

알튜비튜

스택, 큐, 덱

오늘은 STL에서 제공하는 container adaptor인 stack과 queue 그리고 sequence container인 deque에 대해 알아봅니다.
가장 대표적이면서 가장 중요하기도 한 자료구조들 입니다.

/<> 1918번 : 후위 표기식 - Gold 2

문제

- 중위 표기식이 주어졌을 때 후위 표기식으로 고치는 프로그램 만들기

제한 사항

- 수식의 피연산자는 알파벳 대문자
- -A와 같이 연산자가 피연산자 앞에 오거나 AB와 같이 * 가 생략되는 수식은 주어지지 않음
- 표기식은 알파벳 대문자와 +, -, *, /, (,)로만 이루어짐
- 길이는 100을 넘지 않음

예제 입력1

A*(B+C)

예제 출력1

ABC+*

중위 표기법

- 연산자를 두 연산자 사이에 표기하는 방법으로 가장 일반적으로 사용됨
- ex. $A+B$

전위 표기법

- 연산자를 먼저 표시하고 연산자에 필요한 피연산자를 나중에 표기하는 방법
- ex. $+AB$

후위 표기법

- 피연산자를 먼저 표시하고 연산자를 나중에 표기하는 방법
- Ex. $AB+$

중위 표기식 → 후기 표기식

$A + B * C$



연산자 우선순위에 따라
괄호로 묶기

$(A + (B * C))$



$(A + (B * C))$

연산자를
괄호의 **오른쪽**으로
옮겨주기



$A B C * +$

$A + B * C$



$A B C * +$

Hint

1. 피연산자의 순서는 변하지 않아요!
2. 연산자 우선순위에 따라 연산자의 순서를 변화시켜 볼까요?

연산자(+, -, *, /, (,)) 우선순위

$$(A + (B * C)) \rightarrow A B C * +$$

곱셈(*), 나눗셈(/)을 덧셈(+)과 뺄셈(-)보다 먼저 수행한다

→ 곱셈과 나눗셈이 덧셈과 뺄셈보다 우선순위가 높다!



