සියලු ම හිමිකම් ඇවිරිණි / All Rights Reserved]

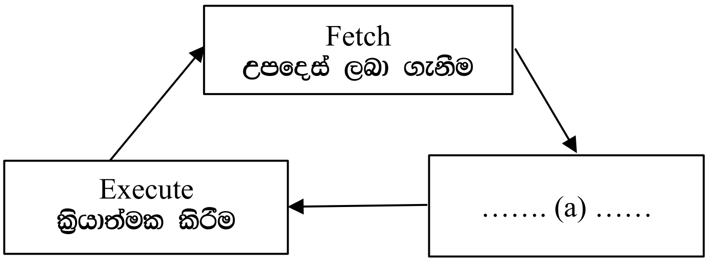
Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 347 MARKING

2025/11/22
ictfromabc
Ravindu Bandaranayake

1. what is the component of fetch execute cycle, that suits the best for the (a) space given below,
සෙවුම් ඉෂ්ට චක්‍රයට අදාලව (a) හිස්තැනට වඩාත් සුදුසු වන්නේ,



1) store ගබඩා කිරීම 2) decode විකේතනය 3) implementation ධාවනය කිරීම 4) advertising ප්‍රචාරණය 5) delete මැකීම

Answer :2)

In the fetch-execute cycle, the instruction is first stored in the main memory. The control unit then decodes the instruction, which is followed by the execution of the decoded instruction by the arithmetic and logic unit. Finally, the result obtained is stored back in the main memory.
සෙවිම් ඉෂ්ට චක්‍රයේදී දී සිදුවන්නේ ප්‍රධාන මතකයේ ගබඩා කර ඇති උපදෙස් ලබාගෙන පාලන ඒකකය මගින් අදාළ උපදෙස විකේතනය කරයි. එම විකේතනය කළ උපදෙස අංක ගණිතමය හා තාර්කික ඒකකය මගින් ක්‍රියාත්මක කරයි. ඉන්පසු අවසාන වශයෙන් එහි ප්‍රතිඵලය ප්‍රධාන මතකයේ ගබඩා කරයි.

2. Which of the following is used to run an application that requires more space than a computer's primary memory space?
පරිගණකයක ප්‍රාථමික මතක අවකාශයට වඩා වැඩි අවකාශයක් අවශ්‍ය වන යෙදුමක් ධාවනයට උපයෝගී කරගන්නේ,

1) RAM සසම්භාවී ප්‍රවේශ මතකය
2) Extended Memory කෙටිකාලීන මතකය
3) ROM පඨන මාත්‍ර මතකය
4) Cache Memory නිහිත මතකය
5) Virtual Memory අතථ්‍යරූපී මතකය

Answer:5)

Virtual memory provides the extra space that RAM needs, if it is unable to perform certain operations due to the lack of space.
අතථ්‍ය මතකය මගින් සිදුවන්නේ RAM එක තුල යම්කිසි මෙහෙයුම් ප්‍රමාණයක් ඉටු කිරීමට ඉඩ නොමැති නම් එයට අවශ්‍ය වන අමතර ඉඩ ප්‍රමාණය ලබාදීමයි.

3. The two's complement of a negative number X is 110101102. The value of X is equal to,
X නම් ඍණ සංඛ්‍යාවක දෙකෙහි අනුපූරකය 11010110_2 වේ. එම X හි අගය සමාන වන්නේ,

A. -52_{10} B. $-2A_{16}$ C. -41_8

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

Answer : 2)

It is given that the X is a negative value of a two’s complement number.
X යනු දෙකේ අනුපූරක අංකයක සෘණ අගයක් බව ලබා දී ඇත.

In order to obtained an answer for X, we have to reverse the process of getting a two’s complement value from a negative number.
X සඳහා පිළිතුරක් ලබා ගැනීම සඳහා, අපට සෘණ අංකයකින් දෙකෙහි අනුපූරක අගය ලබා ගැනීමේ ක්‍රියාවලිය ආපසු හැරවිය යුතුය.

