප්‍රතිදානය ලෙස එම අදානය කල අගයේ ප්‍රතිලෝමය ලැබේ.

$\overline{A.A} = \overline{A}$ (idempotent law)

Then F is obtained by inputting that output and B, to another NAND gate.
පසුව එම ප්‍රතිදානය හා B, තවත් NAND ද්වාරයකට ආදානය වීමෙන් F ලැබේ.

Therefore, the simplified expression for the given circuit is
එබැවින්, දී ඇති පරිපථය සඳහා සරල කළ ප්‍රකාශනය

$F = \overline{\overline{A} \cdot B}$
$= \overline{\overline{A}} + \overline{B}$ (De morgan's)
$= A + \overline{B}$ (Double inverse)

So the correct Asnwer Sould be 2) $A + \overline{B}$
එබැවින් නිවැරදි පිළිතුර වනුයේ 2) $A + \overline{B}$

6. Consider the following Extended Entity-Relationship (EER) diagram components.
පහත දැක්වෙන විස්තිරණ භූතාර්ථ-සම්බන්ධතා (EER) සටහන් සංරචක සලකා බලන්න.

A - Specialization involves partitioning an entity set into subsets.
විශේෂීකරණය යනු යම් වස්තුවක් උප කුලකවලට බෙදීමයි.

B - Generalization represents the abstraction of several lower-level entities into a higher-level entity.
සාමාන්‍යකරණය යනු පහළ මට්ටමේ භූතාර්ථ කිහිපයක් ඉහළ මට්ටමේ වස්තුවක් බවට සාරාංශ කිරීමයි.

C - A disjoint constraint specifies that an entity can belong to one and only one subclass.
අසංවිධිත සීමාවක් මඟින් භූතාර්ථයකට අයත් විය හැක්කේ එක් උප පංතියකට පමණක් බව සඳහන් කරයි.

Which of the following statement(s) is/are correct?
පහත සඳහන් කුමන ප්‍රකාශ(ය) නිවැරදිද?

1) Statement A is false, and Statement B is false / A ප්‍රකාශය අසත්‍ය වන අතර B ප්‍රකාශය අසත්‍ය වේ.
2) Statement C is false, and Statement B is false / C ප්‍රකාශය අසත්‍ය වන අතර B ප්‍රකාශය අසත්‍ය වේ.
3) Statement A is true, and Statement B is false / A ප්‍රකාශය සත්‍ය වන අතර B ප්‍රකාශය අසත්‍ය වේ.
4) Statement B is false, and Statement C is true / B ප්‍රකාශය අසත්‍ය වන අතර C ප්‍රකාශය සත්‍ය වේ.
5) <u>All statements are true / සියලුම ප්‍රකාශ සත්‍ය වේ.</u>

**Answer: 5)**

Explanation of each statement / එක් එක් ප්‍රකාශය පැහැදිලි කිරීම:

**Statement / ප්‍රකාශය A**
This statement is true. Specialization in EER diagrams involves dividing an entity set into smaller, more specific subsets based on certain characteristics or attributes.
මෙම ප්‍රකාශය නිවැරදියි. EER රූප සටහන් වල විශේෂීකරණය යනු යම් යම් ලක්ෂණ හෝ උපලක්ෂණ මත පදනම්ව කුඩා, වඩාත් නිශ්චිත උප කුලකවලට භූතාර්ථයක් බෙදීමයි.

**Statement / ප්‍රකාශය B**
This statement is also true. Generalization is the opposite of specialization. It involves identifying commonalities among several lower-level entities and abstracting them into a higher-level, more general entity
මෙම ප්‍රකාශයත් නිවැරදියි. සාමාන්‍යකරණය යනු විශේෂීකරණයේ ප්‍රතිවිරුද්ධයයි. එයට පහළ මට්ටමේ භූතාර්ථ කිහිපයක් අතර පොදු ලක්ෂණ හඳුනා ගැනීම සහ ඒවා ඉහළ මට්ටමේ, වඩා සාමාන්‍ය වස්තුවකට වියුක්ත කිරීම ඇතුළත් වේ.