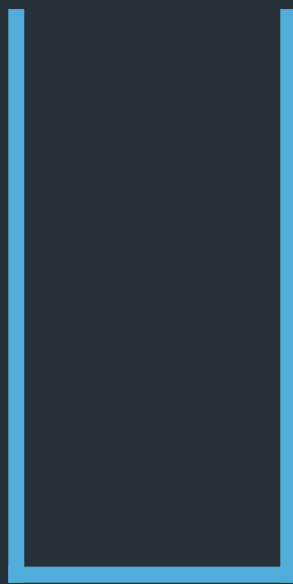
2
↓

1
↓

*, / 의 우선순위 > +, - 의 우선순위

중위표기식

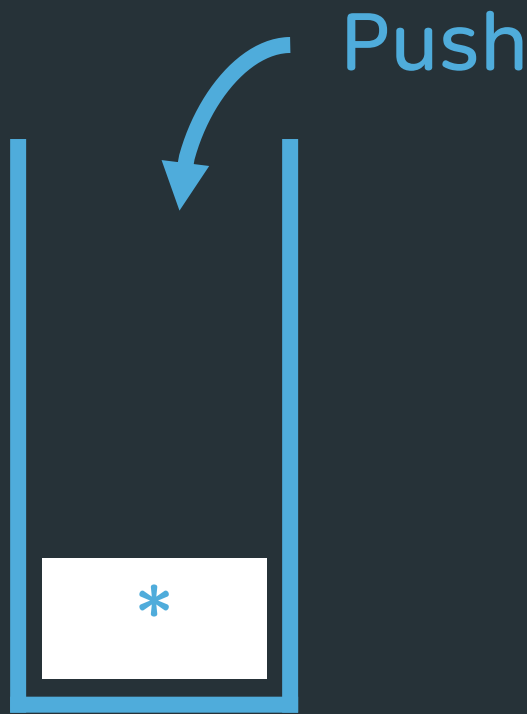
$A*(B+C)$



후위표기식

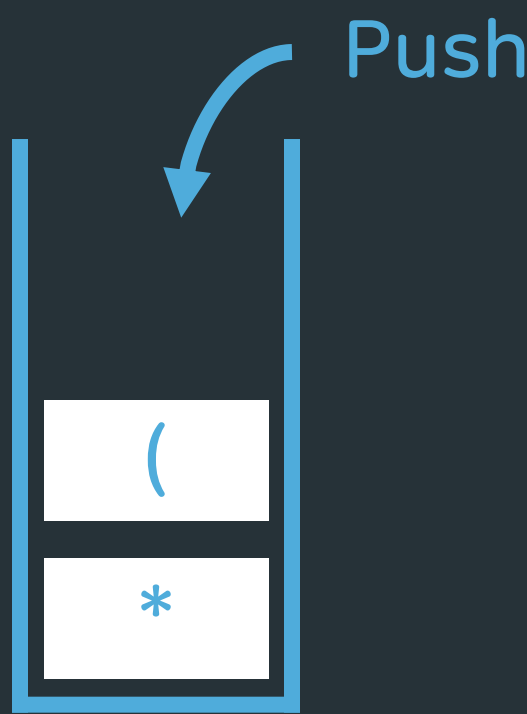
A

$A*(B+C)$



A

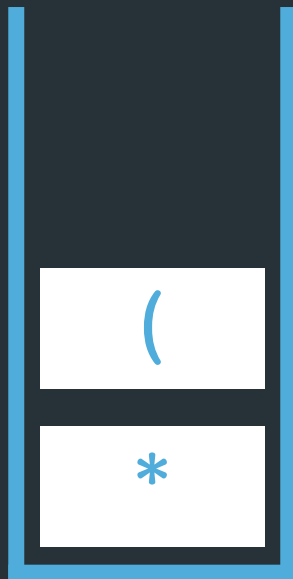
$A*(B+C)$



A

중위표기식

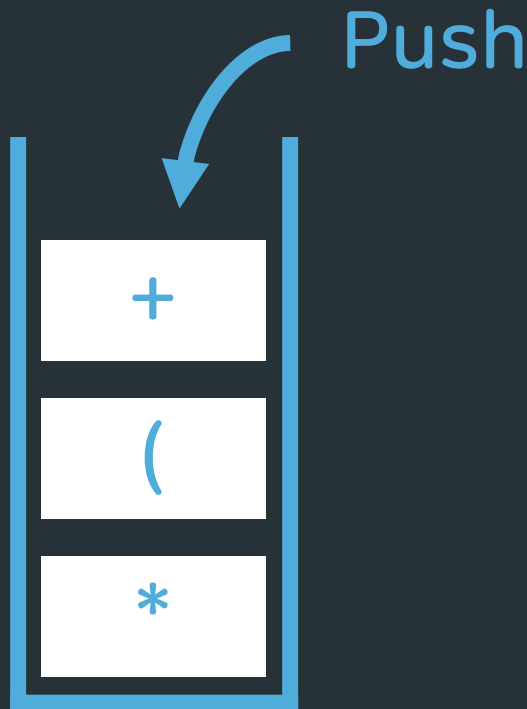
$A*(B+C)$



후위표기식

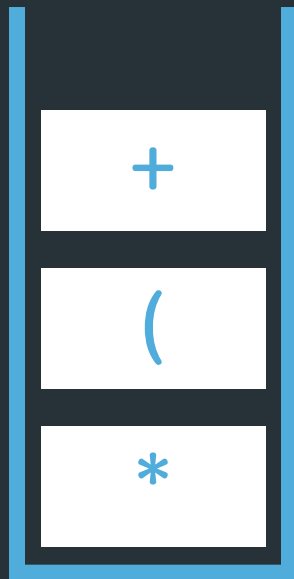
AB

$A*(B+C)$



AB

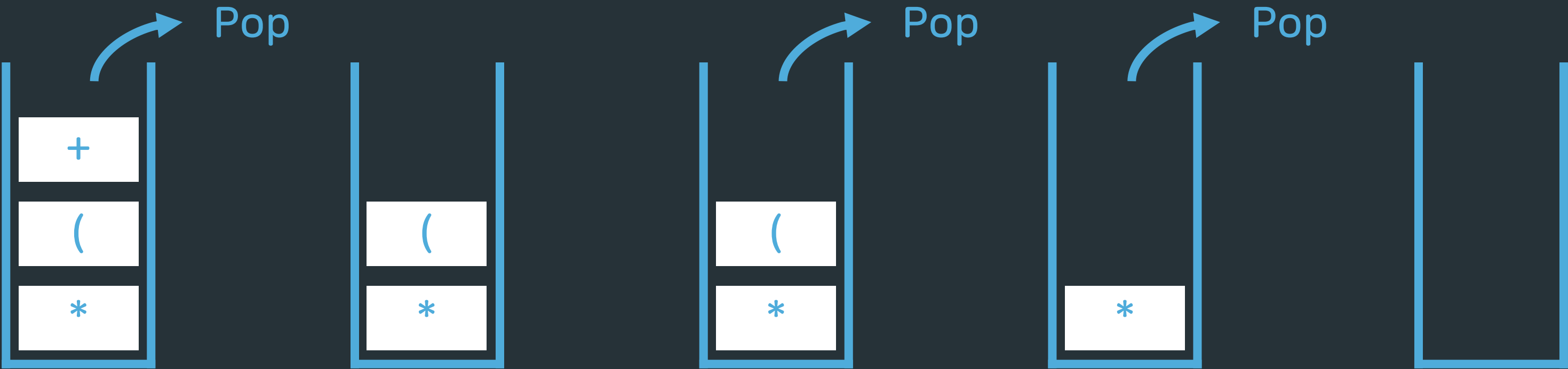
$A*(B+C)$



ABC

중위표기식

A*(B+C)