11010110_2							
Substract 1 from the 11010110_2 11010110_2 න් 1 අඩු කරන්න							
11010110 00000001 11010101							
Inverse the bits of 11010110_2 11010110_2 බිටු ප්‍රතිලෝම කරමු							
00101010_2							
Convert 00101010_2 to Decimal 00101010_2 දශමයට පරිවර්තනය කරමු							
0	0	1	0	1	0	1	0
128	64	32	16	8	4	2	1
32 x 1 + 8 x 1 + 2 x 1 = 42							
42_{10}							

Convert 00101010_2 to octal								
0	0	0	1	0	1	0	1	0
4	2	1	4	2	1	4	2	1
0			4 + 1			2		
5					2			
52_8								

Convert 00101010_2 to Hexa-decimal							
0	0	1	0	1	0	1	0
8	4	2	1	8	4	2	1
2				8 + 2 = 10			
2				A			
$2A_{16}$							

Hence X is a negative number, the answer obtained by above calculation also a negative number.
X යනු සෘණ අංකයක් එබැවින් ඉහත ගණනය කිරීමෙන් ලැබෙන පිළිතුර ද සෘණ අංකයකි.

Out of the above A, B and C value, only equivalent value to decimal 42_{10} is $2A_{16}$.
ඉහත A, B සහ C අගයන් අතරින්, දශම 42_{10} ට සමාන අගය පමණක් $2A_{16}$ වේ.

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

4. Consider the below statements and select the False statement/s.
පහත ප්‍රකාශ සලකා බලා අසත්‍ය ප්‍රකාශ/ය තෝරන්න.

A. You can make an AND gate by only using OR gates
ඔබට OR ද්වාර පමණක් භාවිතා කර AND ද්වාරයක් සාදාගත හැකිය.

B. You can make an OR gate by only using AND gates
ඔබට AND ද්වාර පමණක් භාවිතා කර OR ද්වාරයක් සාදාගත හැකිය.

C. XOR gate can be implemented by only using AND gates and OR gates
AND ද්වාර සහ OR ද්වාර පමණක් භාවිතා කර XOR ද්වාරයක් සාදාගත හැකිය.

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

Answer : 4)

Statement A:
This statement is false. The fundamental operations performed by AND and OR gates are inherently different. An AND gate produces a true output only if all its inputs are true, while an OR gate produces a true output if any of its inputs are true. While you can use a combination of AND, OR and NOT gates to implement various logical functions, it is not possible to create an AND gate using only OR gates. This is because the intrinsic behavior of an OR gate cannot be manipulated to replicate the behavior of an AND gate without additional logic operations like negation.
මෙම ප්‍රකාශය අසත්‍යකි. AND සහ OR ද්වාර මගින් සිදු කරන මූලික මෙහෙයුම් සහජයෙන්ම වෙනස් වේ. AND ද්වාරයක් සත්‍ය ප්‍රතිදානයක් නිපදවන්නේ එහි සියලුම යෙදවුම් සත්‍ය නම් පමණක් වන අතර ධෘෂ ද්වාරයක් එහි කිසියම් ආදානයක් සත්‍ය නම් සත්‍ය ප්‍රතිදානයක් නිපදවයි. ඔබට විවිධ තාර්කික ශ්‍රිත ක්‍රියාත්මක කිරීම සඳහා AND, OR සහ NOT ද්වාරවල එකතුවක් භාවිතා කළ හැකි අතර, OR ද්වාර පමණක් භාවිතයෙන් AND ගේට්ටුවක් සෑදිය නොහැක. මක්නිසාද යත්, නිෂේධනය වැනි අමතර තාර්කික මෙහෙයුම් නොමැතිව AND ද්වාරයක හැසිරීම් අනුකරණය කිරීම සඳහා OR ද්වාරයක ආවේණික හැසිරීම හැසිරවිය නොහැකි බැවිනි.

Statement B:
This statement is also false. Similar to the issue with Statement A, the intrinsic functions of AND and OR gates are different. An OR gate outputs true if at least one input is true, whereas an AND gate outputs true only if all inputs are true. While De Morgan's laws show that logical functions can be transformed using combinations of AND, OR and NOT gates, an OR gate cannot be constructed using only AND gates. The fundamental nature of these gates cannot be altered to replicate each other without incorporating negation.
මෙම ප්‍රකාශයද අසත්‍යයකි. A ප්‍රකාශය සමඟ ඇති ගැටලුවට සමානව, AND සහ OR ද්වාරවල සහජ ශ්‍රිත වෙනස් වේ. ධෘෂ ද්වාරයක් අවම වශයෙන් එක් ආදානයක් සත්‍ය නම් සත්‍ය ප්‍රතිදානය කරයි, නමුත් AND ද්වාරයක් ප්‍රතිදානය සත්‍ය වන්නේ සියලු ආදාන සත්‍ය නම් පමණි. ඩි මෝගන්ගේ නීති මගින් AND, OR සහ NOT ද්වාරවල සංයෝජන භාවිතයෙන් තාර්කික ශ්‍රිත පරිවර්තනය කළ හැකි බව පෙන්නුම් කරන අතර, OR ද්වාර AND ගේට්ටු පමණක් භාවිතා කර තැනිය නොහැක. මෙම ද්වාරවල මූලික ස්වභාවය නිෂේධනය ඇතුළත් නොකර එකිනෙක අනුකරණය කිරීමට වෙනස් කළ නොහැක.

