කලාප අධ්‍යාපන කාර්යාලය - හිඟුරක්ගොඩ
Zonal Educational Office - Hingurakgoda

අධ්‍යයන පොදු සහතික පත්‍ර (උසස් පෙළ) විභාගය - 2025
ආදර්ශ ප්‍රශ්න පත්‍රය I

තොරතුරු හා සන්නිවේදන තාක්ෂණය I
Information & Communication Technology I

20 S I

පැය තුනයි
Three hours

අමතර කියවීමේ කාලය - මිනිත්තු 10 යි
Additinal Reading Time – 10 minutes

1. පහත ප්‍රකාශ අතුරින් නිවැරදි වන්නේ,
   A - ආයතනයක ඇති විශාල හා සංකීර්ණ දත්ත ප්‍රමාණයක් මහා දත්ත ලෙස හඳුනාගනී.
   B - මහා දත්ත සංකල්පය පරිගණක පාදක දත්ත ගබඩා සමග බිහි වූවකි.
   C - මහා දත්ත සකස් කිරීමට සාම්ප්‍රදායික දත්ත සැකසීමේ මෘදුකාංග භාවිතා කළහැක.
   (1) A පමණි.
   (2) B පමණි.
   (3) A හා B පමණි.
   (4) B හා C පමණි.
   (5) A, B, C සියල්ලම.

හරි 👍 මේ ප්‍රශ්නයේ තියෙන ප්‍රකාශ එකින් එක විමසා බලමු.

ප්‍රකාශ A – *“ආයතනයක ඇති විශාල හා සංකීර්ණ දත්ත ප්‍රමාණයක් මහා දත්ත ලෙස හඳුනාගනී.”*
➡ මෙය **නිවැරදි**. Big Data කියන්නේ විශාල, සංකීර්ණ සහ විවිධ ආකාරයේ දත්ත ප්‍රමාණය.

ප්‍රකාශ B – *“මහා දත්ත සංකල්පය පරිගණක පාදක දත්ත ගබඩා සමග බිහි වූවකි.”*
➡ මෙය **වැරදි**. Big Data සංකල්පය SQL-based database එකක් හෝ සාමාන්‍ය data warehouse එකක් සමග නොව, අලුත් තාක්ෂණ (distributed computing, Hadoop, cloud) වලින් වැඩිවී ගිය එකක්.

ප්‍රකාශ C – *“මහා දත්ත සකස් කිරීමට සාම්ප්‍රදායික දත්ත සැකසීමේ මෘදුකාංග භාවිතා කළහැක.”*
➡ මෙයත් **වැරදි**. Big Data එක සාම්ප්‍රදායික software (MS Excel, relational DBMS වගේ) වලින් හැසිරවිය නොහැක. Hadoop, Spark වගේ විශේෂිත tools භාවිතා කළ යුතුයි.

✓ **නිවැරදි ප්‍රකාශය — A පමණයි**

☞ නිවැරදි පිළිතුර: **(1) A පමණි**

2. දත්ත සම්ප්‍රේෂණය වඩාත් දියුණු තත්ත්වයට පත්ව ඇති වර්තමානයේ ව්‍යාපාර ආයතන මෙන්ම පුද්ගලයින් තමන්ට අවශ්‍ය දත්ත අන්තර්ජාලය හා සම්බන්ධ වී දුරස්ථ ස්ථානයන්හි තබාගෙන භාවිත කිරීමට යොමු වී ඇත. මෙය වළාකුළු පරිගණනයයි. වළාකුළු පරිගණන සංකල්පයේ වාසි ලෙස හඳුනාගත හැක්කේ,
   A - අඩු පිරිවැයක් යටතේ මෘදුකාංග භාවිතා කිරීමටත් දත්ත ගබඩා කර තබා ගැනීමටත් හැකියාව ලැබීම.
   B - සීමාරහිත ආචයන ධාරිතාවය.
   C - අඩුවේග සම්බන්ධතා පවතින අවස්ථාවල ක්‍රියාකාරීත්වය ඉහළ මට්ටමක පැවතීම.
   (1) A පමණි.
   (2) B පමණි.
   (3) A හා B පමණි.
   (4) B හා C පමණි.
   (5) A, B, C සියල්ලම.

හරි 👉 මේක එකින් එක විමසා බලමු.

ප්‍රකාශ A – *"අඩු පිරිවැයක් යටතේ මෘදුකාංග භාවිතා කිරීමටත් දත්ත ගබඩා කර තබා ගැනීමටත් හැකියාව ලැබීම."*
➡ **නිවැරදි** ✓. Cloud computing එකේ ගොඩක් සොෆ්ට්වෙයාර් (SaaS) සහ storage services අඩු subscription ගාස්තු වලින් භාවිතා කරන්න පුළුවන්.

ප්‍රකාශ B – *"සීමාරහිත ආචයන ධාරිතාවය."*
➡ **සංවේදීව නිවැරදි** ✓. Practical එකට cloud storage සීමාවක් තිබුණත්, user perspective එකේදී තවත් තවත් storage add කරගන්න පුළුවන් නිසා "virtually unlimited" කියලා කියනවා.

ප්‍රකාශ C – *"අඩුවේග සම්බන්ධතා පවතින අවස්ථාවල ක්‍රියාකාරීත්වය ඉහළ මට්ටමක පැවතීම."*
➡ වැරදි ❌. Cloud services වලට හොඳ internet connection එකක් අත්‍යාවශ්‍යයි. අඩු වේගයේදී වැඩක් වෙන්නේ නෑ.

✓ නිවැරදි ප්‍රකාශ: **A හා B පමණි**
☞ නිවැරදි පිළිතුර: **(3) A හා B පමණි**

3. පරිගණකය යන වචනය නූතන මිනිසාට ඉතාම සමීප වචනයක් බවටත් පරිගණක යන්ත්‍රය ඔහුගේ ජීවිතයේ සෑම පැතිකඩක් සඳහාම අත්‍යාවශ්‍ය මෙවලමක් බවටත් පත්ව ඇත. පරිගණකයේ මෙම විකාශනය පරම්පරා කිහිපයක් යටතේ සිදුවූවකි. එක් එක් පරම්පරාවට අදාල සුවිශේෂීතා දැක්වෙන නිවැරදි පිළිතුර වන්නේ,
   - A - උසස් මට්ටමේ ක්‍රමලේඛන භාෂා භාවිත කිරීම ආරම්භ වූයේ තුන්වන පරම්පරාවේදී ය.
   - B - ආදාන හා ප්‍රතිදාන සඳහා යතුරුපුවරුව හා මොනිටරය භාවිත කිරීම ආරම්භ වූයේ සිව්වන පරම්පරාවේදී ය.
   - C - පස්වන පරම්පරාවේදී පරිගණක සඳහා කෘතිම බුද්ධිය යොදා ගැනිනි.

   (1) A පමණි.
   (2) B පමණි.
   (3) A හා B පමණි.
   (4) B හා C පමණි.
   (5) A,B,C සියල්ලම

හරි ☺ මේ ප්‍රශ්නය **පරිගණක පරම්පරා** (computer generations) පිළිබඳවයි. එක එක ප්‍රකාශ බලමු:

A - "උසස් මට්ටමේ ක්‍රමලේඛන භාෂා භාවිත කිරීම ආරම්භ වූයේ තුන්වන පරම්පරාවේදී ය."
➡ **නිවැරදි** ✓.
3rd Generation (1960s) – ICs (Integrated Circuits) භාවිතා කළා. High-level languages (C, COBOL, FORTRAN) පුළුල්ව භාවිතා වෙන්න පටන් ගත්තේ මේකේදී.

B - "ආදාන හා ප්‍රතිදාන සඳහා යතුරුපුවරුව හා මොනිටරය භාවිත කිරීම ආරම්භ වූයේ සිව්වන පරම්පරාවේදී ය."
➡ **නිවැරදි** ✓.
4th Generation (1970s–90s) – Microprocessor-based PCs ආව. Keyboard සහ Monitor user I/O devices විදියට සාමාන්‍ය මට්ටමට ගොස් තිබුණේ මේ පරම්පරාවේ.

C - "පස්වන පරම්පරාවේදී පරිගණක සඳහා කෘතිම බුද්ධිය යොදා ගැනිනි."
➡ **නිවැරදි** ✓.

5th Generation (1990s onwards) – AI (Artificial Intelligence), Expert Systems, Natural Language Processing, Neural Networks, Robotics ආදී දේවල් ගොඩනගන්න පටන් ගත්තේ මේ පරම්පරාවේ.

📌 එහෙමයින්, **A, B, C සියල්ලම නිවැරදි.**

☞ නිවැරදි පිළිතුර: **(5) A, B, C සියල්ලම** ✓

4. ක්‍රමලේඛයක ක්‍රියාත්මක වේගය අඩුවීම සඳහා සෘජුවම බලපාන සාධකයක් නොවන්නේ කුමක්ද?
   (1) ක්‍රමලේඛය චුම්භක පටියක අඩංගු වීම
   (2) සකසනයක වේගය අඩුවීම
   (3) වාරක මතකයක් අඩංගු නොවීම
   (4) සකසනයේ බසයේ (Bus width) පළල අඩුවීම
   (5) සසම්භාවී ප්‍රවේශ මතකයේ ධාරිතාවය

☺. මෙතන අරමුණ **"ප්‍රොග්‍රෑම් එකක් ක්‍රියාත්මක වෙන්නේ මන්දගාමී වෙන්න සෘජුවම බලපාන සාධක"** තෝරන එක.

---

(1) ක්‍රමලේඛය චුම්භක පටියක අඩංගු වීම
☞ සෘජුවම බලපානවා ✓. Magnetic tape storage slow → program load & run slow.

(2) සකසනයක වේගය අඩුවීම
☞ CPU speed අඩු නම් → execution speed අඩුවෙනවා ✓.

(3) වාරක මතකයක් අඩංගු නොවීම
☞ Cache memory නැත්නම් → CPUට main memory එකෙන් දත්ත අරගෙන වැඩ කරන්න වෙයි → වැඩ slow ✓.

(4) සකසනයේ බසයේ (Bus width) පළල අඩුවීම
☞ Bus width අඩු නම් → එකවර ගෙන යන දත්ත ප්‍රමාණය අඩුවේ → execution slow ✓.

(5) සසම්භාවී ප්‍රවේශ මතකයේ (RAM) ධාරිතාවය
☞ **Capacity එක වැඩි / අඩු වීම execution "වේගය" සෘජුවම බලපාන්නේ නැහැ** ✗.

- Memory capacity අඩු නම් මතකය ඉක්මනට පිරෙයි, එවිට secondary storage භාවිතා කරන්න වෙයි (එක indirect impact එකක්).
- නමුත් capacity වැඩි/අඩු වෙන එක **CPU instructions execute** කරන වේගයට **direct effect** නැහැ.

