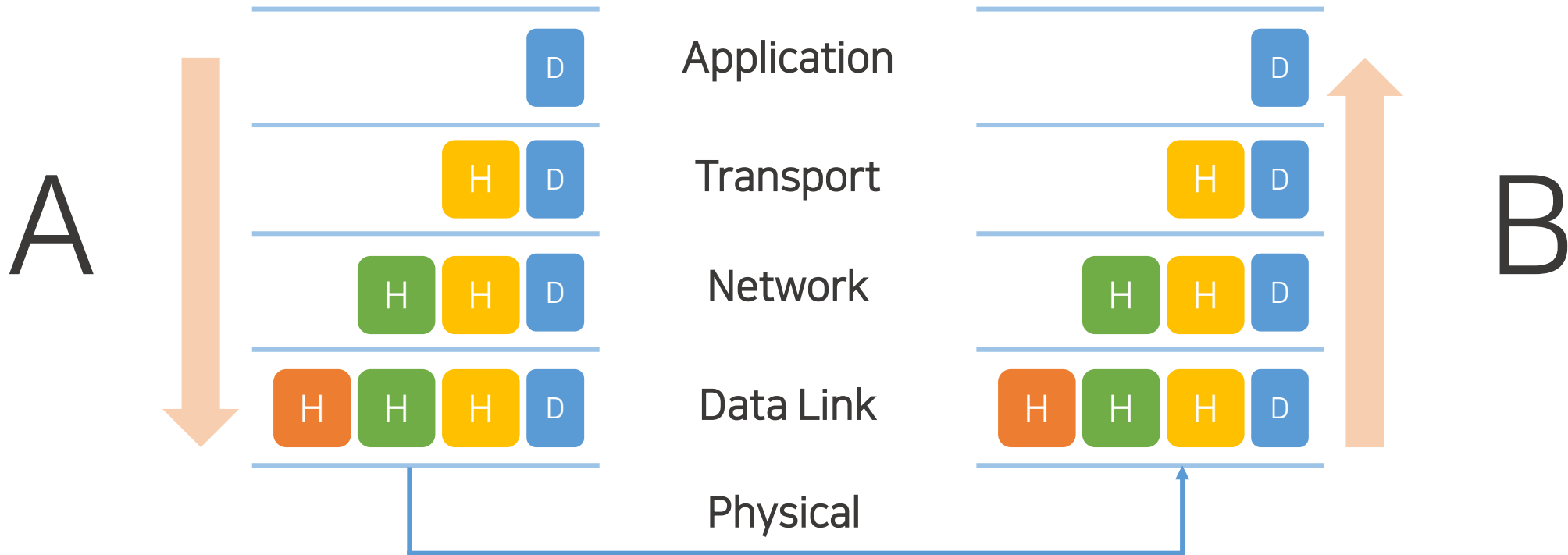


네트워크 기초

encapsulation & decapsulation

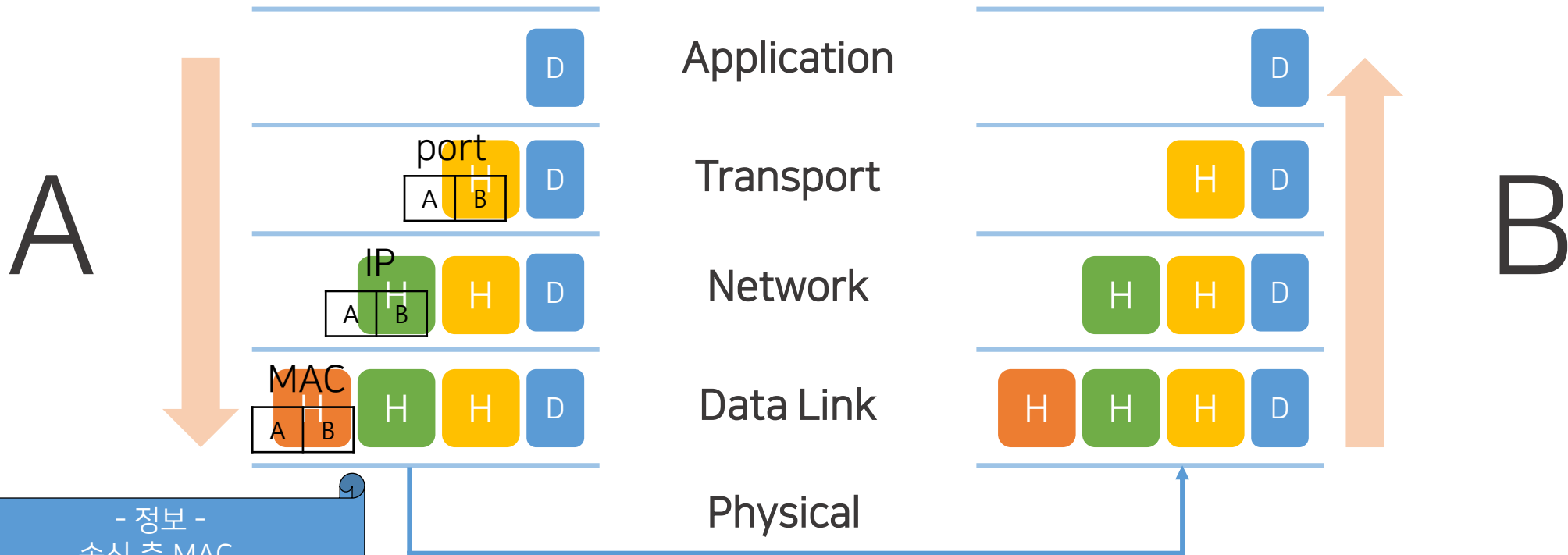
✓ 캡슐화 : 송신 데이터에 필요한 정보를 붙여서 다음 계층에 보내는 기술