Statement C:
This statement is true. A XOR (exclusive OR) gate can be implemented using a combination of AND, OR and NOT gates. The XOR gate outputs true only when the inputs are different. This can be expressed by the Boolean equation:
$\mathrm{XOR}(A, B) = (A\ \mathrm{AND}\ \overline{B})\mathrm{OR}\ (\ \overline{A}\ \mathrm{AND}\ B)$This equation shows that an XOR gate can be built using AND, OR and NOT gates, but not exclusively one type of gate.
මෙම ප්‍රකාශය සත්‍යයකි. XOR (exclusive OR ද්වාරයක් AND, OR සහ NOT ද්වාරවල එකතුවක් භාවිතයෙන් ක්‍රියාත්මක කළ හැක. XOR ද්වාරය සත්‍ය ප්‍රතිදානය කරන්නේ යෙදවුම් වෙනස් වූ විට පමණි. මෙය බූලියන් සමීකරණයෙන් ප්‍රකාශ කළ හැක:
$\mathrm{XOR}(A, B) = (A\ \mathrm{AND}\ \overline{B})\mathrm{OR}\ (\ \overline{A}\ \mathrm{AND}\ B)$ මෙම සමීකරණයෙන් පෙන්නුම් කරන්නේ XOR ද්වාරයක් AND, OR සහ NOT ද්වාරය භාවිතයෙන් ගොඩනැගිය හැකි නමුත්, එක් වර්ගයක ද්වාරය පමණක් නොවන බවයි.

Based on the above analysis, Statements A and B are false because AND and OR gates cannot be created using only the other type of gate. Statement C is true because an XOR gate can indeed be implemented using AND, OR and NOT gates. Therefore, the false statements are 4) A and B only.
ඉහත විශ්ලේෂණය මත පදනම්ව, A සහ B ප්‍රකාශ අසත්‍ය වේ, මන්ද AND සහ OR ද්වාර වෙනත් ආකාරයේ ද්වාරයක භාවිතයෙන් පමණක් සෑදිය නොහැක. ප්‍රකාශය C සත්‍ය වන්නේ XOR ද්වාරයක් ඇත්තෙන්ම AND, OR සහ NOT ද්වාර භාවිතයෙන් ක්‍රියාත්මක කළ හැකි බැවිනි. එබැවින්, අසත්‍ය ප්‍රකාශයන් වන්නේ 4) A and B only.

5. What will be the logic gate corresponding to the following statement?
පහත ප්‍රකාශයට අනුරූප වන තාර්කික ද්වාරය කුමක් වේද?

"Output becomes negative only when all the inputs are positive."
"ප්‍රතිදානය අසත්‍ය වන්නේ සියළු යෙදවුම් සත්‍ය වූ විට පමණි. "

1) NAND gate 2) NOR gate 3) XOR gate 4) OR gate 5) XNOR gate

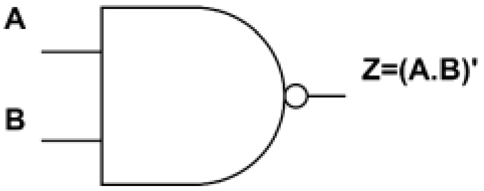
Answer : 1)

Lets develop a truth table according to the above scenario.
ඉහත අවස්ථාව අනුව සත්‍ය වගුවක් සංවර්ධනය කරමු:

ICN	A	B	Z
1	0	0	1
2	0	1	1
3	1	0	1
4	1	1	0

The above truth table with the truth table for the NAND gate. NAND gate outputs 0 (negative) only when all the inputs are 1 (positive). For any other input combination it outputs 1 .
ඉහත සත්‍ය වගුව NAND ද්වාරය සඳහා වන සත්‍ය වගුවට අනුකූල වේ. සියලු ආදාන 1 (ධනාත්මක) වන විට පමණක් හ්හෘ ද්වාරය 0 (ඍණාත්මක) ලෙස ප්‍රතිදානය කරයි. වෙනත් සෑම ආදාන සංයෝජනයකටම එය 1 (ධනාත්මක) ලෙස ප්‍රතිදානය කරයි.

