

The background features a vibrant, abstract composition of overlapping circular splatters in shades of orange, red, purple, and blue, transitioning from left to right. A solid blue horizontal bar is positioned in the upper right quadrant. The text '비트마스킹' is displayed in white on a black background to the right of the splatters.

# 비트마스킹

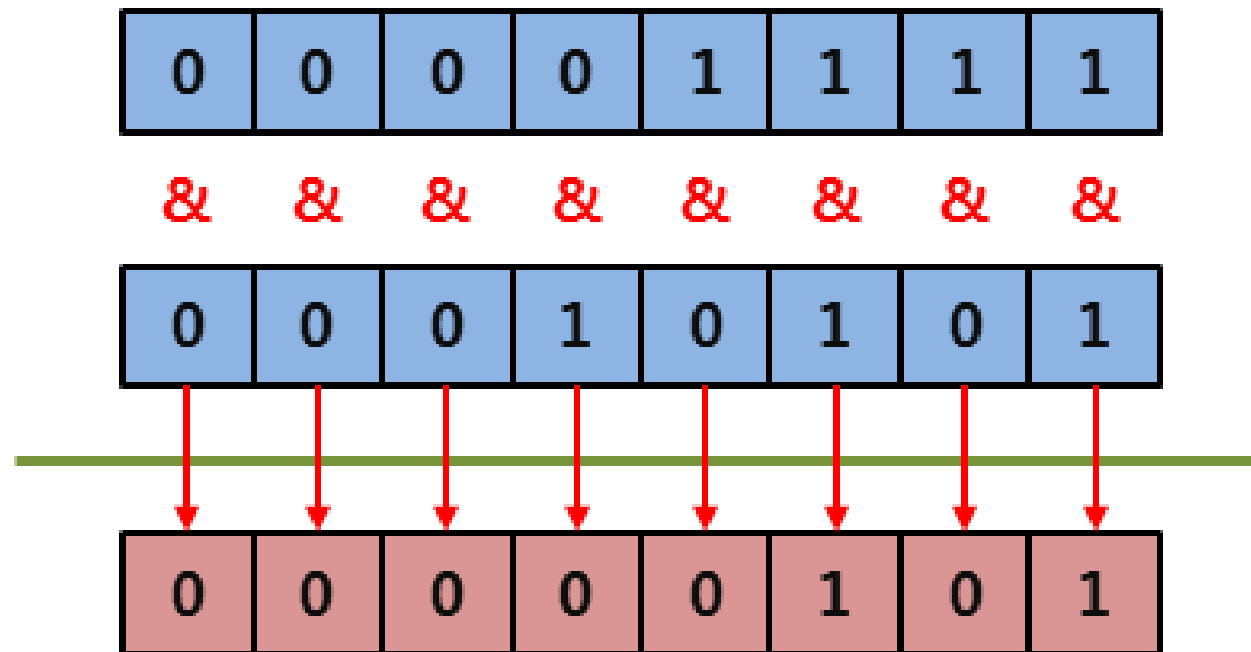
---

# 비트 연산자(bitwise operator)

- $\&$  : 대응되는 비트가 모두 1이면 1을 반환함
- $|$  : 대응되는 비트 중에서 하나라도 1이면 1을 반환함
- $\wedge$  : 대응되는 비트가 서로 다르면 1을 반환함
- $\sim$  : 비트를 1이면 0으로, 0이면 1로 반전
- $\ll$  : 지정한 수만큼 비트들을 전부 왼쪽으로 이동시킴
- $\gg$  : 지정한 수만큼 비트를 전부 오른쪽으로 이동시킴