✓ නිවැරදි පිළිතුර: **(5) සසම්භාවී ප්‍රවේශ මතකයේ ධාරිතාවය**

5. කාලයෙන් කාලයට විවිධ වර්ගයේ පරිගණක තිර පරිගණක සඳහා භාවිත කරන ලදී. පරිගණක තිරයක් මත පික්සල් නිර්මාණය කිරීම සඳහා විවිධ යුග වල භාවිත කරන ලද තාක්ෂණයන් පමණක් අන්තර්ගත වන පිළිතුර වන්නේ
   (1) ද්‍රව ස්ඵටික (Liquid Crystal), ආලෝක විමෝචන දියෝඩ (LED), ඉලෙක්ට්‍රෝන (Electron)
   (2) ද්‍රව ස්ඵටික (Liquid Crystal), ලේසර් ආලෝක ධාරා, ඉලෙක්ට්‍රෝන (Electron)
   (3) පොස්පර් අංශු, ආලෝක විමෝචන දියෝඩ (LED), චුම්භක අංශු (Magnetic Particles)
   (4) චුම්භක අංශු, ද්‍රව ස්ඵටික (Liquid Crystal), ආලෝක විමෝචන දියෝඩ (LED)

(5) චුම්භක අංශු, ලේසර් ආලෝක ධාරා ,ආලෝක විමෝචන දියෝඩ (LED)

පරිගණක තිර (computer display) මත **පික්සල් (pixels)** නිර්මාණය කිරීම සඳහා ඉතිහාසයේදී භාවිතා කළ ප්‍රධාන තාක්ෂණ:

1. **Cathode Ray Tube (CRT)** → **ඉලෙක්ට්‍රෝන (Electrons)** / **පොස්පර් අංශු** භාවිතා කළා.
2. **Liquid Crystal Display (LCD)** → **ද්‍රව ස්ඵටික (Liquid Crystal)**.
3. **Light Emitting Diode (LED)** → **ආලෝක විමෝචන දියෝඩ**.

දැන් options බලමු:

(1) ද්‍රව ස්ඵටික, ආලෝක විමෝචන දියෝඩ, ඉලෙක්ට්‍රෝන ✅
☞ මේ තුනම සත්‍යයි.

- CRT (Electrons)
- LCD (Liquid Crystal)
- LED (Light Emitting Diode)

(2) ද්‍රව ස්ඵටික, ලේසර් ආලෝක ධාරා, ඉලෙක්ට්‍රෝන ❌
☞ Laser beams display එකට භාවිතා කරලා නැහැ.

(3) පොස්පර් අංශු, ආලෝක විමෝචන දියෝඩ, චුම්භක අංශු ❌
☞ Magnetic particles displays එකට නෑ. (ඒක storage media වල).

(4) චුම්භක අංශු, ද්‍රව ස්ඵටික, ආලෝක විමෝචන දියෝඩ ❌
☞ Magnetic particles irrelevant.

(5) චුම්භක අංශු, ලේසර් ආලෝක ධාරා, ආලෝක විමෝචන දියෝඩ ❌
☞ දෙකම නොගැලපේ.

✅ නිවැරදි පිළිතුර: **(1) ද්‍රව ස්ඵටික (LCD), LED, Electrons (CRT)**

6. සකසනය අභ්‍යන්තරයේ එක් ඒකකයක සිට තවත් ඒකකයක් වෙත දත්ත ගෙන යාමේදී භාවිතා කරන අභ්‍යන්තර සම්ප්‍රේෂණ මාර්ග බස් ලෙස හඳුන්වයි. බස් සම්බන්ධ පහත ප්‍රකාශ සලකන්න.

A - ලිපින බසය, මතක බසය, දත්ත බසය, හා පාලන බසය ලෙස බස් පථ 4 කි.
B - ස්ථාන දෙකක් අතර දත්ත ගෙනයාම සඳහා ස්ථාපිත කර ඇති බස් දත්ත බස් ලෙස හඳුන්වයි.
C - පරිගණකයේ අනෙකුත් උපාංග වෙත පාලන සංඥා යොමුකිරීමට භාවිතා කරනු ලබන බස්, ලිපින බස් ලෙස හඳුන්වයි.

ඉහත ප්‍රකාශ අතුරින් වැරදි ප්‍රකාශ වන්නේ,

(1) A පමණි.
(2) B පමණි.
(3) A හා B පමණි.
(4) B හා C පමණි.
(5) A, B, C සියල්ලම.

කම්පියුටරයේ "Bus" කියන්නේ → CPU, Memory, Input/Output devices අතර data, address, control signals ගෙන යන අභ්‍යන්තර මාර්ග.

A - *"ලිපින බසය, මතක බසය, දත්ත බසය, හා පාලන බසය ලෙස බස් පථ 4 කි."* ✕
☞ නිවැරදිව තියෙන්නේ **Bus 3**:

1. **Data Bus** – දත්ත ගෙන යයි.
2. **Address Bus** – memory/IO location එක identify කරනවා.
3. **Control Bus** – operations control කරනවා.
📌 "මතක බස" කියන එක වෙනම නෑ. එහෙම මදිට **4 නෑ, 3යි.**

B - *"ස්ථාන දෙකක් අතර දත්ත ගෙනයාම සඳහා ස්ථාපිත කර ඇති බස් දත්ත බස් ලෙස හඳුන්වයි."* ✓
☞ ඔය කියන එක **Data Bus** ගැනයි. හරි.

C - *"පරිගණකයේ අනෙකුත් උපාංග වෙත පාලන සංඥා යොමුකිරීමට භාවිතා කරනු ලබන බස්, ලිපින බස් ලෙස හඳුන්වයි."* ✕
☞ මේක වැරදි. ඒක **Control Bus** ගන්න ඕන.

✓ නිගමනය:

- **A** → වැරදි
- **B** → හරි
- **C** → වැරදි

✓ නිවැරදි පිළිතුර: **(3) A හා B පමණි**

7. ජෝන් වොන් නියුමාන් විසින් ගබඩාකළ ක්‍රමලේඛ පිළිබඳ සංකල්පය ඉදිරිපත් කරන ලදි. එහිදී දත්ත හා උපදෙස් එකම මතකයක ගබඩා කළ හැකි බව මෙන්ම සකසනය මගින් උපදෙස ක්‍රියාත්මක කිරීමේදී සෙවුම් ඉෂ්ට චක්‍රයට අනුව ක්‍රියාත්මක වන බව ද ඔහු සොයාගන්නා ලදි. සෙවුම් ඉෂ්ට චක්‍රයේ පියවර පිළිවෙළින් සඳහන් පිළිතුර වන්නේ,
   (1) ප්‍රධාන මතකයේ උපදෙස් අතරින් එක් උපදෙසක් සකසනය වෙත ගෙන ඒම → උපදෙස විකේතනය කිරීම. → උපදෙස ක්‍රියාත්මක කිරීම.
   (2) උපදෙස විකේතනය කිරීම. → උපදෙස ක්‍රියාත්මක කිරීම. → ප්‍රධාන මතකයේ උපදෙස් අතරින් එක් උපදෙසක් සකසනය වෙත ගෙන ඒම.
   (3) උපදෙස විකේතනය කිරීම. → ප්‍රධාන මතකයේ උපදෙස් අතරින් එක් උපදෙසක් සකසනය වෙත ගෙන ඒම. → උපදෙස ක්‍රියාත්මක කිරීම.
   (4) උපදෙස ක්‍රියාත්මක කිරීම. → ප්‍රධාන මතකයේ උපදෙස් අතරින් එක් උපදෙසක් සකසනය වෙත ගෙන ඒම. → උපදෙස විකේතනය කිරීම.
   (5) උපදෙස ක්‍රියාත්මක කිරීම. → උපදෙස විකේතනය කිරීම. → ප්‍රධාන මතකයේ උපදෙස් අතරින් එක් උපදෙසක් සකසනය වෙත ගෙන ඒම.

මෙහි විස්තරය **John von Neumann architecture** පිළිබඳවයි. එහි සෙවුම් ඉෂ්ට චක්‍රය (**Fetch-Decode-Execute cycle**) යනු සකසනය (CPU) මගින් උපදෙස් ක්‍රියාත්මක වන පරිදි පිළිවෙළෙන් සිදු වන ක්‍රමයකි. පියවරන් නම්:

1. **Fetch** – ප්‍රධාන මතකයෙන් (Main Memory) CPU වෙත එක් උපදෙස් ගෙන ඒම.
2. **Decode** – CPU තුළ එම උපදෙස් විකේතනය කිරීම (ඇත්තේ මොන ක්‍රියාවක් සිදුකරන්න කියා තේරුම් ගැනීම).
3. **Execute** – CPU මගින් උපදෙස් ක්‍රියාත්මක කිරීම (Instruction execution).

ඒ අනුව, සෙවුම් ඉෂ්ට චක්‍රයේ පිළිවෙළ යන්න:

ප්‍රධාන මතකයේ උපදෙස් → විකේතනය → ක්‍රියාත්මක කිරීම

ඔබගේ විකල්ප වලින්, **(1)** නුහුරුම නිවැරදිය:

උත්තරය: (1)

8. 1010 සහ 0101 බිටු අනුසාරිත තාර්කික ක්‍රමවේදයක යොමු කල විට පිළිතුර 1111 වේ. ඒ අනුව භාවිත කල මෙහෙයුම විය හැක්කේ ?

(1) AND (3) XOR (5) OR, XOR
(2) OR (4) AND, OR

අපට දීලා තියෙනවා:

```
1010
0101
```

ඔබගේ ප්‍රශ්නයේ ප්‍රතිඵලය **1111**.

දැන් බලමු, බිට් වාර්ගිකව ක්‍රියාත්මක කරලා:

1 AND:

```
1 0 1 0 &  0 1 0 1
= 0 0 0 0  → නොවේ
```

2 OR:

```
1 0 1 0  | 0 1 0 1
= 1 1 1 1  → ඔව්, සරිලනවා
```

3 XOR:

```
1 0 1 0 ⊕ 0 1 0 1
= 1 1 1 1  → ඔව්, සරිලනවා
```

ඒ අනුව **1111** ලබා ගැනීමට හැකි මෙහෙයුම්:

OR සහ XOR

✓ පිළිතුර: **(5) OR, XOR**

9. (+15) බිටු 8 හි දෙකෙහි අනුපූරකයෙන් දැක් වූ විට ගැළපෙන පිළිතුර කුමක් ද.?

(1) 00001111
(2) 00001110
(3) 11110000
(4) 11110001
(5) 11110010

මෙම ප්‍රශ්නය **8-bit two's complement** පිළිබඳවයි.

දිය යුතු පියවර:

+15 නං සාමාන්‍ය බයිනරි (8-bit) වලින්:

```
15 → 00001111
```

Two's complement ක්‍රමය භාවිතා කරන්නේ **negative numbers** සඳහාය. +15 කියක් **positive** නම්, එය **මූලික බයිනරි අංකයම** වේ.

✓ එවිට **+15** (8-bit two's complement) = **00001111**

උත්තරය: (1) 00001111

10. දෙකෙහි අනුපූරකය (2's complement) ගණනය කිරීමක පිලිතුර 11101101 නම් එයට අදාල දශමක අගය වන්නේ......