D : 데이터 , H : 헤더

encapsulation & decapsulation

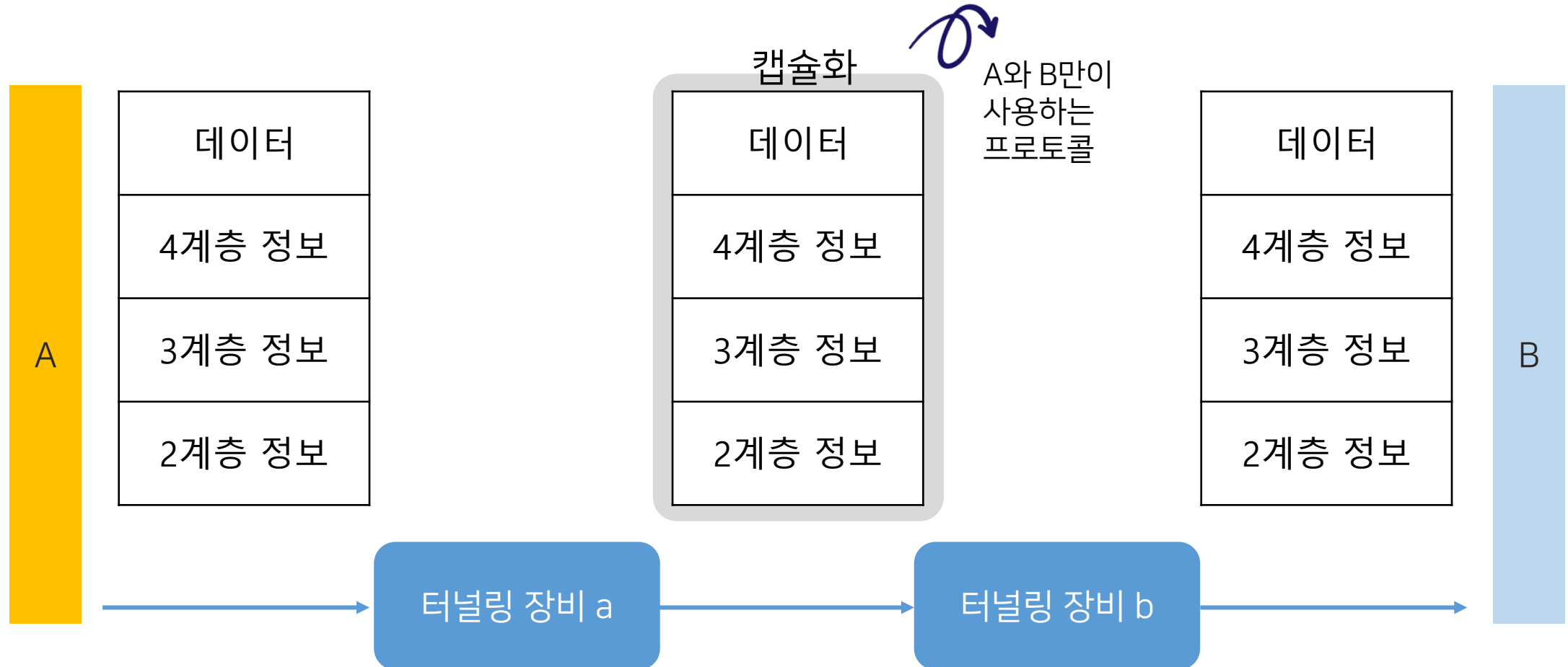
✓ 캡슐화 : 송신 데이터에 필요한 정보를 붙여서 다음 계층에 보내는 기술



