12 ශ්‍රේණිය A/L ICT · පාඩම 3

3.3 ද්වීමය අංක ගණිතය සහ බිටු-අනුව තර්කය: එකතු, අඩු, AND, OR, XOR, NOT

විෂය නිර්දේශය 3.3. CPU තුළ ඇති ALU එක සියලු වැඩ බිටු මගින් කරයි. මෙම කොටසින් **ද්වීමය එකතු කිරීම** (සංඛ්‍යා කිහිපයක් එකතු කරන විට පවා රැගෙන යාම් සමඟ), **ද්වීමය අඩු කිරීම** (ණයට ගැනීම සමඟ, සහ 2 හි අනුපූරකය සමඟ), **පිටාර ගැලීම (overflow)** යනු කුමක්ද, සහ **බිටු-අනුව තාර්කික මෙහෙයුම් NOT, AND, OR සහ XOR** ඉගෙන ගනී.

1. ද්වීමය එකතු කිරීමේ නීති

ද්වීමය එකතු කිරීම **දශම එකතු කිරීම මෙන්ම** ක්‍රියා කරයි, **දකුණේ** සිට තීරුවෙන් තීරුව. එකම වෙනස: ද්වීමයේ **2 ලියන්නේ 10 ලෙස**, එබැවින් 0 ලියා ඊළඟ තීරුවට **1 රැගෙන යන්න (carry)**.

Bits added එකතු කරන බිටු	Sum bit එකතු බිටුව	Carry රැගෙන යාම
0 + 0 0 + 0	**0** 0	**0** 0
0 + 1 0 + 1	**1** 1	**0** 0
1 + 1 1 + 1	**0** 0	**1** 1
1 + 1 + 1 1 + 1 + 1	**1** 1	**1** 1

ද්වීමය එකතු කිරීමේ නීති හතර

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 1 = 10 → **0** ලියන්න, **1** රැගෙන යන්න
- 1 + 1 + 1 = 11 → **1** ලියන්න, **1** රැගෙන යන්න

🧮 **විසඳූ උදාහරණය: 1011_2 + 0110_2 එකතු කරන්න.**
දකුණු තීරුව: 1 + 0 = 1. ඊළඟ: 1 + 1 = 10, 0 ලියා 1 රැගෙන යන්න. ඊළඟ: 0 + 1 + 1 = 10, 0 ලියා 1 රැගෙන යන්න. ඊළඟ: 1 + 0 + 1 = 10, 0 ලියා 1 රැගෙන යන්න. අවසාන carry එක පහළට. පිළිතුර **10001_2**. පරීක්ෂාව: 11 + 6 = 17 ✔

	2^4	2^3	2^2	2^1	2^0
රැගෙන යාම (carry)	1	1	1		
	0	1	0	1	1
	0	0	1	1	0
එකතුව	**1**	**0**	**0**	**0**	**1**

🧮 **විසඳූ උදාහරණය: සංඛ්‍යා තුනක් එකතු කරන්න: 1011 + 1101 + 0111.**
සෑම තීරුවකම 1 තුනක් හෝ හතරක් හමු විය හැක. 1 ගණන් කරන්න (carry ද සමඟ): 3 = 11, එබැවින් **1** ලියා **1** රැගෙන යන්න. සෑම තීරුවකටම මෙසේ කළ විට $\mathbf{11111_2}$. පරීක්ෂාව: 11 + 13 + 7 = 31 ✔

	2^4	2^3	2^2	2^1	2^0
රැගෙන යාම (carry)	1	1	1	1	
	0	1	0	1	1
	0	1	1	0	1
	0	0	1	1	1
එකතුව	**1**	**1**	**1**	**1**	**1**

💡 **සංඛ්‍යා බොහොමයක්ද?** තීරුවක එකතුව 4 (ද්වීමයෙන් 100) නම්, 0 ලියා **තීරු දෙකක්** වමට 1 රැගෙන යන්න. නැත්නම් මුලින් සංඛ්‍යා දෙකක් එකතු කර, පසුව තුන්වැන්න පිළිතුරට එකතු කරන්න – විභාගයේදී මෙය වඩා ආරක්ෂිතයි.