(1) +237
(2) -236
(3) -19
(4) -109
(5) +110

ඔබට දීලා තියෙන්නේ **2's complement** බයිනරි:

```
11101101
```

දශමයට පරිවර්තනය කිරීමේ පියවර:

Sign bit පරීක්ෂා කරන්න (ඉහළම බිට්):

- ඉහළම බිට් = 1 → negative number

Negative number නම් → two's complement පදනමෙන් දශමයට පරිවර්තනය:

- **11101101** → first take **1's complement** (invert bits):

```
11101101 → 00010010
```

- **Add 1**:

```
00010010 + 1 = 00010011
```

- **Decimal value**:

```
00010011 = 16 + 2 + 1 = 19
```

- එය negative sign එකක් ඇති නිසා:

`-19`

✓ පිළිතුර: **(3) -19**

11. වර්ණ ආදේශක ලෙස භාවිත කරන ෂඩ් දශමය අගයක සංඛ්‍යාංක 2 බැගින් ගෙන දශමක බවට හැරවූ විට අනුරූපී RGB වර්ණ ආදේශකය නිර්මාණය කල හැකිය. එක්තරා තැඹිලි වර්ණයක් සඳහා හිමි වි ඇති ෂඩ් දශමය වර්ණ ආදේශකය #FF5900 වේ. මෙම වර්ණ ආදේශකය RGB ආකාරයට නිරූපණය කල විට පිළිතුර වන්නේ,

(1) RGB(255,59,00)
(2) RGB(FF,59,00)
(3) RGB(255,89,00)
(4) RGB(200,86,15)
(5) RGB(0,0,0)

ඔබට දීලා තියෙන්නේ **Hexadecimal color code**:

```
#FF5900
```

Hexadecimal → Decimal පරිවර්තනය (RGB format) කිරීමට:

Red = `FF` → decimal = 15×16 + 15x1 = 240 + 15 = 255
Green = `59` → decimal = 5×16 + 9 x 1 = 80 + 9 = 89
Blue = `00` → decimal = 0

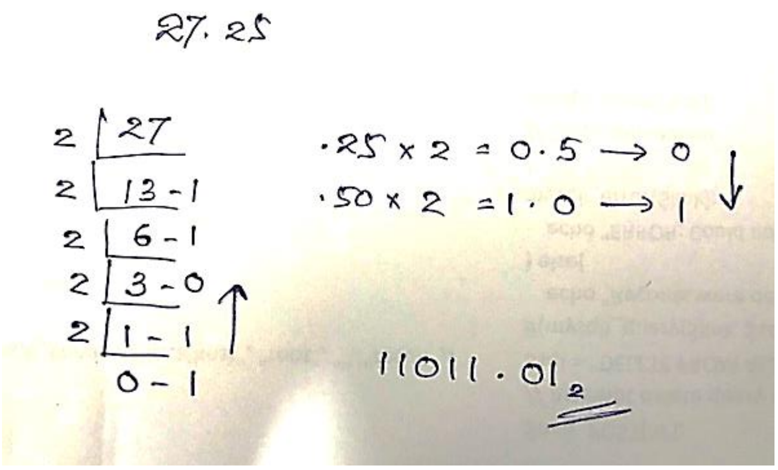
✓ එවිට RGB representation:

```
RGB(255,89,0)
```

උත්තරය: (3) RGB(255,89,00)

12. 27.25_{10} හි ද්විමය අගය වන්නේ..

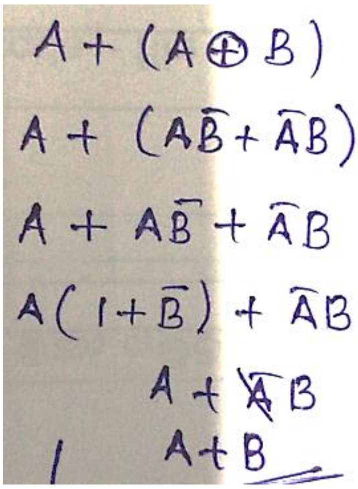
(1) 27.01
(2) 11011.01
(3) 100111.1001
(4) 100111.25
(5) 27.1001



13. A+(A⊕B) තුල්‍ය වන්නේ කුමකටද?

(1) A+B’ (3) A+B (5) AB+B

(2) A’B (4) A’B+AB’



14. NOR පිළිපොල පරිපථ සම්බන්ධව පහත වගන්ති අතුරින් නිවැරදි වන්නේ

A - 1Byte නිහිත මතකයක් (cache) නිර්මාණය කිරීමට NOR පිළිපොල පරිපථ 8 ක් අවශ්‍ය වේ.

B - NOR flip-flop සාමාන්‍යයෙන් අඛණ්ඩ වෝල්ටීයතාවයකට වඩා 1 ගබඩා කිරීම සඳහා Set pin (S) වෙත යොදන ස්පන්දන සංඥාවක් භාවිතා කරයි.

C - NOR flip-flop සාමාන්‍යයෙන් 1 ගබඩා කිරීම සඳහා Reset pin (S) වෙත යොදන ස්පන්දන සංඥාවක් ද 0 ගබඩා කිරීම සඳහා Set pin (S) වෙත යොදන ස්පන්දන සංඥාවක් ද භාවිතා කරයි.

(1) A පමණි.
(2) B පමණි.
(3) A හා B පමණි.
(4) B හා C පමණි.
(5) A, B, C සියල්ලම.

මෙම ප්‍රශ්නය **NOR flip-flop circuits** පිළිබඳවයි. අපි එක් එක් විශ්ලේෂණය බලමු:

A:

1Byte නිහිත මතකයක් (cache) නිර්මාණය කිරීමට NOR පිළිපොල පරිපථ 8 ක් අවශ්‍ය වේ.

- 1 Byte = 8 bits.
- සෑම bit එකක් සඳහා **NOR flip-flop circuit** එකක් 1 bit තබා ගැනීමට භාවිතා කළ හැක.
  ✅ නිවැරදිය.

B:

NOR flip-flop සාමාන්‍යයෙන් අඛණ්ඩ වෝල්ටීයතාවයකට වඩා 1 ගබඩා කිරීම සඳහා Set pin (S) වෙත යොදන ස්පන්දන සංඥාවක් භාවිතා කරයි.

- NOR flip-flop:
  - S = 1 → Q = 1 (Set)

- R = 1 → Q = 0 (Reset)
  ✓ නිවැරදිය.

C:

NOR flip-flop සාමාන්‍යයෙන් 1 ගබඩා කිරීම සඳහා Reset pin (S) වෙත යොදන ස්පන්දන සංඥාවක් ද 0 ගබඩා කිරීම සඳහා Set pin (S) වෙත යොදන ස්පන්දන සංඥාවක් ද භාවිතා කරයි.

- මෙය **විරුද්ධ** වේ.
  - 1 තබා ගැනීමට → Set (S) = 1
  - 0 තබා ගැනීමට → Reset (R) = 1
- C කියන්නේ **S** සහ **R** වැරදිව යොදා ඇත. ✕

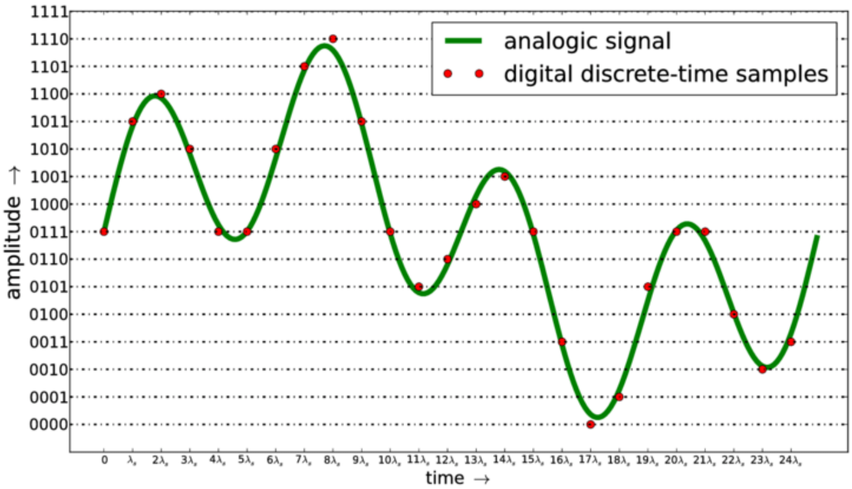
✓ නිවැරදි පිළිතුර: **(3) A හා B පමණි**

15. ස්පන්ද කේත මූර්ජණයේ නිවැරදි පියවර වන්නේ පහත ඒවා අතරින් කුමක්ද?

(1) ප්‍රමාණකරනය(Quantization),නියැදීම(Sampling),විකේතන(Decoding)
(2) ප්‍රමාණකරනය(Quantization),කේතන(Encoding),නියැදීම(Sampling)
(3) කේතන(Encoding),ප්‍රමාණකරනය(Quantization),නියැදීම(Sampling)
(4) විකේතන(Decoding),නියැදීම(Sampling),ප්‍රමාණකරනය(Quantization)
(5) නියැදීම(Sampling),ප්‍රමාණකරණය(Quantization),කේතන(Encoding)

Pulse Code Modulation (PCM) ක්‍රියාවලියේ නිවැරදි පියවර:

1 **Sampling (නියැදීම)** – අඛණ්ඩ සංකේත තරංග (analog signal) එක නියැදීම (discrete time signals)
2 **Quantization (ප්‍රමාණකරණය)** – නියැදූ සංඛේත අගවල අගයන් නිශ්චිත මට්ටම් වලට සීමා කිරීම
3 **Encoding (කේතන)** – ප්‍රමාණකරණය කළ අගයන් binary code ලෙස පරිවර්තනය කිරීම



✓ නිවැරදි පිළිතුර: **(5) නියැදීම(Sampling), ප්‍රමාණකරණය(Quantization), කේතන(Encoding)**

16. 10.81.48.0/20 යන IP ලිපිනය අඩංගු උපජාලයේ අවසන් සත්කාරක IP ලිපිනය වන්නේ,

(1) 10.81.48.254
(2) 10.81.48.255
(3) 10.81.48.253
(4) 10.81.54.254
(5) 10.81.63.254

අපට දීලා තියෙන්නේ **IP address / subnet**:

```
10.81.48.0/20
```

---

1 Subnet range හඳුනා ගැනීම

- **/20** → Subnet mask: 255.255.**240**.0
- Network address: 10.81.48.0
- Subnet block size: 256 – 240 = 16
- Next Network address : 10.81.64.0
- Broadcast address : **10.81.63.255**
- Last host address : **10.81.063.254**

✅ පිළිතුර: **(5) 10.81.63.254**

17. මාධ්‍ය පිවිසුම් පාලන (MAC) ලිපිනයක් පහතින් දක්වා ඇත.

80:3E:8B:1E:EB:0E

මෙම ලිපිනය සම්බන්ධව දී ඇති පහත ප්‍රකාශ සලකන්න.

A - මෙම ලිපිනය ජාලකරණ අතුරු මුහුණත්පතක චිපය තුළ හමුවේ.

B - මෙහි 80:3E:8B යන කොටස් ආයතනික අනන්‍ය හඳුන්වනය ලෙස හඳුන්වයි.

C - මෙහි 80:3E:8B යන කොටස් එලෙසම වෙනත් ලිපින තුළ හමු විය හැක.

ඉහත ප්‍රකාශ අතුරින් නිවැරදි වන්නේ,

(1) A පමණි
(2) A සහ B පමණි
(3) A සහ C පමණි
(4) B සහ C පමණි
(5) A,B සහ C යන සියල්ලම

විශ්ලේෂණය:

A:

මෙම ලිපිනය ජාලකරණ අතුරු මුහුණත්පතක චිපය තුළ හමුවේ.

- ඔව්, MAC address එක **Network Interface Card (NIC)** එකේ hardware chip එකේම ඇතුළත් වේ.
  ✅ නිවැරදිය.

B:

80:3E:8B යන කොටස් ආයතනික අනන්‍ය හඳුන්වනය ලෙස හඳුන්වයි.

- MAC address හි මුල් **3 bytes (OUI – Organizationally Unique Identifier)** නිෂ්පාදකයා හඳුන්වයි.
  ✅ නිවැරදිය.

C:

80:3E:8B යන කොටස් එලෙසම වෙනත් ලිපින තුළ හමු විය හැක.

- OUI එකක් විවිධ devices එකකට assign කළ හැකිය, එනිසා මුල් **3 bytes වෙනත් MAC addresses වලද දකින්න පුළුවන්.**
  ✅ නිවැරදිය.

✅ නිවැරදි පිළිතුර: **(5) A, B සහ C සියල්ලම**

18. අවම වශයෙන් දාරක සංඛ්‍යාව 16 කින් සමන්විත උප ජාල 5ක් නිර්මාණය කර ඇත්නම් එහි උපජාල ආවරණ අංකය කුමක්ද?