후위표기식

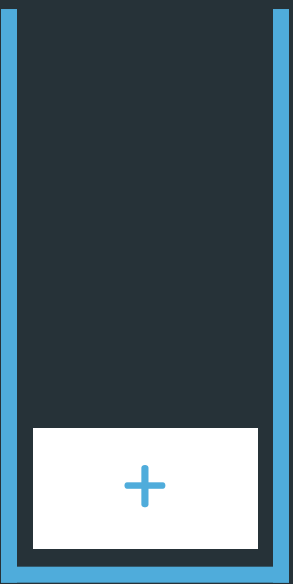
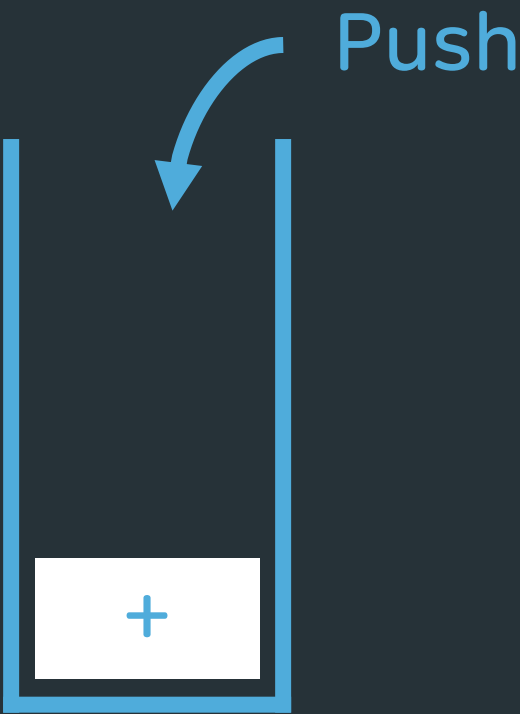
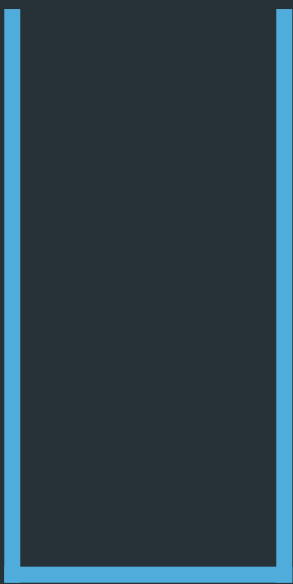
ABC ABC+ ABC+ ABC+ ABC+*