2. ණයට ගැනීම සමඟ ද්වීමය අඩු කිරීම

- 0 − 0 = 0, 1 − 0 = 1, 1 − 1 = 0
- 0 − 1: වම් පස ඊළඟ තීරුවෙන් **1 ණයට ගන්න**. ණයට ගත් 1 මෙහි අගය **2**, එබැවින් 2 − 1 = **1**.
- 1 ණයට දුන් තීරුව දැන් **එකකින් අඩුයි**.

🧮 **විසඳූ උදාහරණය: $\mathbf{11010_2 - 01011_2}$ අඩු කරන්න.**
දකුණු තීරුව: 0 − 1, ණයට ගන්න → 2 − 1 = 1. ඊළඟ: 1 − 1 − 1 (ණය ආපසු ගෙවීම) 0 ට අඩු නිසා නැවත ණයට ගන්න: 2 + 1 − 1 − 1 = 1. එලෙසම ඉදිරියට; සෑම ණයක්ම ගෙවූ තැන වගුවේ ඇත. පිළිතුර $\mathbf{01111_2}$. පරීක්ෂාව: 26 − 11 = 15 ✔

	2^4	2^3	2^2	2^1	2^0
ණයට ගත් (borrow)	−1	−1	−1	−1	
	1	1	0	1	0

–	0	1	0	1	1
වෙනස	**0**	**1**	**1**	**1**	**1**

🧮 **විසඳූ උදාහරණය: $10000_2 - 00001_2$ අඩු කරන්න (දිගු ණය දාමයක්).**
දකුණු තීරුවට ණයක් අවශ්‍ය නමුත් අසල්වැසි තීරු 0 නිසා ණය වම්කෙළවරේ 1 දක්වා යයි. එම 1, 0 වන අතර මාර්ගයේ සෑම 0 ක්ම 1 වේ. පිළිතුර $\mathbf{01111_2}$ (16 − 1 = 15).

	2^4	2^3	2^2	2^1	2^0
ණයට ගත් (borrow)	−1	−1	−1	−1	
	1	0	0	0	0
–	0	0	0	0	1
වෙනස	**0**	**1**	**1**	**1**	**1**

3. 2 හි අනුපූරකය සමඟ අඩු කිරීම

CPU එකකට වෙනම අඩු කිරීමේ පරිපථයක් අවශ්‍ය නැත. **A − B** සෙවීමට එය එකතු කරයි: **A + (B හි 2 හි අනුපූරකය)**. වම් කෙළවරින් පිටතට යන carry එක (බිටු 8 සංඛ්‍යා සඳහා 9 වැනි බිටුව) **ඉවත දමයි**.

🧮 **විසඳූ උදාහරණය: බිටු 8 2 හි අනුපූරකය භාවිතයෙන් 45 − 20 සොයන්න.**
+45 = 0010 1101. −20 = 1110 1100 (00010100 පෙරළා 1 එකතු කරන්න). ඒවා එකතු කරන්න: 100011001. 9 වැනි බිටුව ඉවත දමන්න → **0001 1001** = 25 ✔

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
රැගෙන යාම (carry)	1	1	1		1	1			
	0	0	0	1	0	1	1	0	1
	0	1	1	1	0	1	1	0	0
එකතුව	**1**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**1**

4. පිටාර ගැලීම (Overflow)

බිටු 8 2 හි අනුපූරක සංඛ්‍යාවකට −128 සිට +127 පමණක් රඳවා ගත හැක. සත්‍ය පිළිතුර **මෙම පරාසයෙන් පිටත** නම්, ප්‍රතිඵලය වැරදියි. මෙය **පිටාර ගැලීම** ලෙස හැඳින්වේ. ලකුණ පරීක්ෂා කිරීමෙන් එය හමු වේ: **ධන දෙකක් එකතු කළ විට සෘණ** ලැබීම, හෝ **සෘණ දෙකක් එකතු කළ විට ධන** ලැබීම.

🧮 **විසඳූ උදාහරණය: බිටු 8 2 හි අනුපූරකයෙන් 100 + 50 එකතු කරන්න. වැරදෙන්නේ කුමක්ද?**
0110 0100 + 0011 0010 = **1001 0110**. MSB 1 නිසා පරිගණකය **−106** ලෙස කියවයි, 150 නොවේ. ධන දෙකක් සෘණයක් දුන්නේය: **පිටාර ගැලීම**, 150 > 127 නිසා.