**Statement / ප්‍රකාශය C**
This statement is true. A disjoint constraint in specialization means that an entity can belong to only one of the subclasses. This is often represented in EER diagrams using the keyword `DISJOINT`.
මෙම ප්‍රකාශය නිවැරදියි. විශේෂීකරණයේ අසංවිධිත බාධාවක් යන්නෙන් අදහස් වන්නේ භූතාර්ථයකට අයත් විය හැක්කේ උප පංතිවලින් එකකට පමණක් බවයි. මෙය බොහෝ විට EER රූප සටහන් වල `DISJOINT` යන මූල පදය භාවිතා කරයි.

Therefore, the correct answer is 5).
එබැවින්, නිවැරදි පිළිතුර 5) වේ.

7. What is the purpose of an "index" in a database?
දත්ත සමුදායක "දර්ශකයක්" භාවිතා කරන්නේ කුමක් සඳහාද?

1) To speed up data retrieval operations / දත්ත ලබාගැනීමේ මෙහෙයුම් වේගවත් කිරීමට.
2) To reduce data redundancy / දත්ත සමතිරික්තතාව අඩු කිරීමට.
3) To create backup copies of the database / දත්ත සමුදායක එකේ උපස්ථ පිටපත් සැකසීමට.
4) To encrypt data in the database / දත්ත සමුදා දත්ත සංකේතනය කිරීමට.
5) To delete unused data from the database / දත්ත සමුදායක භාවිතය නොකරන දත්ත මකා දැමීමට.

**Answer : 1)**

**Explanation / පැහැදිලි කිරීම.**

**To speed up data retrieval operations. / Data retrieval operations වේගවත් කිරීමට**
True: Indexes are specifically designed to improve the speed of data retrieval by allowing the database to quickly locate data without scanning every row in a table. This significantly enhances query performance and reduces access time for specific records.
සත්‍ය: වගුවක සෑම පේළියක්ම පරිලෝකනය නොකර දත්ත සමුදායට ඉක්මනින් දත්ත සොයා ගැනීමට ඉඩ දීමෙන් දත්ත ලබා ගැනීමේ වේගය වැඩි දියුණු කිරීම සඳහා දර්ශක විශේෂයෙන් නිර්මාණය කර ඇත. මෙය විමසුම් කාර්ය සාධනය සැලකිය යුතු ලෙස වැඩි දියුණු කරන අතර නිශ්චිත වාර්තා සඳහා ප්‍රවේශ කාලය අඩු කරයි.

**To reduce data redundancy. / දත්ත සමතිරික්තතාව අඩු කිරීමට**
False: While indexes can help improve performance, they do not inherently reduce data redundancy. Reducing redundancy is typically achieved through normalization processes in database design.
අසත්‍ය: දර්ශක කාර්ය සාධනය වැඩි දියුණු කිරීමට උපකාරී වුවද, ඒවා සහජයෙන්ම දත්ත අතිරික්තය අඩු නොකරයි. අතිරික්තය අඩු කිරීම සාමාන්‍යයෙන් දත්ත සමුදා නිර්මාණයේ සාමාන්‍යකරණ ක්‍රියාවලින් හරහා සාක්ෂාත් කරගනු ලැබේ.

**To create backup copies of the database. / දත්ත සමුදායක එකේ backup copies create කිරීමට**
False: Indexes are not used for creating backup copies of the database. Backups are separate processes that involve copying entire databases or tables to ensure data safety.
අසත්‍ය: දත්ත සමුදායේ උපස්ථ පිටපත් නිර්මාණය කිරීම සඳහා දර්ශක භාවිතා නොකෙරේ. උපස්ථ යනු දත්ත ආරක්ෂාව සහතික කිරීම සඳහා සම්පූර්ණ දත්ත සමුදායන් හෝ වගු පිටපත් කිරීම ඇතුළත් වෙනම ක්‍රියාවලින් වේ.