중위표기식

$A+B*C-D/E$

$A+B*C-D/E$

$A+B*C-D/E$



후위표기식

A

A

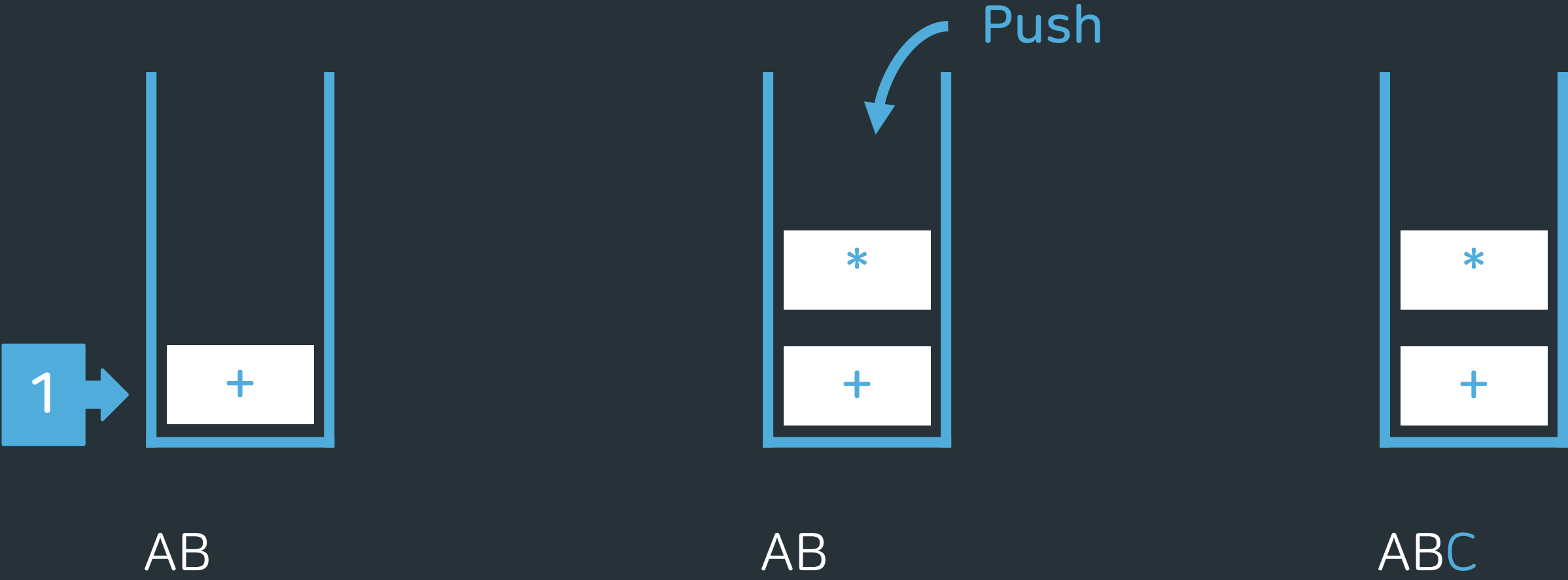
AB

중위표기식

2
↓
 $A+B * C - D / E$

$A+B * C - D / E$

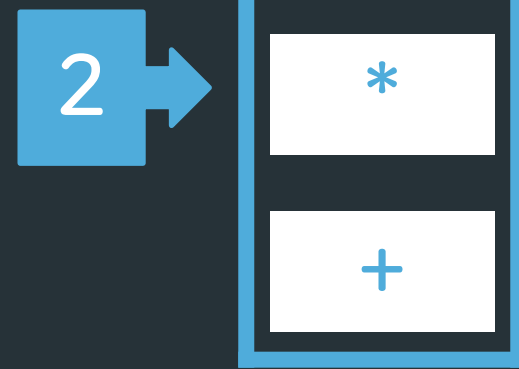
후위표기식



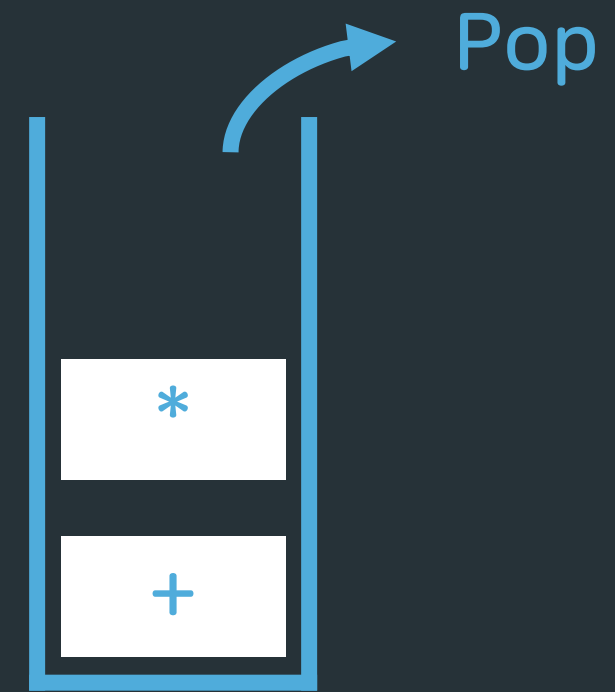
예제 입력4

중위표기식

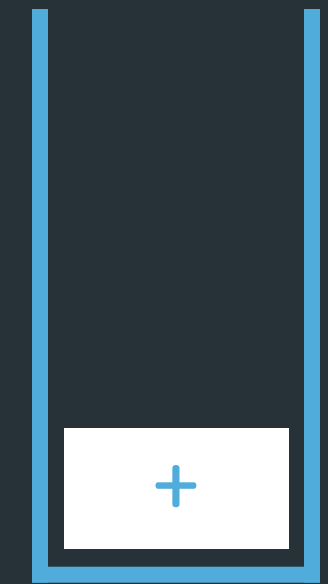
1
↓
 $A+B*C-D/E$



ABC



ABC



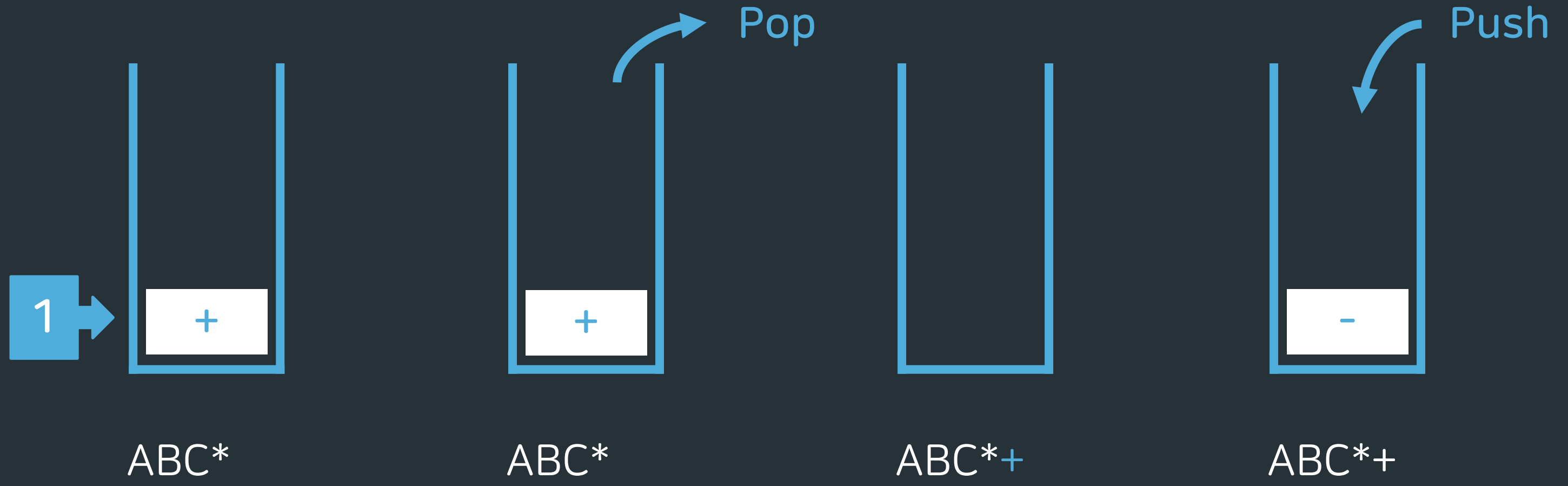
ABC*