5. බිටු-අනුව තාර්කික මෙහෙයුම්

බිටු-අනුව (bitwise) මෙහෙයුමක් **සෑම බිටු යුගලයක් මතම වෙන වෙනම**, තීරුවෙන් තීරුව ක්‍රියා කරයි. **carry එකක් නැත**.

A	B	A AND B	A OR B	A XOR B	NOT A
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

AND
බිටු **දෙකම** 1 නම් පමණක් 1.

OR
අවම වශයෙන් එක් බිටුවක් 1 නම් 1.

XOR
බිටු **වෙනස්** නම් 1.

NOT
සෑම බිටුවක්ම **පෙරළයි** (එක් ආදානයක්).

🧮 **විසඳූ උදාහරණය: A = 11001010 සහ B = 10101100. A AND B, A OR B, A XOR B සහ NOT A සොයන්න.**
සත්‍යතා වගුව භාවිතයෙන් තීරුවෙන් තීරුව කරන්න: AND = **10001000**, OR = **11101110**, XOR = **01100110**, NOT A = **00110101**.

	b7	b6	b5	b4	b3	b2	b1	b0
A	1	1	0	0	1	0	1	0
B	1	0	1	0	1	1	0	0
A AND B	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
A OR B	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**0**
A XOR B	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
NOT A	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**



A සහ B මත බිටු-අනුව මෙහෙයුම්

6. බිටු-අනුව මෙහෙයුම් භාවිතය (Masks)

Mask එකක් යනු ඇතැම් බිටු තෝරා ගැනීමට හෝ වෙනස් කිරීමට භාවිත කරන බිටු රටාවකි. Mask එකක් සමඟ **AND** බිටු **පරීක්ෂා කරයි හෝ හිස් කරයි**, **OR** බිටු 1 ට **සකසයි**, සහ **XOR** බිටු **පෙරළයි**.

🧱 **විසඳූ උදාහරණය: A = 11001010 හි 3 වැනි බිටුව 1 ද? Mask එකක් භාවිත කරන්න.**
11001010 AND 00001000 = **00001000**. පිළිතුර 0 නොවන නිසා 3 වැනි බිටුව **1 කි**.

🧱 **විසඳූ උදාහරණය: එක් බිටු-අනුව මෙහෙයුමකින් ASCII “A” (01000001) අකුර “a” බවට හරවන්න.**
සිම්පල් අකුරු 32 කින් වැඩියි, 32 = 00100000. 01000001 OR 00100000 = **01100001** = 97 = “a”.

2026 නව **බිටු-අනුව AND මගින් ජාලය සොයයි.** IP ලිපිනය 192.168.1.130 සහ subnet mask 255.255.255.0 ඇති පරිගණකයක් ජාල ලිපිනය **192.168.1.0** ලබා ගැනීමට 192.168.1.130 AND 255.255.255.0 කරයි. ඔබේ නිවසේ router එක සෑම පැකට්ටුවක් සඳහාම මෙය කරයි.

🚀 විෂය නිර්දේශයෙන් ඔබ්බට (2026 යාවත්කාලීනය) *විභාගයට අවශ්‍ය නැත – දැනුම සඳහා කියවන්න.* **විෂය නිර්දේශයෙන් ඔබ්බට – සෑම CPU එකකම 2 හි අනුපූරකය.** සෑම නූතන සකසනයක්ම (Intel, AMD, Apple M-series, දුරකථනවල ARM චිප) ලකුණු සහිත නිඛිල 2 හි අනුපූරකයෙන් ගබඩා කරයි. C++20 සහ C23 භාෂා ප්‍රමිති දැන් එය **අනිවාර්ය** කරයි. ALU එක ලකුණු රහිත පිටාර ගැලීම සඳහා **carry flag** එකක් සහ ලකුණු සහිත පිටාර ගැලීම සඳහා **overflow flag** එකක් සකසයි. AES වැනි සංකේතනයේ හදවතද XOR ය: දත්ත XOR යතුර දත්ත සඟවයි, එම යතුරෙන්ම නැවත XOR කළ විට දත්ත ආපසු ලැබේ.