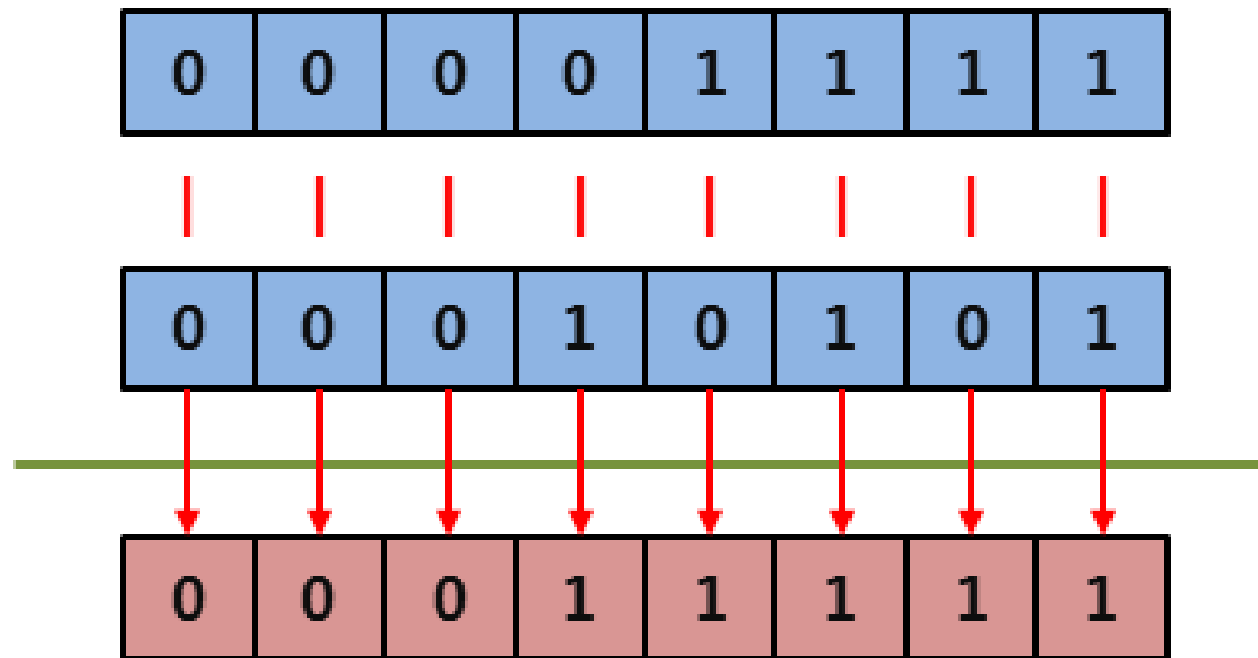
# 비트 AND 연산

비트 AND 연산



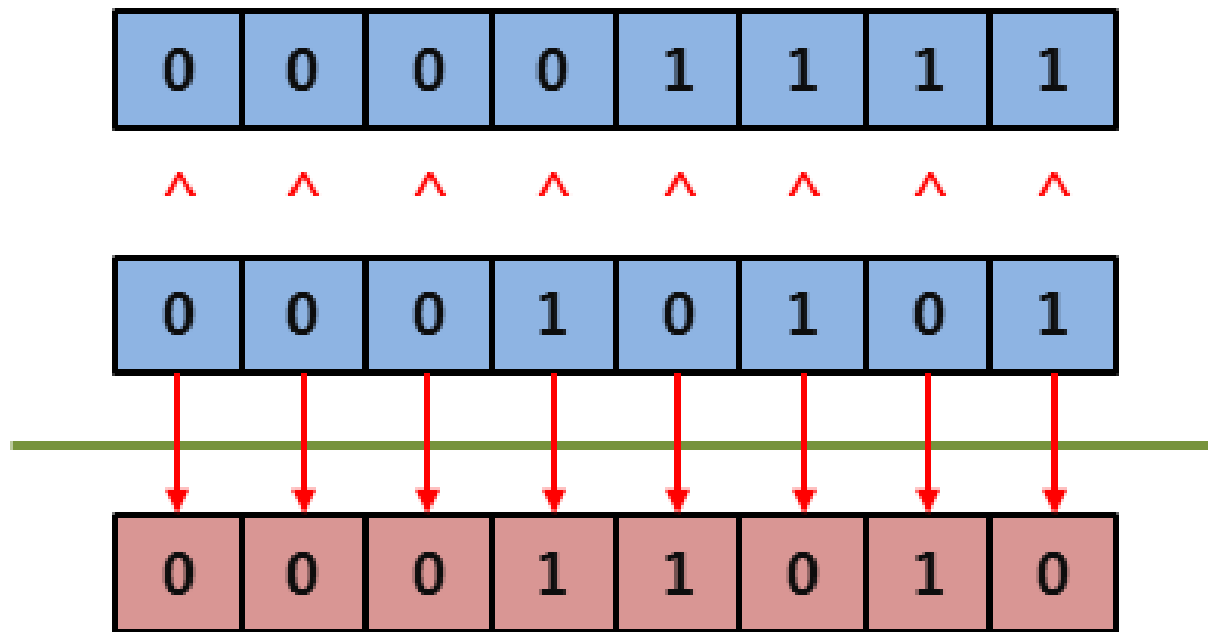
# 비트 OR 연산

비트 OR 연산



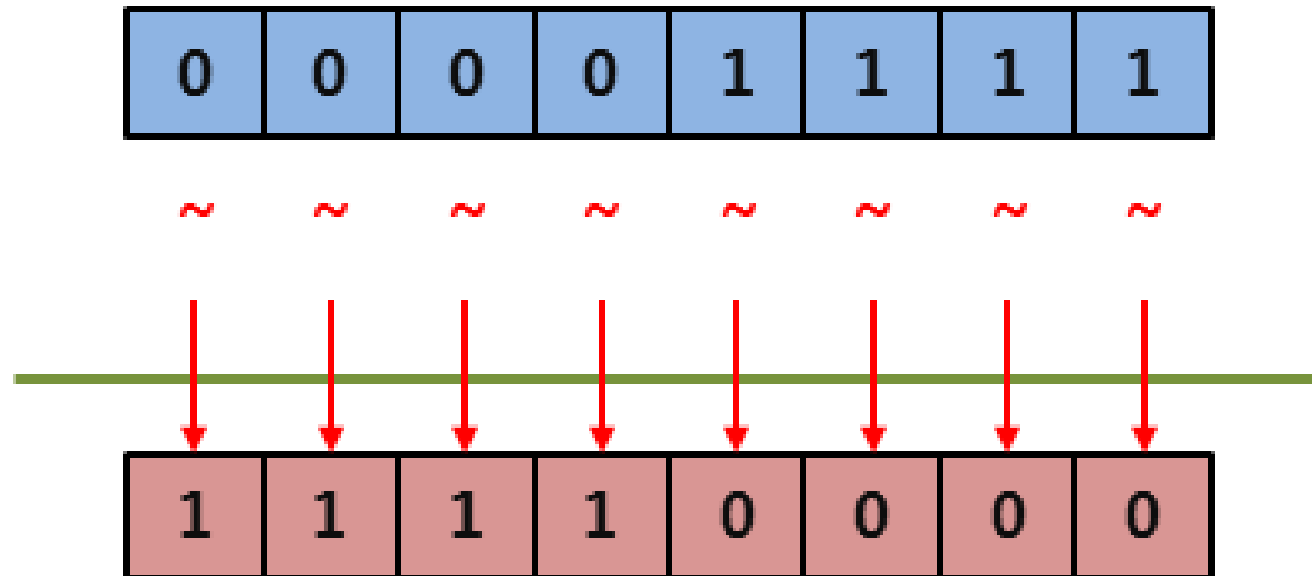
# 비트 XOR 연산

비트 XOR 연산



# 비트 NOT 연산

비트 NOT 연산



# 11723\_집합

## 문제

비어있는 공집합  $S$ 가 주어졌을 때, 아래 연산을 수행하는 프로그램을 작성하시오.

- `add x` :  $S$ 에  $x$ 를 추가한다. ( $1 \leq x \leq 20$ )  $S$ 에  $x$ 가 이미 있는 경우에는 연산을 무시한다.
- `remove x` :  $S$ 에서  $x$ 를 제거한다. ( $1 \leq x \leq 20$ )  $S$ 에  $x$ 가 없는 경우에는 연산을 무시한다.
- `check x` :  $S$ 에  $x$ 가 있으면 1을, 없으면 0을 출력한다. ( $1 \leq x \leq 20$ )
- `toggle x` :  $S$ 에  $x$ 가 있으면  $x$ 를 제거하고, 없으면  $x$ 를 추가한다. ( $1 \leq x \leq 20$ )
- `all` :  $S$ 를  $\{1, 2, \dots, 20\}$  으로 바꾼다.
- `empty` :  $S$ 를 공집합으로 바꾼다.


## 입력

첫째 줄에 수행해야 하는 연산의 수  $M$  ( $1 \leq M \leq 3,000,000$ )이 주어진다.

둘째 줄부터  $M$ 개의 줄에 수행해야 하는 연산이 한 줄에 하나씩 주어진다.

## 출력

`check` 연산이 주어질때마다, 결과를 출력한다.



비트마스킹?





ADD



REMOVE



CHECK



# TOGGLE



ALL