ICN	A	B	(A.B)’
1	0	0	1
2	0	1	1
3	1	0	1
4	1	1	0



Therefore, Z = (A.B)’ answer is 1).
එම නිසා, Z = (A.B)’ පිළිතුර 1) වේ.

6. Which of the following describes a "Many-to-Many" relationship in a relational database?
සම්බන්ධතා දත්ත ගබඩාවක "බහු-බහු" සම්බන්ධතාවයක් විස්තර කරන්නේ පහත සඳහන් කුමක්ද?

1) One record in a table can relate to multiple records in another table.
වගුවක ඇති එක් වාර්තාවක් තවත් වගුවක ඇති වාර්තා කිහිපයකට සම්බන්ධ විය හැක.
2) Multiple records in the first table can relate to multiple records in the second table through a join table.
පළමු වගුවේ ඇති බහු වාර්තා, එකතු කිරීමේ වගුවක් හරහා දෙවන වගුවේ බහු වාර්තා වලට සම්බන්ධ විය හැක.
3) Each record in the first table can be related to many records in the same table.
පළමු වගුවේ ඇති සෑම වාර්තාවක්ම එකම වගුවේ ඇති බොහෝ වාර්තා වලට සම්බන්ධ විය හැක.
4) Each record in the first table is related to one and only one record in the second table.
පළමු වගුවේ ඇති සෑම වාර්තාවක්ම දෙවන වගුවේ එක් වාර්තාවකට පමණක් සම්බන්ධ වේ.
5) A table can only have a one-to-one relationship with another table.
වගුවකට තවත් වගුවක් සමඟ ඒක-ඒක සම්බන්ධයක් පමණක් තිබිය හැක.

Answer: 2)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
Incorrect. This describes a one-to-many relationship, where a single record in the first table can have multiple related records in the second table. For example, a customer placing multiple orders demonstrates a one-to-many relationship.
වැරදියි. පළමු වගුවේ එක් වාර්තාවකට දෙවන වගුවේ සම්බන්ධිත වාර්තා කිහිපයක් තිබිය හැකි ඒක-බහු සම්බන්ධතාවක් මෙය විස්තර කරයි. උදාහරණයක් ලෙස, බහුවිධ ඇණවුම් ලබා දෙන පාරිභෝගිකයෙකු ඒක-බහු සම්බන්ධතාවයක් පෙන්නුම් කරයි.

Option / විකල්ප 2)
Correct. A many-to-many relationship occurs when records in one table can relate to multiple records in another table, and vice versa. This relationship is implemented using a join table that includes foreign keys pointing to both tables. For example, in a "Students" and "Courses" database, a join table like "StudentCourses" can represent students enrolling in multiple courses, and courses having multiple students.
නිවැරදියි. එක් වගුවක ඇති වාර්තා තවත් වගුවක බහුවිධ වාර්තාවලට සම්බන්ධ විය හැකි විට බහු-බහු සම්බන්ධතාවක් ඇති වේ, සහ අනෙක් අතට. මෙම සම්බන්ධතාව ක්‍රියාත්මක කරනු ලබන්නේ වගු දෙකටම යොමු වන ආගන්තුක යතුරු ඇතුළත් එකතු කිරීමේ වගුවක් භාවිතා කරමිනි. උදාහරණයක් ලෙස, "Students" සහ "Courses" දත්ත ගබඩාවක, "StudentCourses" වැනි එක්වීමේ වගුවකට පාඨමාලා කිහිපයකට ඇතුළත් වන සිසුන් සහ සිසුන් කිහිප දෙනෙකු සිටින පාඨමාලා නියෝජනය කළ හැක.