ප්‍රධාන පද

රැගෙන යාම (Carry) – තීරුවක එකතුව 2 හෝ වැඩි වූ විට ඊළඟ තීරුවට ගෙන යන 1.	**ණයට ගැනීම (Borrow)** – 0 − 1 අවශ්‍ය වූ විට වම් පස ඊළඟ තීරුවෙන් 1 ගැනීම.
පිටාර ගැලීම (Overflow) – සත්‍ය ප්‍රතිඵලය බිටු ගණනට වඩා විශාල හෝ කුඩා වීම.	**එකතු බිටුව** – එකතු කිරීමෙන් පසු තීරුවක ලියන බිටුව.
බිටු-අනුව මෙහෙයුම – carry නොමැතිව එක් එක් බිටු යුගලය මත වෙන වෙනම කරන මෙහෙයුම.	**AND** – බිටු දෙකම 1 වූ විට පමණක් 1 දෙයි.
OR – අවම වශයෙන් එක් බිටුවක් 1 වූ විට 1 දෙයි.	**XOR** – බිටු දෙක වෙනස් වූ විට 1 දෙයි.
NOT – සෑම බිටුවක්ම පෙරළයි: 0, 1 වේ සහ 1, 0 වේ.	**Mask (ආවරණය)** – තෝරාගත් බිටු පරීක්ෂා කිරීමට හෝ වෙනස් කිරීමට AND, OR හෝ XOR සමඟ භාවිත කරන බිටු රටාවක්.
ALU – අංක ගණිතය සහ තර්කය කරන CPU කොටස.	**සත්‍යතා වගුව** – සෑම ආදාන සංයෝජනයක් සඳහාම ප්‍රතිදානය පෙන්වන වගුවක්.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. ද්වීමය එකතු කිරීමේ නීති හතර සඳහන් කරන්න.

> **පිළිතුර:** 0 + 0 = 0; 0 + 1 = 1; 1 + 1 = 0 carry 1; 1 + 1 + 1 = 1 carry 1.

2. 1011 + 1101 + 0111 එකතු කරන්න.

> **පිළිතුර:** 11111 (11 + 13 + 7 = 31).

3. 11010 − 01011 අඩු කරන්න.

> **පිළිතුර:** 01111 (26 − 11 = 15).

4. CPU එකක් එකතු කිරීමේ පරිපථයක් පමණක් භාවිතයෙන් අඩු කරන්නේ කෙසේද?

> **පිළිතුර:** අඩු කරන සංඛ්‍යාවේ 2 හි අනුපූරකය සොයා එය එකතු කරයි. වම් කෙළවරින් පිටතට යන carry එක නොසලකා හරියි. උදා: 45 − 20 = 00101101 + 11101100 = 00011001 = 25.

5. පිටාර ගැලීම යනු කුමක්ද? උදාහරණයක් දෙන්න.

> **පිළිතුර:** පිළිතුර බිටුවලට නොගැළපෙන විට පිටාර ගැලීම සිදු වේ. බිටු 8 2 හි අනුපූරකයේ 100 + 50 = 150, 127 ට වඩා වැඩි නිසා ප්‍රතිඵලය 10010110, −106 ලෙස කියවේ.

6. 11001010 XOR 10101100 සොයා XOR පෙන්වන්නේ කුමක්දැයි කියන්න.

> **පිළිතුර:** 01100110. බිටු දෙක වෙනස් සෑම තැනකම XOR 1 පෙන්වයි.

බහුවරණ ප්‍රශ්න

1. 1011 + 0110 කීයද?

A. 1101
B. 10001
C. 10101
D. 11001
E. 10011

2. ද්වීමයෙන් 1 + 1 + 1 ලබා දෙන්නේ:

A. එකතුව 0, carry 1
B. එකතුව 0, carry 0
C. එකතුව 1, carry 1
D. එකතුව 1, carry 0
E. එකතුව 3, carry නැත

3. 1011 + 1101 + 0111 කීයද?

A. 11011
B. 111111
C. 11101
D. 11111
E. 10111

4. 11010 − 01011 කීයද?