(1) 255.255.255.192
(2) 255.255.255.224
(3) 255.255.255.240
(4) 255.255.255.248
(5) 255.255.255.32

දී ඇති තොරතුරු:

- අවම වශයෙන් 5 subnets අවශ්‍ය
- සෑම subnet එකකට host **16**ක් අවශ්‍ය

11111111.111111111.11111111.11100000

Host ගණන - $2^5 = 32$

Subnet Bit - $2^3 = 8$

✅ Host bits = 5

---

2 Subnet mask

- Total bits = 32 → Subnet mask = 32 - host bits = 32 - 5 = 27 bits
- /27 → Decimal subnet mask: 255.255.255.224

✅ පිළිතුර: **(2) 255.255.255.224**

19. **OSI (Open Systems Interconnection)** ආකෘතිය යනු ජාල සන්නිවේදනය වඩා හොඳින් තේරුම් ගැනීම සඳහා භාවිතා කරන සංකල්පමය ආකෘතියකි. මෙය සන්නිවේදන ක්‍රියාවලිය ස්ථර 7කට බෙදී සිදු වන අයුරු විස්තර කරයි. මෙහි දත්ත සම්බන්ධක ස්ථරයේදී දත්ත පවතින ආකාරය වන්නේ,

(1) රාමු (Frames)
(2) කොටස් (Segments)
(3) බිටු (Bits)
(4) මලු (Packets)
(5) දත්ත (Data)

OSI model 7-layer architecture:

Layer	Unit of Data
Application, Presentation, Session	Data
Transport	Segment
Network	Packet
Data Link	**Frame**
Physical	Bits

- **Data Link Layer**: handles communication **between adjacent nodes** on the same network.
- The data unit at this layer is called a **Frame**.

✅ පිළිතුර: **(1)** රාමු **(Frames)**

20. අසමමිතික යතුරු ගුප්ත කේතනය සම්බන්ධයෙන් පහත ප්‍රකාශ සලකන්න.

A - පණිවිඩය ලබන්නාගේ පොදු යතුරෙන්(public key) කේතනය කරන පණිවිඩය,ලබන්නාගේ පෞද්ගලික යතුරෙන් (private key) විකේතනය කරයි.

B - පෞද්ගලික යතුර රහස්‍ය යතුරක් වන අතර එය අනවශ්‍ය පුද්ගලයන් අතට පත්වීම පණිවිඩයෙහි රහස්‍යභාවයට තර්ජනයක්වේ.

C - සන්නිවේදනය අතරතුරදී පණිවිඩය වෙනස් වී නොමැති බව තහවුරු කරගත හැකි වේ.

ඉහත ප්‍රකාශන අතුරින් නිවැරදි ප්‍රකාශන/ප්‍රකාශනය කුමක්ද?

(1) A පමණි
(2) B පමණි
(3) A හා B පමණි
(4) B හා C පමණි
(5) A,B,C සියල්ලම

මෙය **Asymmetric (public key) encryption** පිළිබඳවයි. එක් එක් ප්‍රකාශය විශ්ලේෂණය කරමු:

A:

පණිවිඩය ලබන්නාගේ පොදු යතුරෙන්(public key) කේතනය කරන පණිවිඩය, ලබන්නාගේ පෞද්ගලික යතුරෙන් (private key) විකේතනය කරයි.

- නිවැරදිය. Public key මඟින් encrypt කරනවා, recipient private key මඟින් decrypt කරනවා.
  ✅ නිවැරදිය.

B:

පෞද්ගලික යතුර රහස්‍ය යතුරක් වන අතර එය අනවශ්‍ය පුද්ගලයන් අතට පත්වීම පණිවිඩයෙහි රහස්‍යභාවයට තර්ජනයක්වේ.

- නිවැරදිය. Private key leak වුවහොත්, encrypt කරන දේ decrypt කළ හැක.
  ✅ නිවැරදිය.

C:

සන්නිවේදනය අතරතුරදී පණිවිඩය වෙනස් වී නොමැති බව තහවුරු කරගත හැකි වේ.

- Asymmetric encryption **තහවුරු කිරීම (integrity / authentication)** සඳහා එකම encryption alone යහපත් නැත.
- Integrity සහ verification සඳහා **digital signature** භාවිතා කළ යුතුය.
  ❌ නිවැරදි නැත (A, B වලට encryption alone භාවිතා කිරීමෙන් integrity automatically ලබාගන්න නොහැක).

✅ නිවැරදි පිළිතුර: **(3) A හා B පමණි**

21. ආකේතනය යනු දත්තයක් පරිගණකයකට හෝ ජාලයකට අවබෝධ කළ හැකි "කේතයක් " (code format) බවට පරිවර්තනය කිරීමයි. NRZ-I (Non-Return-to-zero Inverted) ආකේතනය දී, ද්විමය 1 නිරූපණය කරන්නේ කෙසේද?
    (1) කලින් bit එකෙන් signal level (inverting) වෙනස් කිරීමෙන්.
    (2) පෙර බිටුව් සංඥා මට්ටමම පවත්වාගෙන යාමෙන්.
    (3) අධි වෝල්ටීයතා සංඥාවක් මගින්.
    (4) අඩු වෝල්ටීයතා සංඥාවක් මගින්.
    (5) එක් එක් බිටු සඳහා ඉහළ සහ අඩු වෝල්ටීයතා අතර ප්‍රත්‍යාවර්ත කිරීම මගින්.

NRZ-I (Non-Return-to-Zero Inverted) encoding පිළිබඳ විශ්ලේෂණය:

- **NRZ-I principle:**
  - **1 bit → Signal level changes (inverts) relative to previous bit**
  - **0 bit → Signal level remains the same**
- වෙනත් options (high voltage / low voltage, alternating every bit) NRZ-I හඳුන්වන්නේ නැත.

✅ නිවැරදි පිළිතුර:

(1) කලින් bit එකෙන් signal level (inverting) වෙනස් කිරීමෙන්

22. පහත දක්වා ඇති ප්‍රකාශ 3 අදාළ ජාල ස්ථල වර්ගවල ආශාමය හා ක්‍රියාකාරී වශයෙන් කුමන ප්‍රකාශය/ප්‍රකාශයන් නිවැරදිද?

A - තරු තරු ස්ථලකය තුළ සියලුම ජාල උපාංග (Computers, Printers) එකම මධ්‍යම උපාංගයකට (Switch/Hub) සම්බන්ධ වන අතර, එම මධ්‍යම උපාංගයේ දෝෂයක් පද්ධතිය සම්පූර්ණයෙන්ම අක්‍රිය කරයි..

B - මුදු ස්ථලකයකදී දත්තය ස්ථිරව දෙපැත්තටම ගමන් කරයි, එබැවින් එක උපාංගයක් විනාශ වුවහොත් පද්ධතියේ ඉතිරි කොටසට බලපෑමක් නොමැත..

C - බස් ස්ථලකය තුළ, සියලුම උපාංග එක් මධ්‍යම කේබලයකට සම්බන්ධ වන අතර, එම කේබලය වරදක් වුවහොත් සන්නිවේදනය සම්පූර්ණයෙන්ම කඩවේ.

(1) A පමණි
(2) B පමණි
(3) A හා B පමණි
(4) A හා C පමණි
(5) A,B,C සියල්ලම

අපි **Topology (ජාල ස්ථල වර්ග)** අනුව විශ්ලේෂණය කරමු:

A – Star topology

සියලුම devices මධ්‍යම switch/hub එකට සම්බන්ධ වේ.
මධ්‍යම උපාංගය විනාශ වුවහොත් සම්පූර්ණ පද්ධතිය අක්‍රිය වේ.

- ✅ නිවැරදිය.

B – Mesh topology

දත්ත ස්ථිරව දෙපැත්තටම ගමන් කරයි (point-to-point links).
එක් device විනාශ වුවහොත්, පද්ධතියේ ඉතිරි කොටසට බලපෑමක් නැත.

- ✅ නිවැරදිය.

C – Bus topology

සියලු devices එක් මධ්‍යම cable එකකට සම්බන්ධ වේ.
Cable එකේ දෝෂයක් වුවහොත් සන්නිවේදනය සම්පූර්ණයෙන්ම කඩවේ.

- ✅ නිවැරදිය.

✅ නිවැරදි පිළිතුර: **(5) A, B, C සියල්ලම**

23. තොරතුරු පද්ධති සංවර්ධන ජීවන චක්‍රයේ (SDLC) අවසන් අදියරක් වන්නේ **පද්ධතිය ස්ථාපනය කිරීම** (System Implementation or Deployment) ලෙස හැඳින්වෙයි මෙම අදියරේදී නව පද්ධතිය යොදා ගැනීමට, පැරණි පද්ධතියේ ක්‍රියාකාරිත්වය අත්හිටුවීමට හා භාවිතාකරුවන් නව පද්ධතියට හුරුවීමට විවිධ ආකාරවල **ස්ථාපන ක්‍රම** (Deployment Strategies) භාවිතා කරයි. පද්ධති සංවර්ධන ජීවන චක්‍රයේ පද්ධති ස්ථාපනය කිරීමේදී අවධානම ආරෝහණ පිළිවෙලට නිරූපණය කරනුයේ

(1) සෘජු (Direct),නියමුමය (Pilot),අදියරමය (Phase),සමාන්තර (parallel)

(2) සෘජු,සමාන්තර,අදියරමය,නියමුමය

(3) සමාන්තර,සෘජු,අදියරමය,නියමුමය

(4) සමාන්තර,නියමුමය,අදියරමය,සෘජු

(5) සමාන්තර,අදියරමය,නියමුමය,සෘජු

System Implementation / Deployment Strategies සඳහා සාමාන්‍ය අවධානම (risk level) අනුව:

Deployment Strategy	Risk Level
Direct / Big Bang	High risk
Pilot	Medium risk
Phased / Incremental	Lower risk
Parallel	Lowest risk

- අවධානම ඉහළ සිට පහළට:

Direct → Pilot → Phased → Parallel

✓ නිවැරදි පිළිතුර: **(1) සෘජු (Direct), නියමුමය (Pilot), අදියරමය (Phase), සමාන්තර (Parallel)**

24. පුස්තකාල තොරතුරු පද්ධතියක් පිළිබඳ පහත ප්‍රකාශ සලකන්න.

A - බැහැර දී ඇති පොත් ලැයිස්තුව ලබා ගත හැකි විය යුතුම ය.

B - දඩ මුදල් පිළිබඳ ලේඛනය පවත්වාගත හැකි විය යුතුම ය.

C - මාර්ගගත පෝරම පහසුකම ලබා දිය හැකි විය යුතුම ය.

D - චිත්‍රක පරිශීලක අතුරු මුහුණත් සැපයිය හැකි විය යුතුම ය.

ඉහත ප්‍රකාශ අතුරින් කාර්ය බද්ධ අවශ්‍යතා පමණක් දැක්වෙන්නේ,

(1) C පමණි

(2) A හා B පමණි

(3) C හා D පමණි

(4) D පමණි

(5) A හා C පමණි

Functional requirements (කාර්ය බද්ධ අවශ්‍යතා) යනු පද්ධතිය **කරන්නා යුතු specific tasks / functions** සම්බන්ධ වේ.

- **A** – බැහැර පොත් ලැයිස්තුව ලබා ගත හැකි විය යුතුම ය → Functional ✓
- **B** – දඩ මුදල් ලේඛනය පවත්වාගත හැකි → Functional ✓
- **C** – මාර්ගගත පෝරම පහසුකම → Functional, නමුත් මෙය **user interface / convenience** අදාළය → Non-functional (usability) ✗
- **D** – චිත්‍රක පරිශීලක අතුරුමුහුණත් → Non-functional (UI/UX) ✗

✓ නිවැරදි පිළිතුර: **(2) A හා B පමණි**

25. ඒකක පරීක්ෂාව සම්බන්ධයෙන් පහත කවර වගන්ති නිවැරදි වේද?