Option / විකල්ප 3)
Incorrect. This describes a self-referential one-to-many relationship, such as an employee hierarchy where one manager supervises multiple employees. It does not represent a many-to-many relationship between two separate tables.
වැරදියි. මෙය එක් කළමනාකරුවෙකු බොහෝ සේවකයින් අධීක්ෂණය කරන සේවක ධුරාවලියක් වැනි ස්වයං-යොමු කරන ඒක-බහු සම්බන්ධතාවයක් විස්තර කරයි. එය වෙනම වගු දෙකක් අතර බහු-බහු සම්බන්ධතාවක් නියෝජනය නොකරයි.

Option / විකල්ප 4)
Incorrect. This describes a one-to-one relationship, where each record in one table corresponds to exactly one record in another table. For example, a table of employees and a table of their corresponding ID cards.
වැරදියි. මෙය ඒක-ඒක සම්බන්ධතාවයක් විස්තර කරයි, එහිදී එක් වගුවක ඇති සෑම වාර්තාවක්ම හරියටම තවත් වගුවක එක් වාර්තාවකට අනුරූප වේ. උදාහරණයක් ලෙස, සේවකයින්ගේ වගුවක් සහ ඔවුන්ගේ අනුරූප හැඳුනුම්පත් වගුවක්.

Option / විකල්ප 5)
Incorrect. This statement is completely false. Tables in a relational database can have various types of relationships, including one-to-one, one-to-many, and many-to-many. This is not limited to a single relationship type.
වැරදියි. මෙම ප්‍රකාශය සම්පූර්ණයෙන්ම අසත්‍යයකි. සම්බන්ධක දත්ත ගබඩාවක ඇති වගු වලට ඒක-ඒක, ඒක-බහු සහ බහු-බහු සම්බන්ධතා ඇතුළුව විවිධ ආකාරයේ සම්බන්ධතා තිබිය හැක. මෙය තනි සම්බන්ධතා වර්ගයකට සීමා නොවේ.

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

7. Which of the following is a part of data manipulation language (DML) of SQL used in relational database?
සම්බන්ධක දත්ත ගබඩාවේ භාවිතා වන SQL හි දත්ත හැසිරවීමේ භාෂාවේ (DML) කොටසක් වන්නේ පහත සඳහන් කුමක්ද?

1) CREATE 2) <u>INSERT</u> 3) ALTER 4) DROP 5) GRANT

Answer: 2)

Explanation / පැහැදිලි කිරීම:

Data Manipulation Language (DML) in SQL is used to manipulate or manage the data within tables in a relational database. Common DML commands include,
SQL හි දත්ත හැසිරවීමේ භාෂාව (DML) සම්බන්ධතා දත්ත ගබඩාවක වගු තුළ දත්ත හැසිරවීමට හෝ කළමනාකරණය කිරීමට භාවිතා කරයි. පොදු DML විධානවලට ඇතුළත් වන්නේ,

`INSERT`: Adds new rows to a table / වගුවකට නව පේළි එක් කරයි.
`UPDATE`: Modifies existing rows in a table / වගුවක පවතින පේළි වෙනස් කරයි.
`DELETE`: Removes rows from a table / වගුවකින් පේළි ඉවත් කරයි.
`SELECT`: Retrieves data from one or more tables / එක වගුවකින් හෝ වැඩි ගණනකින් දත්ත ලබා ගනී.

Analysis of options / විකල්ප විශ්ලේෂණය:

1) **CREATE:**
Part of Data Definition Language (DDL), used to define database structures like tables or databases.
වගු හෝ දත්ත සමුදායන් වැනි දත්ත සමුදා ව්‍යුහයන් නිර්වචනය කිරීමට භාවිතා කරන දත්ත නිර්වචන භාෂාවේ (DDL) කොටසකි.

2) **INSERT:**
Correct. It is a DML command that adds new records to a table.
නිවැරදි. එය වගුවකට නව වාර්තා එකතු කරන DML විධානයකි.

3) **ALTER:**
Part of Data Definition Language (DDL), used to modify database objects like tables or columns.
දත්ත නිර්වචන භාෂාවේ (DDL)කොටසක්, වගු හෝ තීරු වැනි දත්ත සමුදා වස්තු වෙනස් කිරීමට භාවිතා කරයි.

4) **DROP:**
Part of Data Definition Language (DDL), used to delete database objects like tables or databases.
වගු හෝ දත්ත සමුදායන් වැනි දත්ත සමුදා වස්තු මකා දැමීමට භාවිතා කරන දත්ත නිර්වචන භාෂාවේ (DDL) කොටසකි.