A. 01110
B. 01111
C. 01101
D. 10001
E. 10111

5. 10000 − 00001 කීයද?

A. 11111
B. 01110
C. 10001
D. 00001
E. 01111

6. ද්වීමය අඩු කිරීමේදී 0 − 1 සඳහා අවශ්‍ය වන්නේ:

A. ඊළඟ තීරුවට carry එකක්
B. කිසිවක් නැත, පිළිතුර 0
C. overflow flag එකක්
D. NOT මෙහෙයුමක්
E. ඊළඟ තීරුවෙන් ණයක්

7. A − B සෙවීමට CPU එකක් සාමාන්‍යයෙන් ගණනය කරන්නේ:

A. A + B හි 2 හි අනුපූරකය
B. A XOR B
C. A AND NOT B
D. B − A ඉන්පසු NOT
E. A + B

8. බිටු 8 2 හි අනුපූරකයේ 00101101 + 11101100 (9 වැනි බිටුව ඉවත් කළ පසු):

A. 10011001
B. 11100011
C. 00011001
D. 00011000
E. 100011001

9. බිටු 8 2 හි අනුපූරකයේ 100 + 50 එකතු කිරීම සෘණ පිළිතුරක් දෙයි. මෙය:

A. පිටාර ගැලීම
B. Mask එකක්
C. ණයක්
D. float එකක underflow
E. නිවැරදියි, 150 සෘණයි

10. 11001010 AND 10101100 = ?

A. 10001010
B. 10001000
C. 01100110
D. 11101110
E. 01110111

11. 11001010 OR 10101100 = ?

A. 11101110
B. 01100110
C. 10001000
D. 11101111
E. 11111110

12. 11001010 XOR 10101100 = ?

A. 10011001
B. 01100111
C. 10001000
D. 11101110
E. 01100110

13. NOT 11001010 = ?

A. 11001010
B. 11001011
C. 01010011
D. 00110101
E. 00110100

14. බිටු දෙකම 1 වූ විට පමණක් 1 දෙන මෙහෙයුම කුමක්ද?

A. ADD
B. OR
C. NOT
D. AND
E. XOR

15. Mask 00100000 සමඟ “A” “a” බවට හරවන මෙහෙයුම කුමක්ද?

A. NOT
B. 2’s complement
C. OR
D. AND
E. Subtraction

16. IP 192.168.1.130 AND mask 255.255.255.0 ලබා දෙන්නේ:

A. 192.168.1.0
B. 192.168.1.255
C. 0.0.0.130
D. 255.255.255.130
E. 192.168.1.130

17. “තීරුවක එකතුව 2 හෝ වැඩි වූ විට ඊළඟ තීරුවට ගෙන යන 1” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ALU
B. පිටාර ගැලීම (Overflow)
C. රැගෙන යාම (Carry)
D. XOR
E. AND

18. “0 − 1 අවශ්‍ය වූ විට වම් පස ඊළඟ තීරුවෙන් 1 ගැනීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. OR
B. NOT
C. එකතු බිටුව
D. ණයට ගැනීම (Borrow)
E. සත්‍යතා වගුව

19. “සත්‍ය ප්‍රතිඵලය බිටු ගණනට වඩා විශාල හෝ කුඩා වීම” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. Mask (ආවරණය)
B. පිටාර ගැලීම (Overflow)
C. බිටු-අනුව මෙහෙයුම
D. රැගෙන යාම (Carry)
E. XOR

20. “එකතු කිරීමෙන් පසු තීරුවක ලියන බිටුව” යන්නෙන් අදහස් වන පදය කුමක්ද?

A. ණයට ගැනීම (Borrow)
B. AND
C. NOT
D. ALU
E. එකතු බිටුව

ව්‍යුහගත ප්‍රශ්න

1. (a) carry පෙන්වමින් 1011 + 0110 එකතු කරන්න. (b) දශමයෙන් පිළිතුර පරීක්ෂා කරන්න. *(ලකුණු 3)*

> (a) තීරු තුනක carry; එකතුව 10001; (b) 11 + 6 = 17.

2. ද්වීමය සංඛ්‍යා තුන 1011, 1101, 0111 එකතු කර පිළිතුර ද්වීමය සහ දශමයෙන් දෙන්න. *(ලකුණු 3)*

> සෑම තීරුවකම: 1 ගණන + carry; එකතුව 11111; දශම 31.