후위표기식

중위표기식

1
↓
 $A+B*C-D/E$

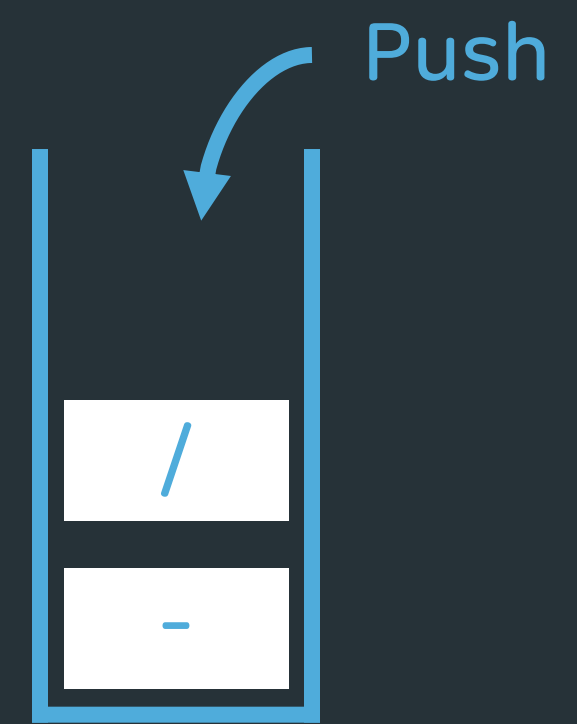
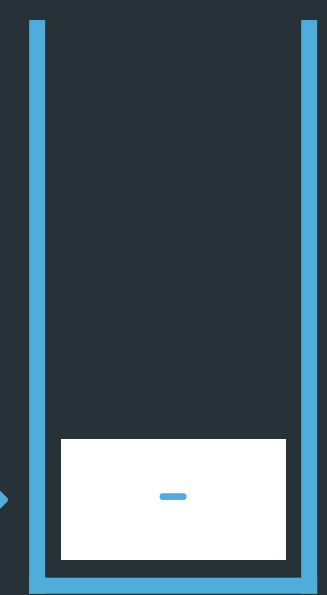
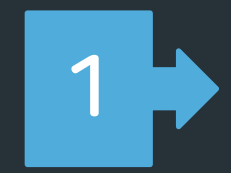
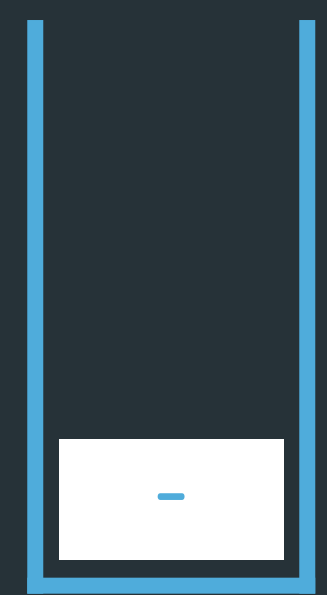
후위표기식



중위표기식

$A+B*C-D/E$

$A+B*C-D/E$



후위표기식

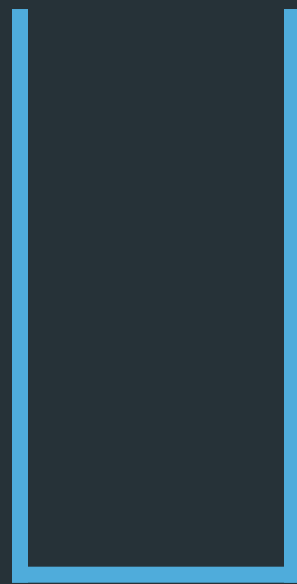
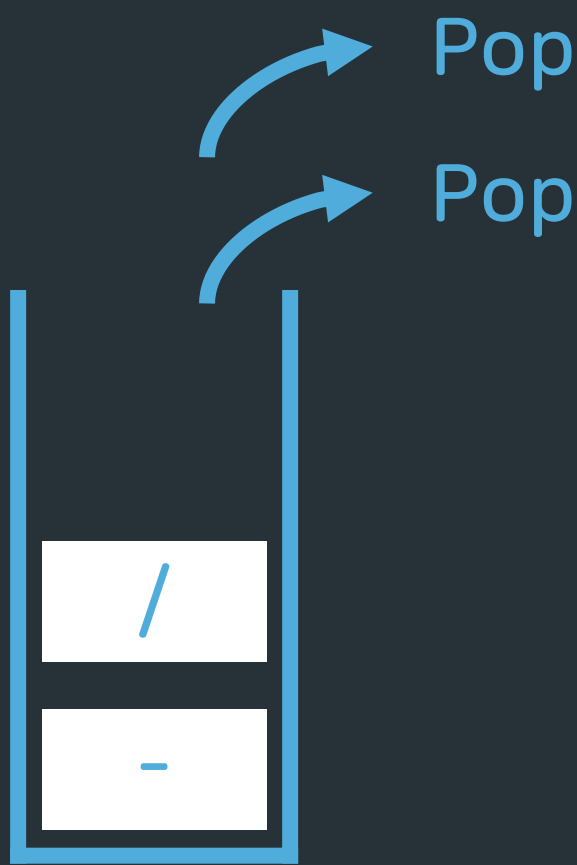
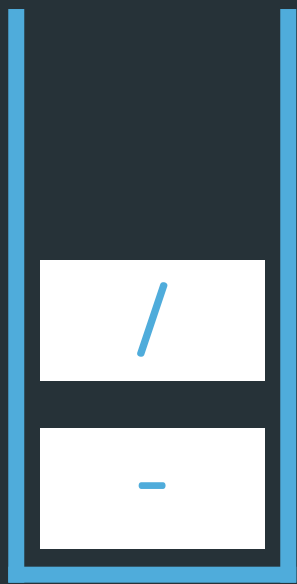
$ABC*+D$