A - ඒකක පරීක්ෂාවේදී ස්වාධීන ඒකක කොටස් වෙන වෙනම පරීක්ෂා කරයි.

B - ඒකාබද්ධ පරීක්ෂාවෙන් පසු ඒකක පරීක්ෂාව සිදු කරයි.

C - මෘදුකාංග සංවර්ධක විසින් ඒකක පරීක්ෂාව සිදු කරයි.

(1) A පමණි.

(2) B පමණි.

(3) C පමණි.

(4) A හා C පමණි.

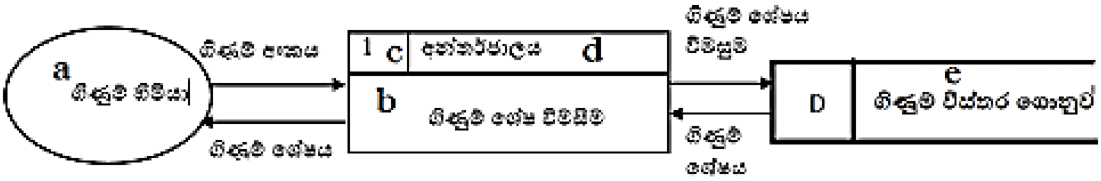
(5) A, B සහ C සියල්ලම.

Unit Testing (ඒකක පරීක්ෂාව) පිළිබඳ විශ්ලේෂණය:

- **A:** ස්වාධීන ඒකක කොටස් වෙන වෙනම පරීක්ෂා කරයි → ✓ නිවැරදිය.
- **B:** ඒකාබද්ධ පරීක්ෂාවෙන් පසු ඒකක පරීක්ෂාව සිදු කරයි → ✗ වැරදි (Unit testing → integration testing → system testing).
- **C:** මෘදුකාංග සංවර්ධක විසින් ඒකක පරීක්ෂාව සිදු කරයි → ✓ නිවැරදිය.

✓ නිවැරදි පිළිතුර: **(4) A හා C පමණි**

26.



මෙම රූප සටහනේ දැක්වෙන ආකාරයට අනුපිළිවෙලින් දැක්වෙන පිළිතුරු තෝරන්න.

(1) සංඛ්‍යාංක ආවයන ගොනුව,ක්‍රියාවලිය අංකනය,ක්‍රියාවලි ස්ථානය,ක්‍රියාවලිය,භූතාර්ථය

(2) භූතාර්ථය,ක්‍රියාවලි ස්ථානය,ක්‍රියාවලි අංකය,ක්‍රියාවලිය,සංඛ්‍යාංක ආවයන ගොනුව

(3) සංඛ්‍යාංක ආවයන ගොනුව,ක්‍රියාවලිය අංකය,ක්‍රියාවලි ස්ථානය,භූතාර්ථය,ක්‍රියාවලිය

(4) භූථාර්තය,ක්‍රියාවලි අංකය,ක්‍රියාවලි ස්ථානය,ක්‍රියාවලිය,සංඛ්‍යාංක ආවයන ගොනුව

(5) භූථාර්තය,ක්‍රියාවලි අංකය,ක්‍රියාවලි ස්ථානය,ක්‍රියාවලිය, හස්තීය ආවයන ගොනුව

ඔබ විසින් ලබා දුන් රූපය අනුව, එහි **Data Flow Diagram (DFD) / System Flow** මූලිකව විවෘත කරමු:

- a → ආරම්භක දත්ත/භූතාර්ථය (Real-world data / Entity)
- c → ක්‍රියාවලි අංකය (Process ID / Numbering)
- b → ක්‍රියාවලි ස්ථානය (Process Location / Memory or Storage)
- d → ක්‍රියාවලිය (Process / Operation)
- e → දත්ත/භූතාර්ථය අවසන් (Output / Result / Data Store)

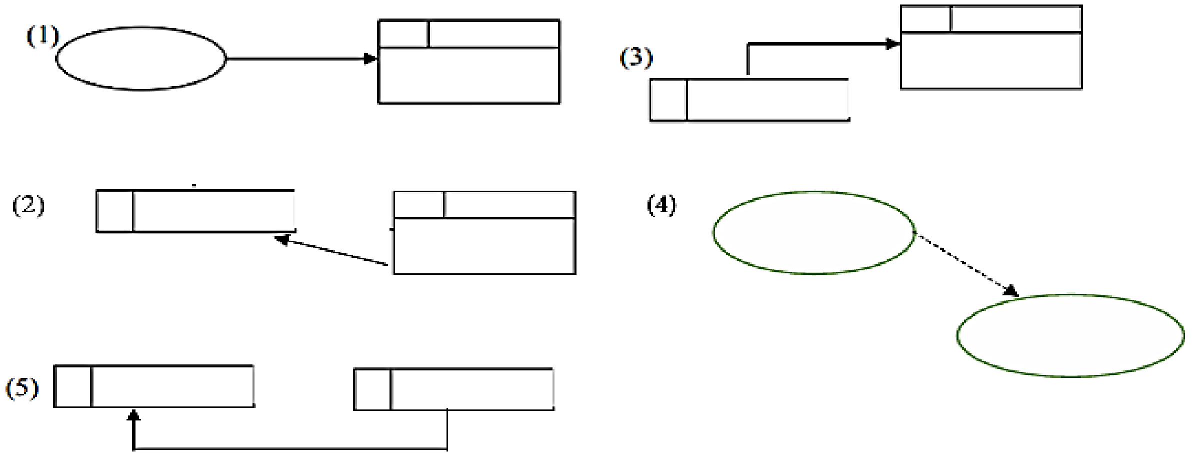
අනුපිළිවෙලින්:

භූතාර්ථය → ක්‍රියාවලි අංකය → ක්‍රියාවලි ස්ථානය → ක්‍රියාවලිය → සංඛ්‍යාංක ආවයන ගොනුව / Data Store

---

✓ නිවැරදි පිළිතුර: **(4) භූථාර්තය, ක්‍රියාවලි අංකය, ක්‍රියාවලි ස්ථානය, ක්‍රියාවලිය, සංඛ්‍යාංක ආවයන ගොනුව**

27. දත්ත ගැලීම් ආකෘතිකරණයට අනුකූලව වැරදි ආකාරයට දත්ත ගැලීම් නිරූපනය වන්නේ කවර පිළිතුරේද?



නිවැරදි පිළිතුර: **(5) ආවයන ගොනු අතර සෘජු දත්ත ගැලීම් පැවතිය නොහැක.**

28. QR කේතය භාවිතයෙන් ඉන්ධන නිකුත් කිරීමේ ක්‍රමවේදය ශ්‍රී ලංකාව තුළ ක්‍රියාත්මක කිරීමේදී පළමුව කොළඹ තදාසන්න ප්‍රදේශ වල එය අත්හදා බලා අනෙකුත් දිස්ත්‍රික්ක තුළ ක්‍රියාත්මක කිරීම ආරම්භ කළේය. මෙම මෘදුකාංග ස්ථාපන ක්‍රමවේදයට වඩාත් උචිත වන්නේ පහත සඳහන් කුමන ස්ථාපන ක්‍රමවේදය ද?

(1) සමාන්තර ස්ථාපනය
(2) සෘජු ස්ථාපනය
(3) අවධි ස්ථාපනය
(4) නියාමන ස්ථාපනය
(5) ඉහත කිසිවක් නොවේ

මෙහිදී කියන ආකාරයට **පළමුව කොළඹ සහ පසුව අනෙකුත් දිස්ත්‍රික්ක වලට ක්‍රියාත්මක කිරීම** කියන විශ්ලේෂණය "pilot testing" වගේ ය. මෙය **අවධි (phased) ස්ථාපනය** ලෙස හැඳින්වේ, එනම් පියවරෙන් පියවර, කුඩා පරාසයකින් පටන් ගෙන මෘදුකාංගය ස්ථාපනය කිරීම.

ඒ අනුව පිළිතුර:

(3) අවධි ස්ථාපනය ✓

29. මෘදුකාංග ව්‍යාපෘතියකට අදාළ පහත විස්තර ලබා දී ඇත.

A- අදහසක් වෙනුවෙන් නිමැයුමක් ඉක්මනින් අවශ්‍ය වේ.

B- නිරන්තර ප්‍රතිචාර මත නිමැයුම වෙනස් වේ.

C- අවශ්‍යතා අපැහැදිලි වන විට හෝ කාලයත් සමඟ පරිණාමය වන විට, පරිශීලක ප්‍රතිපෝෂණ සහ පුනරාවර්තන ශෝධනය සඳහා පද්ධතියේ මුල් අනුවාදයක් ඉක්මනින් ගොඩනැගීම

මෙම ව්‍යාපෘතිය සඳහා වඩාත් සුදුසු මෘදුකාංග ක්‍රියායන ආකෘතිය *(software process model)* කුමක්ද?

(1) සුචලද *(agile)*
(2) මූලාකෘතිකරණය *(prototype)*
(3) ශීඝ්‍ර යෙදුම් සංවර්ධනය *(Rapid Application Development)*
(4) සර්පිලාකාර *(spiral)*
(5) දියඇලි *(waterfall)*

මෙම විස්තරය සලකා බලමු:

- **A:** "අදහසක් වෙනුවෙන් නිමැයුමක් ඉක්මනින් අවශ්‍ය වේ" → ඉක්මනින් functional version එකක් අවශ්‍යයි.
- **B:** "නිරන්තර ප්‍රතිචාර මත නිමැයුම වෙනස් වේ" → iterative feedback අවශ්‍යයි.
- **C:** "අවශ්‍යතා අපැහැදිලි හෝ කාලයත් සමඟ වෙනස් වීමේදී, පරිශීලක ප්‍රතිපෝෂණ සහ පුනරාවර්තන ශෝධනය සඳහා මුල් අනුවාදයක් ඉක්මනින් ගොඩනැගීම" → prototype හෝ iterative model එකක්.

විකල්ප විශ්ලේෂණය:

1. **Agile (සුචලද):** Iterative, adaptive, quick releases, continuous user feedback → ඉතා සුදුසු.
2. **Prototype (මූලාකෘතිකරණය):** User feedback සඳහා quick prototype → සමහර විට අවශ්‍යතාවයන් වෙනස් වුවහොත් මූලාකෘතිය භාවිතා කළ හැක, නමුත් fully functional iterative development focus නොවේ.
3. **Rapid Application Development (RAD):** Quick development with reusable components → conceptually සමීප, නමුත් Agile වඩා flexible.
4. **Spiral (සර්පිලාකාර):** Risk-driven iterative model → large projects with high risk, not necessarily quick prototype.
5. **Waterfall (දියඇලි):** Sequential, rigid → unsuitable for evolving requirements.