**To encrypt data in the database. / දත්ත සමුදා දත්ත සංකේතනය කිරීමට**
False: Indexes do not serve the purpose of encrypting data. Encryption is a security measure applied to protect data confidentiality, while indexes focus on improving access speed.
අසත්‍ය: දර්ශක දත්ත සංකේතනය කිරීමේ අරමුණ ඉටු නොකරයි. සංකේතනය යනු දත්ත රහස්‍යභාවය ආරක්ෂා කිරීම සඳහා යොදන ආරක්ෂක පියවරක් වන අතර, දර්ශක ප්‍රවේශ වේගය වැඩි දියුණු කිරීම කෙරෙහි අවධානය යොමු කරයි.

**To delete unused data from the database. / දත්ත සමුදායක භාවිතය නොකරන දත්ත මකා දැමීමට**
False: Indexes do not facilitate the deletion of unused data. Data deletion is managed through specific SQL commands, and indexes simply assist in locating records more efficiently.
අසත්‍යයි: දර්ශක භාවිතයට නොගත් දත්ත මකා දැමීමට පහසුකම් සපයන්නේ නැත. දත්ත මකා දැමීම නිශ්චිත SQL විධාන හරහා කළමනාකරණය කරනු ලබන අතර, දර්ශක හුදෙක් වාර්තා වඩාත් කාර්යක්ෂමව සොයා ගැනීමට උපකාරී වේ.

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

8. What will be the output when the Python code given below is executed and given 3.25 as its input?
**පහත දක්වා ඇති පයිතන් කේතය ක්‍රියාත්මක කර එහි ආදානය ලෙස 3.25 ලබා දුන් විට ප්‍රතිදානය කුමක් වේද?**

```
number=int(input("Enter a Number"))
print(number*2)
```

1) `6.5` 2) `6` 3) `3.25` 4) `"6.25"` 5) `Error`

**Answer : 5 )**

The code provided will result in an error when executed with the input 3.25.
**ලබා දී ඇති කේතය 3.25 ආදානය සමඟ ක්‍රියාත්මක කරන විට දෝෂයක් ඇති කරයි.**

int(input("Enter a Number"))
This line attempts to convert the input to an integer using the int( ) function.
**මෙම රේඛාව int( ) ශ්‍රිතය භාවිතයෙන් ආදානය පූර්ණ සංඛ්‍යාවකට පරිවර්තනය කිරීමට උත්සාහ කරයි.**

If the user inputs 3.25, which is a floating-point number, the int( ) function will raise a ValueError because it cannot convert the string "3.25" directly to an integer.
**පරිශීලකයා දශමය අංකයක් වන 3.25 ඇතුළත් කළහොත්, int( ) ශ්‍රිතය මඟින් "3.25" තන්තුව සෘජුවම පූර්ණ සංඛ්‍යාවකට පරිවර්තනය කළ නොහැකි නිසා ValueError ඉහළ නංවයි.**

The correct answer is, 5) `Error`
**නිවැරදි පිළිතුර නම්, 5) `Error`**

9. What will be the output of the following code if the input is **`0.5`**?
**ආදානය `0.5` නම් පහත කේතයේ ප්‍රතිදානය කුමක් වේද?**

```
number = int(input("Enter a number: "))
if (number >= -0.5):
    print("Positive number")
else:
    print("Non-positive number")
```

1) `Non-positive number`
2) `Positive number`
3) `Float number`
4) `Error`
5) No output

**Answer : 4)**

**1) `number = int(input("Enter a number: "))`**
In this line, the input function reads user input as a string, and the `int()` function attempts to convert that input to an integer. However, when `0.5` is entered, it is a floating-point number, not an integer. The `int()` function cannot convert a non-integer string like `0.5` into an integer, which raises a `ValueError`.
**මෙම පේළියේහි, ආදාන ශ්‍රිතය පරිශීලක ආදානය** string **එකක් ලෙස කියවන අතර, `int()` ශ්‍රිතය එම ආදානය පූර්ණ සංඛ්‍යාවකට පරිවර්තනය කිරීමට උත්සාහ කරයි. කෙසේ වෙතත්, `0.5` ඇතුළත් කළ විට, එය ඉපිලෙන ලක්ෂ්‍ය සංඛ්‍යාවක් මිස පූර්ණ සංඛ්‍යාවක් නොවේ. `ValueError` මතු කරන `0.5` වැනි නිඛිල නොවන** string **එකක් පූර්ණ සංඛ්‍යාවක් බවට පරිවර්තනය කිරීමට `int()` ශ්‍රිතයට නොහැක.**