$ABC*+D$

$ABC*+D$

중위표기식

$A+B*C-D/E$



후위표기식

$ABC*+DE$

$ABC*+DE$

$ABC*+DE/-$

코딩테스트 연습 > 스택/큐 > 기능개발 - Level 3

문제

먼저 배포되어야 하는 순서대로 작업의 진도가 적힌 정수 배열 progresses와 각 작업의 개발 속도가 적힌 정수 배열 speeds가 주어질 때, 각 배포마다 몇 개의 기능이 배포되는지를 return 하는 프로그램 만들기

- 각 기능은 진도가 100%일 때 서비스에 반영할 수 있다.
- 뒤에 있는 기능이 앞에 있는 기능보다 먼저 개발될 수 있고, 이때 뒤에 있는 기능은 앞에 있는 기능이 배포될 때 함께 배포된다.

제한 사항

- 작업의 개수는 100개 이하
- 작업 진도는 100 미만의 자연수 $1 \leq \text{progresses}[i] < 100$
- 작업 속도는 100 이하의 자연수 $1 \leq \text{speeds}[i] \leq 100$
- 배포는 하루에 한 번만 할 수 있으며, 하루의 끝에 이루어진다고 가정한다.

입출력 예

progresses	speeds	return
[93, 30, 55]	[1, 30, 5]	[2, 1]

Hint

1. 개발이 끝난 후에 배포할 수 있어요. 먼저 개발 완료 시간을 구해봅시다!
2. 먼저 배포되어야 하는 작업 순서가 정해져 있네요! 큐를 이용해볼까요?