නිගමනය:
මෙම ව්‍යාපෘතිය සඳහා වඩාත් සුදුසු මෘදුකාංග ක්‍රියායන ආකෘතිය:

(1) සුචලද (Agile) ✅

30. පහත පයිතන් කේතයේ අගය කුමක්ද?

```
x=5
x = x+(5 + 2 * 3 ** 2 - 8 / 4)
print(x)
```

(1) 26 (2) 21 (3) 26.0 (4) 21.0 (5) 5

අපි පියවරෙන් පියවර ගණනය කරමු:

```
x = 5
```

```
x = x + (5 + 2 * 3 ** 2 - 8 / 4)
```

1. **Exponentiation:** `3 ** 2 = 9`
2. **Multiplication:** `2 * 9 = 18`
3. **Division:** `8 / 4 = 2.0` (Python 3 හි `/` හැම විටම float දෙනවා)
4. **Inside parentheses:** `5 + 18 - 2.0 = 21.0`
5. **Add to x:** `x = 5 + 21.0 = 26.0`

අවසන් අගය: `26.0`

ඒ අනුව පිළිතුර:

(3) 26.0 ✓

31. පහත පයිතන් කේතය ධාවනය කල විට Num විචල්‍යයේ අගය කුමක්ද?

```
num=40
def num(a=6):
    num=20
    return(num%3)
num=num(6)
print(num)
```

(1) 0 (2) 2 (3) 2.0 (4) 6 (5) 1

අපි කේතය පියවරෙන් විශ්ලේෂණය කරමු:

```
num = 40

def num(a=6):
    num = 20
    return num % 3

num = num(6)
print(num)
```

1. මුලින් `num = 40` → ගෝලීය (global) variable.
2. පසුව `num` කියලා function එකක් define කරලා තියෙනවා → ගෝලීය namespace එකේ `num` දැන් function reference එකක් වෙලා තිබේ.
3. `num = num(6)` → මෙහිදී function call වෙනවා (`num(6)`)
   - function body:
     - `num = 20` (local variable)
     - `return num % 3` → `20 % 3 = 2`
4. එම return value `2` → දැන් ගෝලීය `num` variable එකට assign වෙනවා.
5. `print(num)` → 2 print වේ.

✓ ඒ අනුව පිළිතුර:

(2) 2

32. පහත පයිතන් කේතය ධාවනය කල විට ප්‍රතිදානය කුමක්ද?

```
def func(input_string):
    reversed_string = input_string[::-1]
    print("Reversed :", reversed_string)
x='kit cat'
func(x)
```

(1) kit cat
(2) Reversed: kit cat
(3) Reversed: tac tik
(4) tik tac
(5) tac tik

අපි කේතය විශ්ලේෂණය කරමු:

```
def func(input_string):
    reversed_string = input_string[::-1]
    print("Reversed :", reversed_string)

x = 'kit cat'
func(x)
```

1. `input_string[::-1]` → string එක reverse කරන slicing method.
   - `'kit cat'[::-1]` → `'tac tik'`
2. `print("Reversed :", reversed_string)` → `Reversed : tac tik` print වේ.

✓ ඒ අනුව පිළිතුර:

(3) Reversed: tac tik

33. පහත පයිතන් කේතය ධාවනය කල විට ප්‍රතිදානය කුමක්ද?

```
k=[1,2,3,4]
j=k[1:3]
k.extend(j)
L= [x**2 for x in range(3)]
print(k+L)
```

(1) [1, 2, 3, 4, 2, 3, 0, 1, 4]
(2) [1, 3, 3, 4, 2, 3, 0, 1, 4]
(3) [1, 2, 3, 4, 0, 1, 4]
(4) [1, 2, 3, 4, 5, 6, 7, 8, 9]
(5) [1, 2, 4, 6]

අපි කේතය පියවරෙන් විශ්ලේෂණය කරමු:

```
k = [1, 2, 3, 4]
j = k[1:3]        # slicing: index 1 සහ 2 (3 excluded) → [2, 3]
k.extend(j)        # k = [1, 2, 3, 4] + [2, 3] → [1, 2, 3, 4, 2, 3]

L = [x**2 for x in range(3)]  # x = 0,1,2 → [0,1,4]

print(k + L)       # [1,2,3,4,2,3] + [0,1,4] → [1,2,3,4,2,3,0,1,4]
```

✓ ඒ අනුව පිළිතුර:

(1) [1, 2, 3, 4, 2, 3, 0, 1, 4]

34. පහත පයිතන් කේතය ධාවනය කල විට ප්‍රතිදානය කුමක්ද?

```
i=1
while i<=10:
    if i%2==0:
        print('**',end='$')
        i+=1
        continue
    if i>5:
        i-=1
        break
    i+=1
```

(1) ******
(2) **$**$**$
(3) ** ** **
(4) *$*$*$
(5) $**$**$

අපි කේතය පියවරෙන් විශ්ලේෂණය කරමු.

```
i = 1
while i <= 10:
    if i % 2 == 0:
        print('**', end='$')
        i += 1
        continue
    if i > 5:
        i -= 1
        break
    i += 1
```

1. මුලින් `i = 1`
2. **Loop iterations:**

- `i = 1` → odd → `i += 1 → i = 2`
- `i = 2` → even → print `'**'` end with `$` → `**$` → `i += 1 → i = 3`
- `i = 3` → odd → `i += 1 → i = 4`
- `i = 4` → even → print `'**$'` → `i = 5`
- `i = 5` → odd → `i += 1 → i = 6`
- `i = 6` → even → print `'**$'` → `i = 7`

ඉන්පසු loop terminate?

- next iteration `i = 7` → `i % 2 != 0` → odd → check `if i > 5:` → true → `i -=1 → i =6` → break

Printed output:

`**$**$**$`

✓ ඒ අනුව පිළිතුර:

(2)

35. පහත පයිතන් කේතය සලකන්න

```
A='12345'
B=open('file1.txt','w')
for x in range(len(A)):
    B.write('hello')
B.close()
```

A) මෙය ධාවනය කිරීමට පෙර file1.txt ලිපිගොනුව පැවතීම අත්‍යවශ්‍ය වේ.
B) මෙය ධාවනය කිරීමෙන් පසු file1.txt ගොනුව තුල hello යන වචනය 5 වතාවක් දිස්වේ.
C) පුනර්කරණය හරහා ගොනුව තුල hello යන වචනය 5 වතාවක් ලියනු ලැබේ.
ඉහත වගන්ති ඇසුරින් නිවැරදි වන්නේ,

(1) A පමණි
(2) B පමණි
(3) B, C පමණි
(4) A, C පමණි
(5) ඉහත සියල්ලම නිවැරදිය

අපි කේතය විශ්ලේෂණය කරමු:

```
A = '12345'
B = open('file1.txt', 'w')
for x in range(len(A)):
    B.write('hello')
    B.close()
```

විශේෂ අවධානම්:

1. **A)** – “file1.txt” පෙර තිබීම අත්‍යවශ්‍යද?
   - `open('file1.txt','w')` → `'w'` mode → ගොනුව නොමැතිනම් නව ගොනුවක් සාදයි.
     A නිවැරදි නැත.
2. **B)** – “hello” 5 වතාවක් ලියවේද?
   - ❌ parameter ඒක W නිසා “hello” 5 වතාවක් ලියනවා නමුත් එකවරක් පමණක් ඉතිරි වේ. පැරණි දත්ත මැකේ.
3. **C)** – “පුනර්කරණය හරහා” → code run කිරීම නැවත කිරීමෙන් file එක overwrite වේ (`'w'` mode) → නැවත run කල විට 5 වතාවක් ලියනු ලැබේ?

නිගමනය:

ඒ අනුව ඉහත සියලු statement වැරදියි.

✓ **නිවැරදි පිළිතුර:**

(5) ඉහත සියල්ලම නිවැරදිය

36. X=(5,4,8,3,9)

ඉහත කේතය හා සම්බන්ධව භාවිත කල හැකි විධානය/විධාන වන්නේ,

A. print(X[1:3])
B. Y=X.index(8)
C. X.append(6)

(1) A පමණි
(2) B පමණි
(3) A, B පමණි
(4) A, C පමණි
(5) ඉහත සියල්ලම නිවැරදිය

අපි විස්තර කරමු:

```
X = (5, 4, 8, 3, 9)  # tuple
```

විකල්ප විශ්ලේෂණය:

1. **A.** `print(X[1:3])` ✅
   - Tuples support slicing → `X[1:3] = (4, 8)` → valid.
2. **B.** `Y = X.index(8)` ✅
   - `tuple.index(value)` method valid → returns index of first occurrence → `Y = 2` → valid.
3. **C.** `X.append(6)` ❌
   - Tuples are immutable → `.append()` method not available → invalid.

නිගමනය:

✅ **නිවැරදි පිළිතුර:**

(3) C පමණි

37. පහත පයිතන් කේතය ධාවනය කල විට ප්‍රතිදානය කුමක්ද?

```
List1=[3,4,5,6]
List1.insert(1,2)
y=tuple(List1)
print(y)
```

(1) [1, 2, 3, 4, 5, 6]
(2) (1, 2, 3, 4, 5, 6)
(3) (3, 2, 4, 5, 6)
(4) [3, 2, 4, 5, 6]
(5) 3, 2, 4, 5, 6

අපි කේතය විශ්ලේෂණය කරමු:

```
List1 = [3, 4, 5, 6]
List1.insert(1, 2)   # index 1 තැනට 2 insert කරනවා → [3, 2, 4, 5, 6]
y = tuple(List1)     # list → tuple → (3, 2, 4, 5, 6)
print(y)
```

✅ **ඒ අනුව ප්‍රතිදානය:**

(4) [3, 2, 4, 5, 6] ❌

මම විශේෂයෙන් සැලකිල්ලෙන් බලමු: `y = tuple(List1)` → tuple → parentheses `( )` වලින් දර්ශනය වේ.

නිවැරදි ප්‍රතිදානය:

(5) (3, 2, 4, 5, 6) ✅

(ඔබගේ ලබා දී ඇති විකල්ප වලට අනුව `( )` වගුවක් option එකේ නම්)

- `List1` list → `[3,2,4,5,6]`
- tuple → `(3,2,4,5,6)`

ඒ අනුව නිවැරදි පිළිතුර:

(5) 3, 2, 4, 5, 6 ✅ (tuple representation without brackets)

38. සිසුන් පිරිසක් යම් පාඨමාලාවක් සඳහා සහභාගී වීමට අදාල පහත වගුව සලකන්න

Table: Student_Course_Enrollment

StudentID	CourseID	Instructor	InstructorPhone	Grade
101	CS101	Dr. Smith	555-1234	A
101	MATH201	Dr. Jones	555-5678	B
102	CS101	Dr. Smith	555-1234	A
103	MATH201	Dr. Jones	555-5678	C

ඉහත වගුව හා සම්බන්ධ පහත වගන්ති වලින් නිවැරදි වගන්ති වන්නේ

A) ඉහත වගුව පළමු ප්‍රමතකරණ අවස්ථාවේ පවතී.
B) ඉහත වගුවේ සංයුක්ත ප්‍රාථමික යතුරක් ඇත
C) ඉහත වගුව දෙවන ප්‍රමතකරණ අවස්ථාවේ පවතී.

(1) A පමණක් නිවැරදිය
(2) B පමණක් නිවැරදිය
(3) C පමණක් නිවැරදිය
(4) A හා B පමණක් නිවැරදිය
(5) B හා C පමණක් නිවැරදිය

අපි මේ වගුව විශ්ලේෂණය කරමු:

Table: Student_Course_Enrollment

StudentID	CourseID	Instructor	InstructorPhone	Grade
101	CS101	Dr. Smith	555-1234	A
101	MATH201	Dr. Jones	555-5678	B
102	CS101	Dr. Smith	555-1234	A
103	MATH201	Dr. Jones	555-5678	C

විශ්ලේෂණය:

A) "වගුව පළමු ප්‍රමතකරණ (1NF) අවස්ථාවේ පවතී"

- 1NF → Atomic values, duplicate rows නැත.
- මේ වගුවේ සියලු column අගය atomic → **A නිවැරදිය.**

B) "සංයුක්ත ප්‍රාථමික යතුරක් ඇත"

- Primary key ලෙස `(StudentID, CourseID)` යොදා ගත හැක (සෑම row එකක්ම unique).
- **B නිවැරදිය.**

C) "දෙවන ප්‍රමතකරණ (2NF) අවස්ථාවේ පවතී"

- 2NF → 1NF satisfy වීම සහ partial dependency නැත.
- InstructorPhone, Instructor → CourseID මත පමණක් partial dependent (StudentID + CourseID composite key එකක් විදියට consider කරනවා නම්)
- InstructorPhone අගය repeated → **2NF නොවේ**
- **C වැරදිය.**

නිගමනය:

✓ **නිවැරදි පිළිතුර:**

(4) A හා B පමණි

39. ප්‍රමතකරණය හා සම්බන්ධ පහත වගන්ති සලකන්න

A - තෙවන ප්‍රමතකරණය වූ වගුවක් දෙවන ප්‍රමතකරණ අවස්ථාව නිතැතින් තෘප්ත කරයි
B - ප්‍රමතකරණය නොවූ වගුවක අනුපිටපත්වීම සිදුවීම අනිවාර්යයෙන් සිදු වේ
C - ප්‍රමතකරණය වූ වගුවක සංයුක්ත ප්‍රාථමික යතුරක් අඩංගු විය නොහැක.