5) **GRANT:**
Part of Data Control Language (DCL), used to give permissions to users on database objects.
දත්ත පාලන භාෂාවේ (DCL) කොටසක්, දත්ත සමුදා වස්තු මත පරිශීලකයින්ට අවසර ලබා දීමට භාවිතා කරයි.

Therefore, the correct option is 2).
එබැවින්, නිවැරදි විකල්පය 2) වේ.

8. What will be the output of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් වේවිද?

```
def generate_sequence(n):
    seq = []
    for i in range(n):
        seq.append(i * 2)
    return seq

a = generate_sequence(3)
b = generate_sequence(2)

for i in range(2):
    b.append(i)

print(a)
print(b)
```

1)	2)	3)	4)	5)
`[0, 2, 4]` `[0, 2, 4]`	`[0,0,4]` `[0,2,0]`	`[0, 2, 4]` `[0, 2, 0]`	`[0, 2, 4]` <u>`[0, 2, 0, 1]`</u>	`[0, 2, 1]` `[0, 2, 0, 1]`

Answer : 4)

Explanation / පැහැදිලි කිරීම

The function `generate_sequence(n)` is defined to generate a sequence based on the input n.
`generate_sequence(n)` යන ශ්‍රිතය නිර්වචනය කර ඇත්තේ `n` ආදානය මත පදනම්ව අනුක්‍රමයක් ජනනය කිරීමටයි.

Inside the function, an empty list `seq` is initialized to store the values.
ශ්‍රිතය තුළ, අගයන් ගබඩා කිරීම සඳහා `seq` යනුවෙන් හිස් ලැයිස්තුවක් ආරම්භ කෙරේ.

The for loop iterates over a range from `0` to `n-1`.
`for` ලූපය මගින් `0` සිට `n-1` දක්වා පරාසයක පුනරාවර්තනය වේ.

For each iteration, it calculates `i * 2` and appends this value to the `seq` list.
එක් එක් පුනරාවර්තනය සඳහා, එය `i * 2` ගණනය කර මෙම අගය `seq` ලැයිස්තුවට එකතු කරයි.

After the loop completes, the function returns the populated list `seq`.
ලූපය සම්පූර්ණ වූ පසු, ශ්‍රිතය ජනනය කල `seq` ලැයිස්තු අනුක්‍රමය ලබා දෙයි.

`a = generate_sequence(3)` calls the function with `n=3`.
`a = generate_sequence(3)` මගින් ශ්‍රිතය `n=3` සමඟින් අමතයි.
The resulting list `a` becomes `[0, 2, 4]`, as it appends `0*2`, `1*2`, and `2*2`.
`0*2`, `1*2` සහ `2*2` එකතු වන විට ප්‍රතිඵලය වන `a` ලැයිස්තුව `[0, 2, 4]` බවට පත් වේ.

`b = generate_sequence(2)` calls the function with `n=2`.
`b = generate_sequence(2)` මගින් ශ්‍රිතය `n=2` සමඟින් කැඳවයි.
The resulting list `b` becomes `[0, 2]`, as it appends `0*2` and `1*2`.
`0*2` සහ `1*2` එකතු වන බැවින් ප්‍රතිඵලය වන `b` ලැයිස්තුව `[0, 2]` බවට පත් වේ.

`for i in range(2):` runs for `i = 0` and `i = 1`.
`i = 0` සහ `i = 1` සඳහා ධාවනය වේ.

During each iteration, it appends the current index value (0 and then 1) to the list `b`. After these iterations, `b` becomes `[0, 2, 0, 1]`.
එක් එක් පුනරාවර්තනය අතරතුර, එය වත්මන් දර්ශක අගය (0 සහ පසුව 1) `b` ලැයිස්තුවට එකතු කරයි. මෙම පුනරාවර්තනයෙන් පසුව, b, `[0, 2, 0, 1]` බවට පත් වේ.

Finally, `print(a)` outputs `[0, 2, 4]`.
අවසාන වශයෙන්, `print(a)` මගින් `[0, 2, 4]` ප්‍රතිදානය කරයි.

`print(b)` outputs `[0, 2, 0, 1]`.
`print(b)` මගින් `[0, 2, 0, 1]` ප්‍රතිදානය කරයි.

9. Which of the following sorting algorithms has the worst average-case time complexity?
අවම සාමාන්‍ය අවස්ථා කාල සංකීර්ණතාවක් ඇත්තේ පහත වර්ග කිරීමේ ඇල්ගොරිතම අතරින් කුමකද?