입출력 예1

progresses

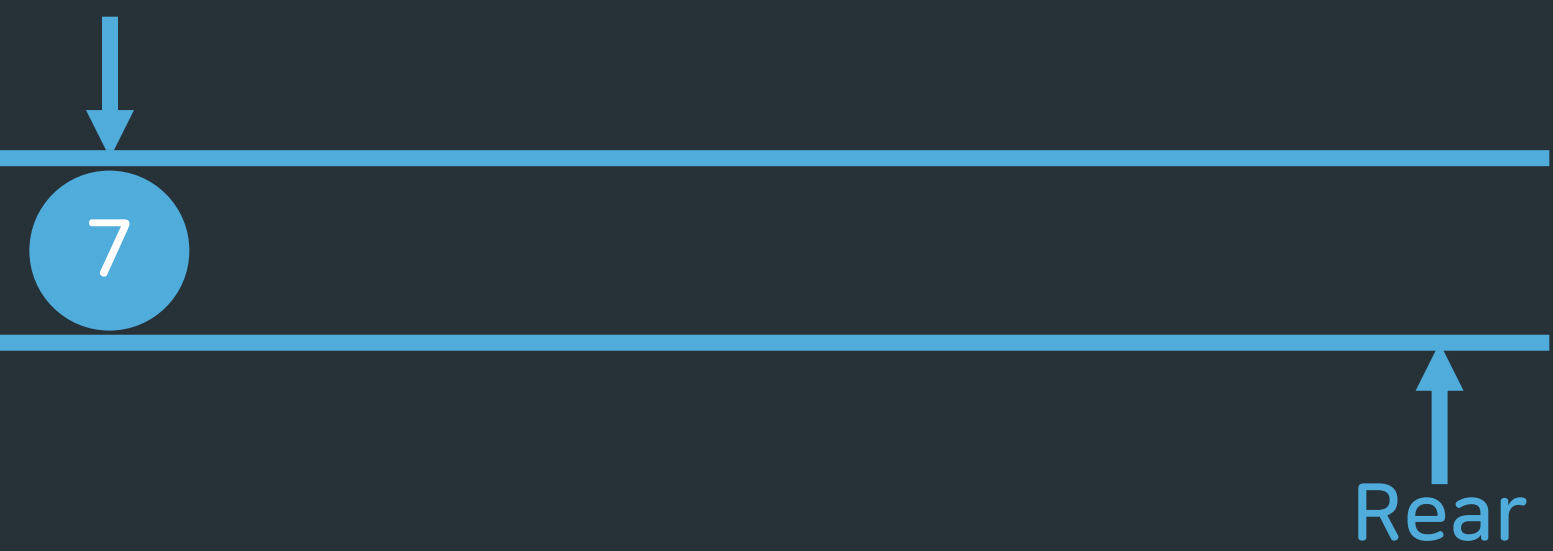
93	30	55
----	----	----

speeds

1	30	5
---	----	---

완료 시간
 $\text{Ceil}((100 - 93) / 1)$

Front



입출력 예1

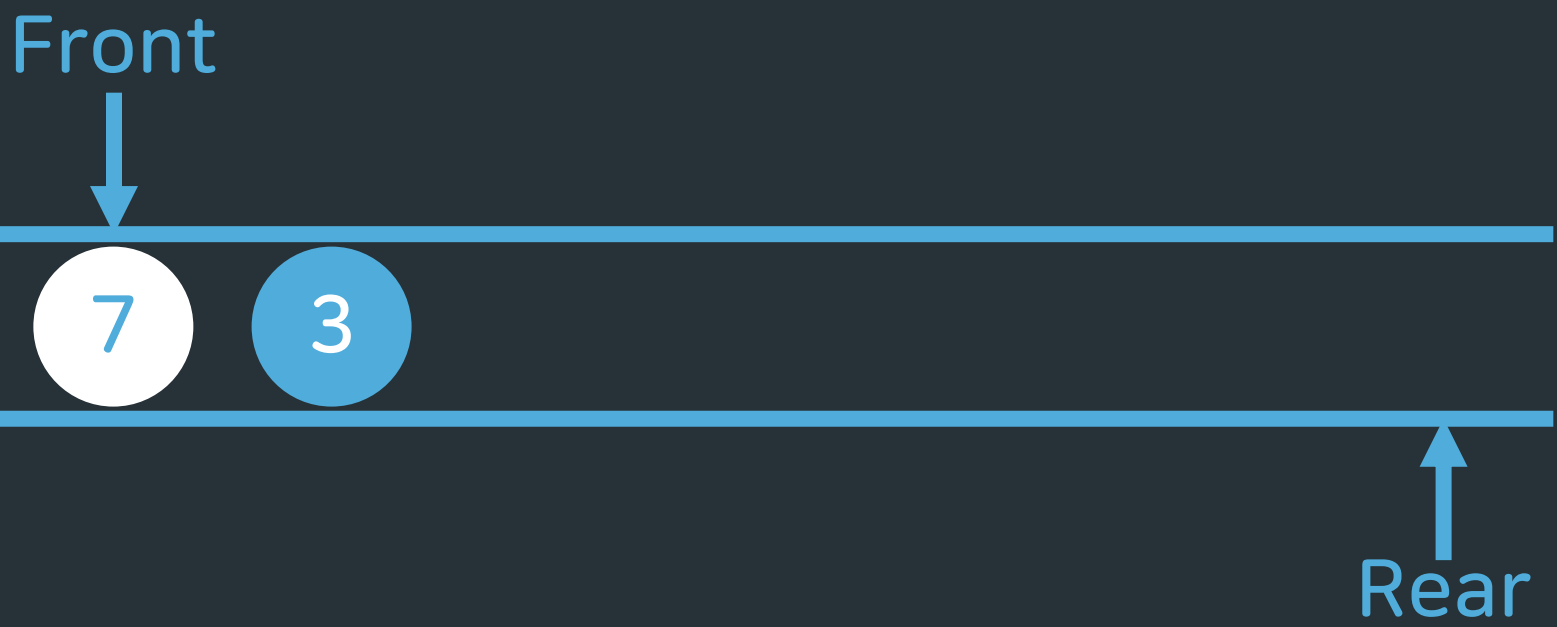
progresses