D : 데이터, H : 헤더

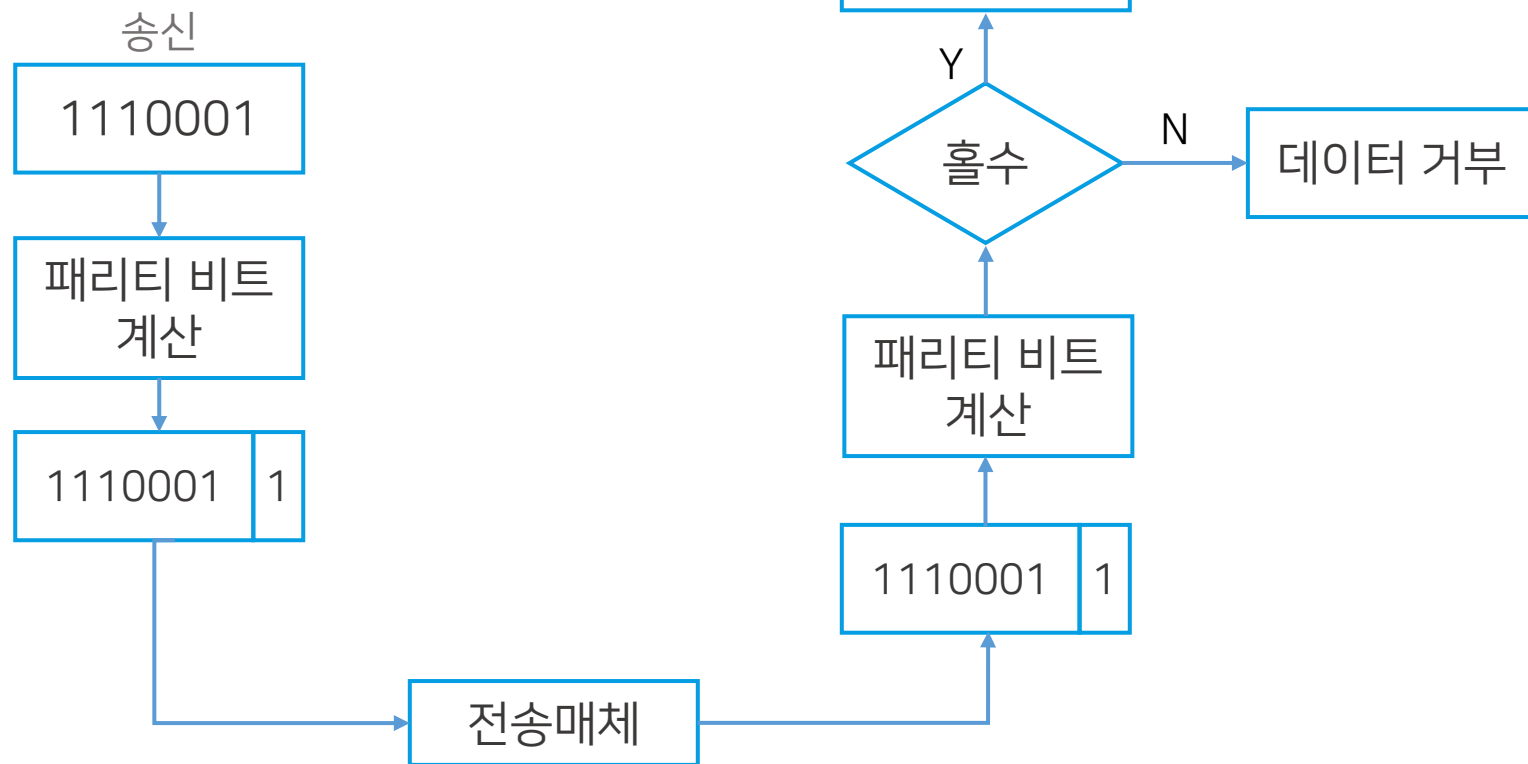
Tunneling encapsulation

- ✓ 인터넷을 사적이고 안전한 네트워크의 일부로 사용하게 하는 기술



Parity bit

- ✓ 오류를 검출하기 위해 자료 마지막에 추가한 한자리(0 or 1) 숫자
- ✓ 짝수 패리티, 홀수 패리티



Parity bit

- ✓ 오류를 검출하기 위해 자료 마지막에 추가한 한자리(0 or 1) 숫자
- ✓ 짝수 패리티, 홀수 패리티

※ 짝수 오류 발생 시 검출 못 함

1110001/1 \longrightarrow 11100**1**1/1 짝수 => 검출
 \searrow 11100**10**/1 홀수 => 검출 X

Two-Dimensional Parity bit

10011001	11100010	00100100	10000100
----------	----------	----------	----------

10011001	0
11100010	0
00100100	0
10000100	0
11011011	0

100110010	111000100	001001000	100001000	110110110
-----------	-----------	-----------	-----------	-----------

Two-Dimensional Parity bit

10011001	11100010	00100100	10000100
----------	----------	----------	----------

10011001	0
11100110	0
00100100	0
10000100	0
11011011	0

100110010	111000100	001001000	100001000	110110110
-----------	-----------	-----------	-----------	-----------

Hamming Code

- ✓ 오류를 스스로 검출해 교정이 가능한 코드
- ✓ 2bit의 오류 검출 가능하고 1bit 오류 교정 가능.
- ✓ 1, 2, 4, 8, 16 ... 2^n 번째는 오류 검출을 위한 패리티 비트
- ✓ 오류 검출 및 교정을 위한 잉여 비트가 많이 필요하다.
- ✓ n 번째 패리티 비트는 n 비트에서 시작해 n 비트 만큼을 포함하고, n 비트 씩 건너뛴 비트들을 대상으로 패리티 비트가 결정된다.

10011101 ← 0110 (원래 데이터)

Hamming Code

- ✓ n번째 패리티 비트는 n비트에서 시작해 n비트 만큼을 포함하고, n비트 씩 건너뛴 비트들을 대상으로 패리티 비트가 결정된다.

< 문제 >

정보 비트 1101에 홀수 패리티 비트를 적용하여 해밍 코드로 변환하시오.