1) <u>Bubble Sort</u> 2) Merge Sort 3) Quick Sort 4) Heap Sort 5) nsertion Sort

Answer : 1)

Let's explain / පැහැදිලි කරමු.

Bubble Sort is considered to have the worst average-case time complexity because its time complexity is **$O(n^2)$**. This results in inefficiency, as adjacent elements are compared and swapped repeatedly. Due to this **quadratic** complexity, it performs much more slowly on larger datasets, making it the least efficient algorithm on average among the options provided.
බුබුළු තේරීම එහි කාල සංකීර්ණතාව **$O(n^2)$** නිසා නරකම සාමාන්‍ය අවස්ථා කාල සංකීර්ණය ලෙස සැලකේ. යාබද මූලද්‍රව්‍ය නැවත නැවත සංසන්දනය කර හුවමාරු වන බැවින් මෙය අකාර්යක්ෂමතාවයට හේතු වේ. මෙම **චතුරස්‍ර** සංකීර්ණත්වය හේතුවෙන්, එය විශාල දත්ත කට්ටල මත බොහෝ සෙමින් ක්‍රියා කරයි, එය ලබා දී ඇති විකල්ප අතර සාමාන්‍යයෙන් අවම කාර්යක්ෂම ඇල්ගොරිතමයක් බවට පත් කරයි.

So, the correct answer number is 1).
එබැවින්, නිවැරදි පිළිතුරු අංකය 1) වේ.

10. What will be the result of the following python code?
පහත පයිතන් කේතයේ ප්‍රතිඵලය කුමක් වේවිද?

```
i = 1
while True:
    if i % 3 == 0:
        break
    print(i)
    i += 1
```

1) [4] [4]	2) [0]	3) 1 2	4) 4 1	5) [1] [1]

Answer : 3)

Explanation / පැහැදිලි කිරීම

The Python code initiates an infinite while True loop, which continues running until it is interrupted by a break statement.
පයිතන් කේතය අසීමිත අතර සත්‍ය ලූපයක් ආරම්භ කරයි, එය break ප්‍රකාශයකින් බාධා වන තෙක් දිගටම ක්‍රියාත්මක වේ.

The variable `i` starts at 1, and in each iteration, the code checks whether `i` is divisible by 3 using the condition if `i % 3 == 0`. If this condition is true, the loop is broken using the break statement. If the condition is false, the current value of `i` is printed, and then `i` is incremented by 1.
`i` විචල්‍යය 1 න් ආරම්භ වන අතර, එක් එක් පුනරාවර්තනයකදී, `i % 3 == 0` නම් කොන්දේසිය භාවිතා කර 3 න් බෙදිය හැකිද යන්න කේතය පරීක්ෂා කරයි. මෙම කොන්දේසිය සත්‍ය නම්, break ප්‍රකාශය භාවිතයෙන් ලූපය කැඩී යයි. කොන්දේසිය අසත්‍ය නම්, `i` හි වත්මන් අගය මුද්‍රණය කරනු ලැබේ, පසුව `i` 1 කින් වැඩි වේ.

In the first iteration, `i` is 1, which is not divisible by 3, so 1 is printed. In the second iteration, `i` becomes 2, which is also not divisible by 3, so 2 is printed. In the third iteration, when `i` reaches 3, the condition `i % 3 == 0` becomes true, and the loop terminates without printing 3.
පළමු පුනරාවර්තනයේ දී, `i` යනු 1 වන අතර එය 3 න් බෙදිය නොහැක, එබැවින් 1 මුද්‍රණය වේ. දෙවන පුනරාවර්තනයේ දී, `i` 2 බවට පත් වේ, එය 3 න් බෙදිය නොහැක, එබැවින් 2 මුද්‍රණය වේ. තුන්වන පුනරාවර්තනයේදී, `i` 3 වෙත ළඟා වූ විට, `i % 3 == 0` කොන්දේසිය සත්‍ය වන අතර, 3 මුද්‍රණය නොකර ලූපය අවසන් වේ.

Therefore, the final output of the code is 1 followed by 2, and the loop ends before the value 3 is printed.
එබැවින්, කේතයේ අවසාන ප්‍රතිදානය 1 න් පසුව 2 වන අතර, 3 අගය මුද්‍රණය කිරීමට පෙර ලූපය අවසන් වේ.

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