3. (a) ණයට ගැනීමෙන් 11010 − 01011 අඩු කරන්න. (b) බිටු 8 2 හි අනුපූරකයෙන් 45 − 20 කරන්න. *(ලකුණු 3)*

> (a) 01111; (b) 00101101 + 11101100 = 100011001, 9 වැනි බිටුව ඉවත් කරන්න; 00011001 = 25.

4. (a) පිටාර ගැලීම යනු කුමක්ද? (b) බිටු 8 2 හි අනුපූරකයේ 100 + 50 පිටාර ගලන බව පෙන්වන්න. *(ලකුණු 4)*

> (a) බිටුවලට රඳවා ගත හැකි පරාසයෙන් පිටත ප්‍රතිඵලය; (b) 01100100 + 00110010 = 10010110; MSB 1 නිසා −106 ලෙස කියවේ; ධන දෙකක් සෘණයක් දුන්නේය.

5. A = 11001010, B = 10101100. සොයන්න (a) A AND B (b) A OR B (c) A XOR B (d) NOT A (e) mask එකක් සමඟ AND හි එක් භාවිතයක්. *(ලකුණු 5)*

> (a) 10001000; (b) 11101110; (c) 01100110; (d) 00110101; (e) බිටුවක් පරීක්ෂා කිරීම හෝ subnet mask එකකින් ජාල ලිපිනය සෙවීම.

සාරාංශය

- ද්වීමය එකතු කිරීම: 1 + 1 = 0 carry 1, 1 + 1 + 1 = 1 carry 1; දකුණේ සිට කරන්න.
- සංඛ්‍යා බොහොමයක් සඳහා, සෑම තීරුවකම 1 ගණන සහ carry ගණන් කරන්න.
- අඩු කිරීම: 0 − 1 වම් තීරුවෙන් 1 (අගය 2) ණයට ගනී.
- CPU එක 2 හි අනුපූරකය එකතු කර අමතර carry එක ඉවත් කිරීමෙන් අඩු කරයි.
- පිටාර ගැලීම: සත්‍ය පිළිතුර බිටුවල පරාසයෙන් පිටත (බිටු 8: −128 සිට +127).
- බිටු-අනුව: AND දෙකම 1, OR ඕනෑම 1, XOR වෙනස්, NOT පෙරළයි; masks බිටු පරීක්ෂා, සකස් හෝ පෙරළයි.

MCQ පිළිතුරු – 3.3

1. **B** – 11 + 6 = 17 = 10001.
2. **C** – 3 = 11.
3. **D** – 11 + 13 + 7 = 31.
4. **B** – 26 − 11 = 15.
5. **E** – 16 − 1 = 15.
6. **E** – 1 ණයට ගන්න (අගය 2).
7. **A** – අඩු කිරීම එකතු කිරීමක් වේ.
8. **C** – 45 + (−20) = 25 = 00011001.
9. **A** – 150 > 127.
10. **B** – දෙකම 1 තැන්වල පමණක් 1.
11. **A** – අවම වශයෙන් එකක් 1 තැන්වල 1.
12. **E** – බිටු වෙනස් තැන්වල 1.
13. **D** – සෑම බිටුවක්ම පෙරළන්න.
14. **D** – AND සත්‍යතා වගුව.
15. **C** – OR මඟින් 5 වැනි බිටුව 1 කරයි (+32).

16. **A** – 130 AND 0 = 0.
17. **C** – රැගෙන යාම (Carry) – තීරුවක එකතුව 2 හෝ වැඩි වූ විට ඊළඟ තීරුවට ගෙන යන 1.
18. **D** – ණයට ගැනීම (Borrow) – 0 − 1 අවශ්‍ය වූ විට වම් පස ඊළඟ තීරුවෙන් 1 ගැනීම.
19. **B** – පිටාර ගැලීම (Overflow) – සත්‍ය ප්‍රතිඵලය බිටු ගණනට වඩා විශාල හෝ කුඩා වීම.
20. **E** – එකතු බිටුව – එකතු කිරීමෙන් පසු තීරුවක ලියන බිටුව.

© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.