ඉහත වගන්ති අතුරින් නිවැරදි වන්නේ

(1) A පමණි
(2) B පමණි
(3) C පමණි
(4) A හා B පමණි
(5) A හා C පමණි

අපි statements විශ්ලේෂණය කරමු:

A. "තෙවන ප්‍රමතකරණය වූ වගුවක් දෙවන ප්‍රමතකරණ අවස්ථාව නිතැතින් තෘප්ත කරයි" ✓

- Third Normal Form (3NF) → 2NF satisfy කරන අතර, non-prime attributes transitively dependent නොවිය යුතුයි.

B. "ප්‍රමතකරණය නොවූ වගුවක අනුපිටපත්වීම සිදුවීම අනිවාර්යයෙන් සිදු වේ" ✕

- Normalization කිරීම optional, compulsory නැත.

C. "ප්‍රමතකරණය වූ වගුවක සංයුක්ත ප්‍රාථමික යතුරක් අඩංගු විය නොහැක" ✕

- 3NF වගුවක් composite primary key අඩංගු විය හැක, restrictions non-prime attributes සඳහා පමණයි.

නිගමනය:

✓ **නිවැරදි පිළිතුර:**

(1) A පමණි

40 සිට 42 දක්වා සියලු ප්‍රශ්න පහත SQL කේතය මත පදනම් වේ.

CREATE TABLE Student(EntNo int(5), Grade char(5), Name varchar(15), ID_No varchar(12) NOT NULL UNIQUE, PRIMARY KEY(EntNo), FOREIGN KEY (Grade) REFERENCES Grade(Grade));

40. වගුවක් නිර්මානය කිරීම සඳහා සිසුවකු විසින් ලියන ලද SQL වගන්තියක් ඉහත දැක්වේ.

ඉහත වගන්තිය හා සම්බන්ධ ව වැරදි ප්‍රකාශය වන්නේ,

(1) ඉහත නිර්මාණය වූ වගුවේ නම student වේ.
(2) වගුවේ EntNo හිස්ව තැබිය නොහැක.
(3) වගුවේ IDNo තුල එකම දත්තය නැවත නො යෙදේ.
(4) වගුවේ Grade උපලැකිය හිස්ව තැබිය නොහැක.
(5) මෙම වගුවේ IDNo විකල්ප යතුරක් ලෙස හඳුනාගත හැකි ය.

SQL වගුව හා සම්බන්ධ වැරදි ප්‍රකාශය

මෙහිදී බලමු:

1. **Student** යන නම නිවැරදි → **සත්‍යය**
2. **EntNo හිස්ව තැබිය නොහැක** – නියමිත primary key එකක් වන අතර primary key වල NULL විය නොහැක → **සත්‍යය**
3. **IDNo එකම දත්ත නැවත යෙදිය නොහැක** – UNIQUE constraint නිසා → **සත්‍යය**
4. **Grade හිස්ව තැබිය නොහැක** – FOREIGN KEY එකක් පමණක්, NOT NULL කියලා සඳහන් නැත → වැරදිය
5. **IDNo විකල්ප යතුරක්** – UNIQUE constraint ඇත, PRIMARY/FOREIGN key ලෙස නොවේ, "විශේෂ හැඳුනුම්" ලෙස හඳුනාගත හැක → **සත්‍යය**

✅ **වැරදි ප්‍රකාශය:** (4)

41. ඉහත වගන්තිය ට අදාලව නිවැරදි සම්බන්ධක පරිපාටිය වන්නේ,

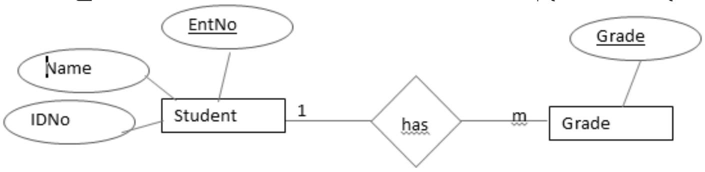
(1) Student (<u>EntNo</u>, Name, IDNo)
(2) Student (<u>EntNo</u>, Name, <u>IDNo</u>)
(3) Student (<u>EntNo</u>, Name, IDNo, <u>Grade</u>)
(4) Student (<u>EntNo</u>, Name, IDNo, Grade)
(5) Student (<u>EntNo</u>, Name, <u>IDNo</u>, Grade)

✅ **නිවැරදි පිළිතුර:**

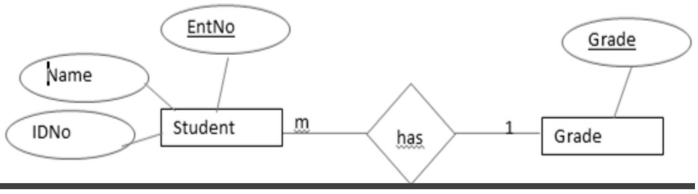
(1) 3 (Primary Key should be underlined but foreign or alternate key should not be underlined)

42. එක් ශ්‍රේණියක සිසුන් රැසක් සිටී නම් ඉහත වගුව වගු වලට අදාල නිවැරදි ER රූප සටහන කුමක්ද?

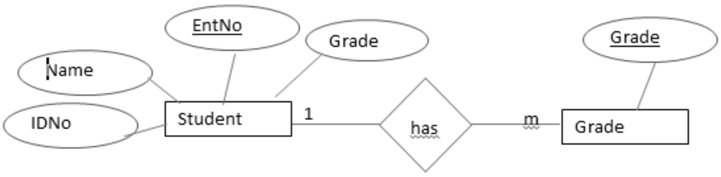
(1) .



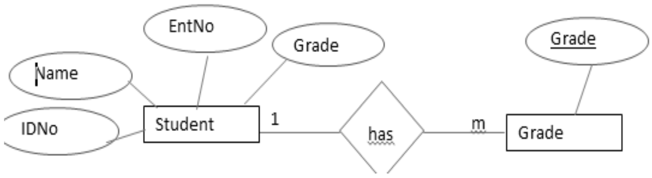
(2) .



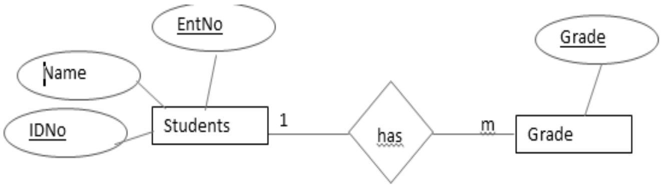
(3) .



(4) .



(5) .



එක් ශ්‍රේණියක සිසුන් **රැසක්** සිටි
එක් සිසුවකුට **එක්** ශ්‍රේණියක් තිබේ

43. සිසුන් පිළිබඳව හා ඔවුන්ගේ ක්‍රීඩා නිවාසය පිළිබඳව දත්ත පවත්වා ගන්නා වගු 2ක් ඇත. ඒවා එකිනෙක හා සම්බන්ධව පවතී. මෙහි නිවාසයක් පිළිබඳ රෙකෝඩයක් මකා දැමීම සඳහා කේතයක් ධාවනය කලද එය මකා දැමීමට ඉඩ නොදීය. එයට හේතු වූ යේ කුමන සම්බාධකයද (Constraint) ?

(1) අභිශුන්‍ය නොවන සම්බාධක (Not null)
(2) ප්‍රාථමික යතුරු සම්බාධකය (Primary Key)
(3) අනන්‍ය සම්බාධක (Unique)
(4) වගු පරීක්ෂක සම්බාධකය (Table Check)
(5) ආගන්තුක යතුරු සම්බාධක (Foreign Key)

මෙකේ සරල explanation එක 👇

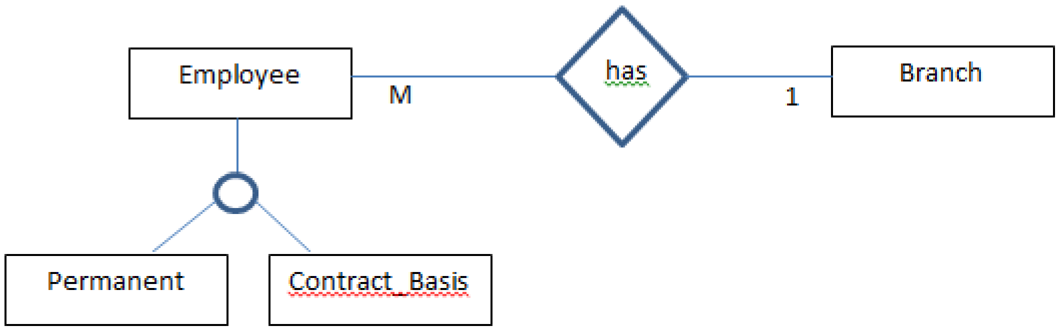
- **Student** table එකේ record එකක් "House" (නිවාස) table එකට foreign key එකක් use කරලා connected.
- නිවාස (House) table එකෙන් record එකක් delete කරන්න ගියේ නම්, එය **still referenced by students** (අදාළ සිසුන්ට ඒ නිවාස assign වෙලා තියෙනවා).
- ඒක නිසා database එක **delete** එක **prevent** කරන්නේ referential integrity වලට.

☞ ඒක හේතු කරන්නේ **Foreign Key Constraint**. ✅

අපේ පිලිතුර:

(5) ආගන්තුක යතුරු සම්බාධක (Foreign Key) ✓

44. පහත දැක්වෙන EER රූප සටහන සලකන්න



ඉහත රූප සටහන හා සම්බන්ධ වැරදි වගන්තිය වන්නේ

(1) මෙහි සුපිරි පන්තියක් (Super Class) හා උප පන්තියක් (Sub Class) අඩංගු වේ
(2) මෙම EER සටහන සම්බන්ධක 4ක් ඇසුරින් ගොඩනැගිය හැකිය
(3) මෙම EER සටහන සම්බන්ධක 3ක් ඇසුරින් ගොඩනැගිය හැකිය
(4) මෙම ආයතනයේ Permanent සහ Contract_Basis ලෙස සේවක වර්ග 2ක් සිටී
(5) සෑම සේවකයෙකුටම අවම වශයෙන් ශාඛා 1ක් හෝ කිහිපයක් හිමි වේ

Diagram එකේ විස්තර:

- **Employee** → Superclass
- **Permanent** සහ **Contract_Basis** → Subclass (specialization)
- Relationship: **Employee "has" Branch**
  - Branch : Employee = **1 : M** (එනම් branch එකකට බොහෝ **Employees** හිමි වෙනවා, නමුත් employee එකට **branch එකක්** විතරයි තියෙන්නේ).