**2) `if (number >= -0.5):`**
This line would be executed if the `int()` conversion was successful. However, since the code throws an error in the previous step, this line is never reached.
`int()` **පරිවර්තනය සාර්ථක නම් මෙම රේඛාව ක්‍රියාත්මක වේ. කෙසේ වෙතත්, කේතය පෙර පියවරේදී දෝෂයක් ඇති බැවින්, මෙම රේඛාව කිසි විටෙකත් ළඟා නොවේ.**

**3) `print("Positive number") / print("Non-positive number")`**
These statements print based on the value of `number`, but again, because the code has already encountered an error, these conditions are not evaluated.
**මෙම ප්‍රකාශයන් මුද්‍රණය වන්නේ `number` අගය මත පදනම්වය, නමුත් නැවතත්, කේතය දැනටමත් දෝෂයකට මුහුණ දී ඇති නිසා, මෙම කොන්දේසි ඇගයීමට ලක් නොකෙරේ.**

When the input is `0.5`, the `int()` function tries to convert it into an integer, which is not possible for a float like `0.5`. This leads to a `ValueError`, and the program stops execution. Therefore, the correct answer is `Error`.
**ආදානය `0.5` වූ විට, `int()` ශ්‍රිතය එය පූර්ණ සංඛ්‍යාවක් බවට පරිවර්තනය කිරීමට උත්සාහ කරයි, එය `0.5` වැනි ඉපිලෙන ලක්ෂ්‍ය සඳහා කළ නොහැක. මෙය `ValueError` වෙත යොමු කරයි, සහ වැඩසටහන ක්‍රියාත්මක කිරීම නවත්වයි. එබැවින් නිවැරදි පිළිතුර `Error` වේ.**

10. What is the output of the below python code?
**පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?**

```
pattern = 5
for i in range(pattern, 0, -1):
    for j in range(1, i + 1):
        print(j, end=" ")
    print()
```

1)
```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

2)
```
1
1 0
1 0 1
1 0 1 0
1 0 1 0 1
```

3)
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

4)
```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

5)
```
1
1 3
1 3 5
1 3 5 7
1 3 5 7 9
```

**Answer : 4 )**

Let's consider the above python code. / **ඉහත python කේතය සලකා බලමු.**

- The outer loop starts from 5 and counts down to 1.
  **පිටත ලූපය 5 සිට ආරම්භ වන අතර 1 දක්වා ගණන් කරයි.**
- The inner loop prints numbers from 1 to i, where i is the current value from the outer loop.
  **අභ්‍යන්තර ලූපය 1 සිට i දක්වා සංඛ්‍යා මුද්‍රණය කරයි, එහිදී i යනු පිටත ලූපයේ වත්මන් අගයයි.**
- After printing each line, the `print()` command moves to the next line.
  **සෑම පේළියක්ම මුද්‍රණය කිරීමෙන් පසු, `print()` විධානය ඊළඟ පේළියට ගමන් කරයි.**

1. When `i = 5`, the inner loop prints 1 2 3 4 5.
   **`i = 5` වූ විට, අභ්‍යන්තර ලූපය 1 2 3 4 5 මුද්‍රණය කරයි.**
2. When `i = 4`, the inner loop prints 1 2 3 4.
   **`i = 4` වූ විට, අභ්‍යන්තර ලූපය 1 2 3 4 මුද්‍රණය කරයි.**
3. When `i = 3`, the inner loop prints 1 2 3.
   **`i = 3` වූ විට, අභ්‍යන්තර ලූපය 1 2 3 මුද්‍රණය කරයි.**
4. When `i = 2`, the inner loop prints 1 2.
   **`i = 2` වූ විට, අභ්‍යන්තර ලූපය 1 2 මුද්‍රණය කරයි.**
5. When `i = 1`, the inner loop prints 1.
   **`i = 1` වූ විට, අභ්‍යන්තර ලූපය 1 මුද්‍රණය කරයි.**

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