1	2	3	4	5	6	7
0	✓	1	✓	1	0	1

1, 3, 5, 7

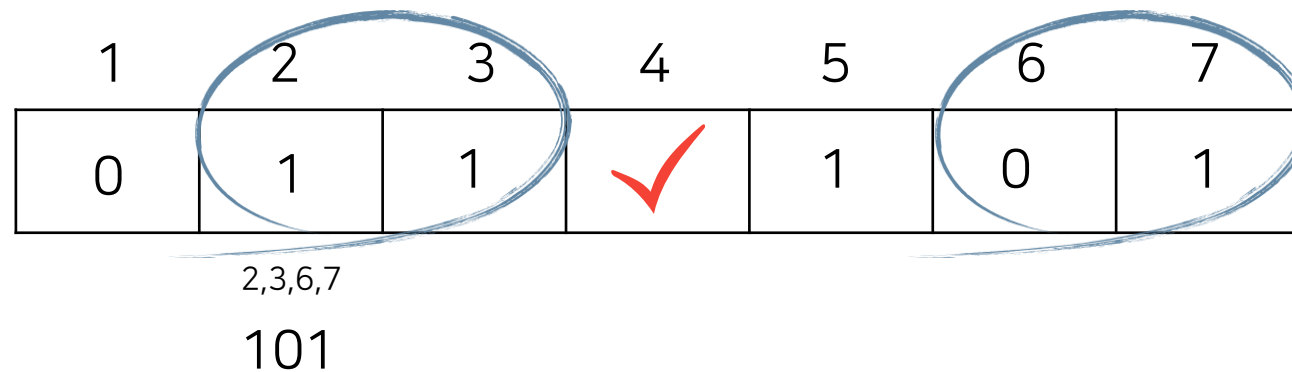
111

Hamming Code

- ✓ n번째 패리티 비트는 n비트에서 시작해 n비트 만큼을 포함하고, n비트 씩 건너뛴 비트들을 대상으로 패리티 비트가 결정된다.

< 문제 >

정보 비트 1101에 홀수 패리티 비트를 적용하여 해밍 코드로 변환하시오.



Hamming Code

- ✓ n번째 패리티 비트는 n비트에서 시작해 n비트 만큼을 포함하고, n비트 씩 건너뛴 비트들을 대상으로 패리티 비트가 결정된다.

< 문제 >

정보 비트 1101에 홀수 패리티 비트를 적용하여 해밍 코드로 변환하시오.

1	2	3	4	5	6	7
0	1	1	1	1	0	1

4,5,6,7

101

Hamming Code

✓ 2bit의 오류 검출 가능하고 1bit 오류 교정 가능.

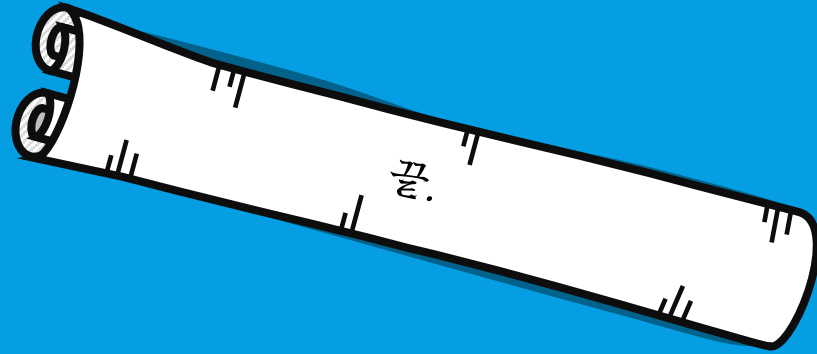
1	2	3	4	5	6	7
0	1	0 ✓	1	1	0	1

1, 3, 5, 7 => 0 0 1 1 1

2, 3, 6, 7 => 1 0 0 1 1 거짓 1, 참 0

4, 6, 7, 8 => 1 1 0 1 0

011 => 3



감사합니다.