දැන් Statements check කරමු:

1. ✅ **Super Class / Sub Class තියෙනවා** → හරි (Employee → superclass, Permanent/Contract Basis → subclasses).
2. ✅ **4 relations** → හරි.
   - Employee(EntNo, Name, …, BranchID)
   - Permanent(EntNo, …)
   - Contract_Basis(EntNo, …)
   - Branch(BranchID, …)

3. ✅ **3 relations** → හරි.
   - Permanent(EntNo, …)
   - Contract_Basis(EntNo, …)
   - Branch(BranchID, …)

4. ✅ **Employees 2 types** (Permanent, Contract_Basis) → හරි.
5. ❌ වැරදි → Diagram එකේ "Employee —M : 1— Branch" කියනවා. එනම් **employee එකකට එක branch එකක් විතරයි** (not "අවම වශයෙන් branch එකක් හෝ කිහිපයක්").

නිවැරදි පිළිතුර (වැරදි වගන්තිය):

☞ **(5) සෑම සේවකයෙකුටම අවම වශයෙන් ශාඛා 1ක් හෝ කිහිපයක් හිමි වේ ✕**

45. DML (දත්ත කළමණාකරණ භාෂාව) ට පමණක් අයත් වන විධාන වන්නේ
    (1) SELECT, UPDATE, INSERT, DELETE
    (2) CREATE, UPDATE, INSERT, ALTER
    (3) ALTER, UPDATE, SELECT, DELETE
    (4) DELETE, DROP, SELECT, UPDATE
    (5) SELECT, INSERT, DROP, ALTER

SQL වලදී

- **DML (Data Manipulation Language)** → `SELECT`, `INSERT`, `UPDATE`, `DELETE`
- **DDL (Data Definition Language)** → `CREATE`, `ALTER`, `DROP`, `TRUNCATE`

දැන් options බලමු:

1. **SELECT, UPDATE, INSERT, DELETE** ✅ (ඇත්තේම DML එකේ විධාන)
2. CREATE (DDL), ALTER (DDL) → ✕
3. ALTER (DDL) තියෙනවා → ✕
4. DROP (DDL) තියෙනවා → ✕
5. DROP, ALTER (DDL) තියෙනවා → ✕

☞ නිවැරදි පිළිතුර:
(1) SELECT, UPDATE, INSERT, DELETE ✅

46. IOT උපාංගයක් ලෙස සැලකිය නොහැක්කේ කුමක්ද?
    (1) දුරස්ථව දැල්වීමට හෝ නිවා දැමීමට හැකි බල්බයක්
    (2) අන්තර්ජාලය හරහා පාලනය කල හැකි නිෂ්පාදන ක්‍රියාවලියක්
    (3) වෙබ් අඩවිය හරහා පාලනය කල හැකි වතුර මෝටරයක්
    (4) ජංගම දුරකථනය හරහා ලොව යම් තැනක සිට නැරඹීමට හැකි කැමරාවක්
    (5) අන්තර්ජාලය හරහා අගුල් දැමිය හැකි කර්මාන්ත ශාලාවක ගේට්ටුවක්

IoT (Internet of Things) යනු → Physical devices (sensors, actuators, machines) that connect to the internet and can send/receive data.

දැන් options බලමු:

1. **X bulb** → Not IoT device (පාලනය අන්තර්ජාලය හරහා නොවේ)
2. ✅ **Production process** → එය IoT **device** එකක්, එය ක්‍රියාවලියක් (process). (IoT හරහා පාලනය කළ හැකි).
3. ✅ **Water pump** (remote via web) → IoT device

4. ✅ **CCTV / IP Camera** → IoT device
5. ✅ **Smart gate lock** → IoT device

☞ නිවැරදි පිළිතුර:
(2) අන්තර්ජාලය හරහා පාලනය කල හැකි නිෂ්පාදන ක්‍රියාවලියක් ⊘

47. ආර්ඩියුනෝ පුවරුවක අංකිත තුඩුවක් භාවිත කල හැක්කේ
    (1) උෂ්ණත්ව සංවේදකයක් භාවිතයෙන් උෂ්ණත්වය ආදානය කිරීමට
    (2) දීප්තිය වෙනස් වන බල්බයක් වෙනුවෙන්
    (3) LDR සංවේදකයක් භාවිතයෙන් පරිසර අලෝකය වෙනස් වීම හඳුනාගැනීම සඳහා
    (4) PIR සංවේදකයක් චලනයක් හඳුනා ගැනීම සඳහා
    (5) විචල්‍ය ප්‍රතිරෝධකයක් ඇසුරින් ආදානයක් ලබා දීම

හරි 🔌 Arduino ගැන කතා කරනකොට **අංකිත තුඩු (Digital pins)** → input/output signals (HIGH/LOW = 1/0) විතරයි handle කරන්න පුළුවන්.

📌 දැන් options බලමු:

1. **Temperature sensor** → (උදා: LM35) → analog output දෙනවා → **Analog pin** යොදාගන්න ඕන. ❌
2. **Dimmable bulb (varying brightness)** → Brightness vary කරන්න PWM (analog output simulation) ඕන → analogWrite() → **analog pin**. ❌
3. **LDR sensor** → analog voltage change කරනවා → **Analog pin** යොදාගන්න ඕන. ❌
4. **PIR motion sensor** → normally digital output (HIGH/LOW) දෙනවා → **Digital pin** යොදාගන්න පුළුවන් ✅
5. **Variable resistor (potentiometer)** → analog voltage divider output → **Analog pin** යොදාගන්න ඕන. ❌

☞ නිවැරදි පිළිතුර:
(4) PIR සංවේදකයක් චලනයක් හඳුනා ගැනීම සඳහා ✅

48. **ස්වභාවයෙන් ආවේණික ගණනය (Nature-Inspired Computing)** යනු පරිගණක විද්‍යාවේ එවැනි ක්‍රම හා තාක්ෂණයන් වන අතර ඒවා ස්වභාවික පද්ධති හා ක්‍රියාකාරකම් වලින් ආදර්ශ ගනිමින් නිර්මාණය වේ. පහත සඳහන් කරුණු අතර ස්වභාවයෙන් ආවේණික ගණනයට අදාළ නොවන සිද්ධිය කුමක්ද?
    (1) අන්තකූට විද්‍යාව (Swarm Intelligence) යොදාගෙන ගැටලු විසඳීම
    (2) ජීව විද්‍යානුකූල සංස්කරණ (Genetic Algorithms) මගින් හොඳම විසඳුම් සොයා ගැනීම
    (3) වෘක්ෂ මොළය (Tree-based Neural Networks) මගින් දත්ත සැකසුම
    (4) ස්වයංක්‍රීය ගමන් මාර්ග සැලසුම සඳහා දියුණු වූ යාන්ත්‍ර ඉගෙනුම් (Machine Learning) ක්‍රම
    (5) ස්වභාවික පරිසරයේ ඇති ශාඛා පද්ධති වල සංකේතනය

All Marks

49. **අන්තර්ජාල මුදල් ගෙවීම් සඳහා භාවිතා කරන ගෙවීම් ද්වාර (Payment Gateway) සම්බන්ධව නිවැරදි ප්‍රකාශය කුමක්ද?**

(1) ගෙවීම් ද්වාරය යනු පාරිභෝගිකයින්ගෙන් ගෙවීම් විස්තර රහසිගතව ආරක්ෂා කරමින් වෙළඳසැලට ලබාදෙන මෘදුකාංගයක් ය.
(2) ගෙවීම් ද්වාරය පමණක් මුදල් ගබඩා කිරීම සඳහා පමණක් භාවිතා වේ.
(3) ගෙවීම් ද්වාරයකට සම්බන්ධ වීම සඳහා පාරිභෝගිකයින්ට අන්තර්ජාල සම්බන්ධතාවයක් අවශ්‍ය නැත.
(4) ගෙවීම් ද්වාරය මගින් පරිශීලකයින්ට බහුගුණිත මුදල් ගෙවීම් ක්‍රම ලබා දීමේ හැකියාවක් නොමැත.
(5) ගෙවීම් ද්වාරය නිසා පරිශීලක දත්ත ආරක්ෂා කිරීම අසම්පූර්ණ වේ.

Payment Gateway යනු → Online payments කිරීමේදී පාරිභෝගිකයාගේ sensitive card/bank/payment දත්ත **encrypt** කරලා secureව merchant (වෙළඳසැල/සංවිධානය) වෙත යවන මධ්‍යස්ථ මෘදුකාංග/සේවාවකි.

දැන් Options විමසමු:

1. ✅ **හරි** → "ගෙවීම් ද්වාරය යනු පාරිභෝගිකයින්ගෙන් ගෙවීම් විස්තර රහසිගතව ආරක්ෂා කරමින් වෙළඳසැලට ලබාදෙන මෘදුකාංගයක් ය."
2. ❌ පමණක් මුදල් **ගබඩා කිරීම** සඳහා නෙවෙයි → මුදල් transfer/authorize/validate කරන්නයි.
3. ❌ **Internet connection** අවශ්‍යයි → නැත්නම් payment gateway use කරන්න බැහැ.
4. ❌ Payment gateways එකේ **multiple payment methods** (credit card, debit card, mobile wallets, etc.) support කරනවා.
5. ❌ එවටම හේතුව → දත්ත ආරක්ෂාව **සම්පූර්ණවම implement** කරනවා (Encryption, SSL, PCI-DSS compliance).

☞ නිවැරදි පිළිතුර:
(1) ගෙවීම් ද්වාරය යනු පාරිභෝගිකයින්ගෙන් ගෙවීම් විස්තර රහසිගතව ආරක්ෂා කරමින් වෙළඳසැලට ලබාදෙන මෘදුකාංගයක් ය. ✅

50. **නියෝජිත මෘදුකාංගයක් පරිශීලකගේ විද්‍යුත් තැපැල් පණිවිඩ ස්වයංක්‍රීයව පරීක්ෂා කර අවශ්‍ය ක්‍රියාමාර්ග ගන්නා පරිදි ක්‍රියා කරන බව සලකා බලමු. එවැනි මෘදුකාංගයක ප්‍රධාන ලක්ෂණ මොනවද?**

A) පරිශීලකගේ මැදිහත් වීමක් අවශ්‍ය නොවී ස්වයංක්‍රීයව ක්‍රියා කිරීම
B) පරිසරයෙහි සිදුවන වෙනස්කම් වලට වහාම ප්‍රතිචාර දැක්වීම
C) වෙනත් මෘදුකාංග හෝ සේවා සමඟ සන්නිවේදනය කරගනිමින් ක්‍රියා කිරීම
D) පරිශීලක ඉල්ලීම් අනුව ඉලක්කිත කාර්ය ඉටු කිරීම

ඉහත ප්‍රකාශ අතර, නියෝජිත මෘදුකාංගයක මූලික ලක්ෂණ ලෙස සැලකිය යුතු එකතුව කුමක්ද?

(1) A හා B පමණි
(2) A, B හා C පමණි
(3) B හා D පමණි
(4) A, C හා D පමණි
(5) A, B, C හා D සියල්ලම

Agent Software (නියෝජිත මෘදුකාංග) වල මූලික ලක්ෂණ:

- **A)** Automation → පරිශීලක මැදිහත් වීමක් නැතුව ක්‍රියා කරනවා ✅
- **B)** Reactivity → පරිසරයෙහි වෙනස්කම් වලට ප්‍රතිචාර දක්වනවා ✅
- **C)** Social Ability → වෙනත් මෘදුකාංග/සේවා/ඇජන්ට්ස් සමඟ සන්නිවේදනය කරගන්නවා ✅
- **D)** Goal-oriented behavior → පරිශීලක ඉල්ලීම් අනුව ඉලක්කිත කාර්ය ඉටු කරනවා ✅

☞ ඒකෙන් හොඳම එකතුව:
(5) A, B, C හා D සියල්ලම ✅