93	30	55
----	----	----

speeds

1	30	5
---	----	---

완료 시간
 $\text{Ceil}((100 - 30) / 30)$



입출력 예1

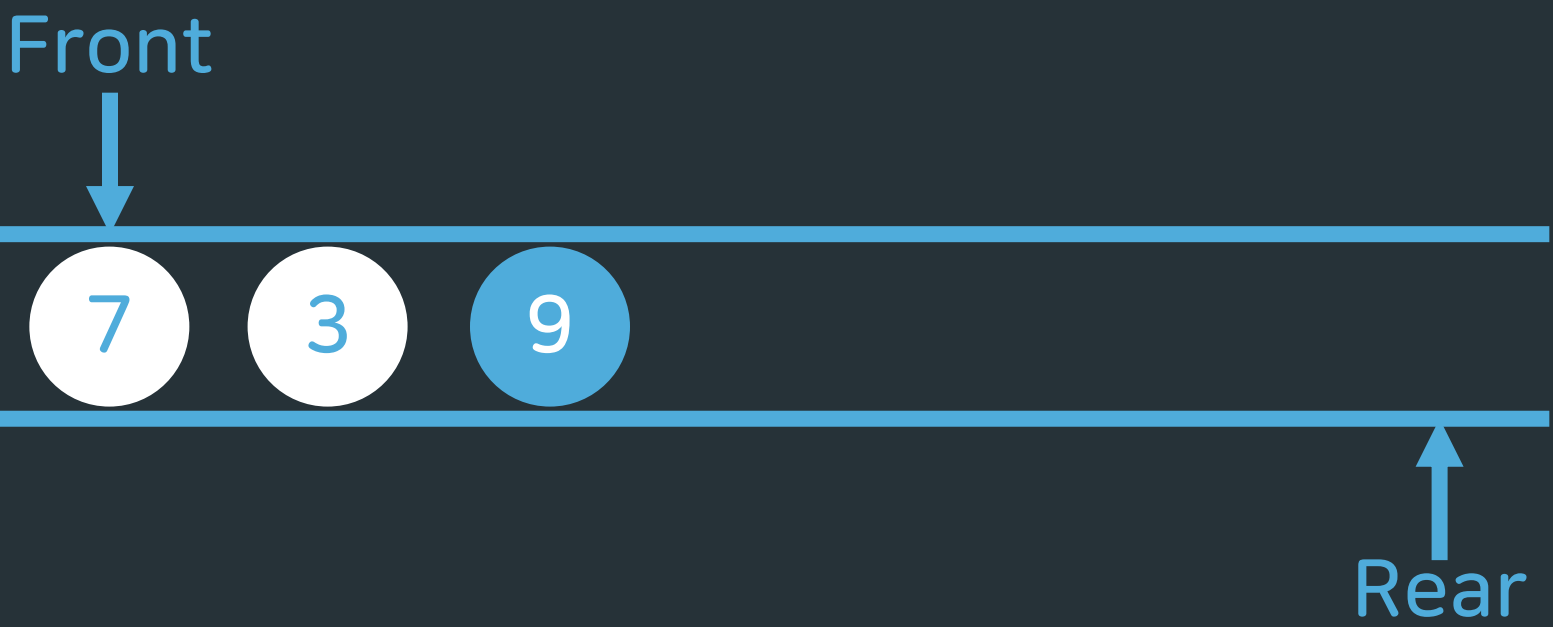
progresses

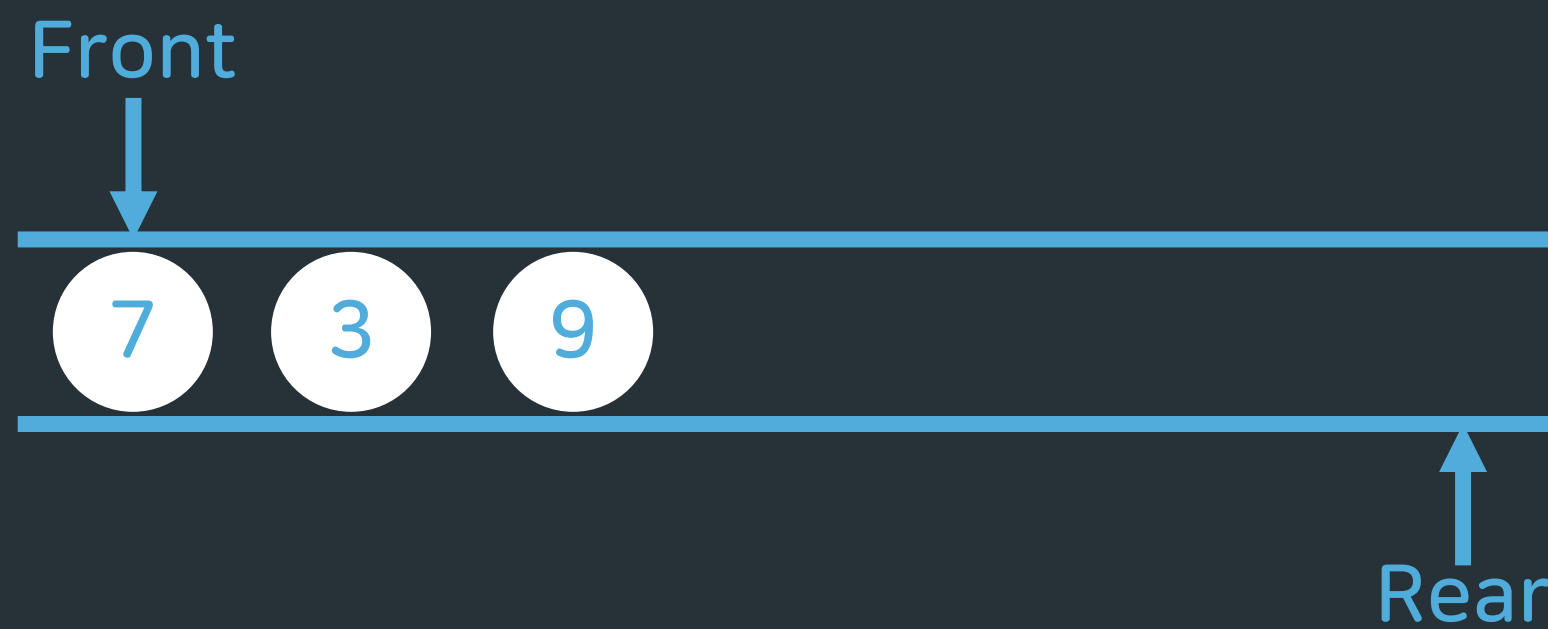
93	30	55
----	----	----

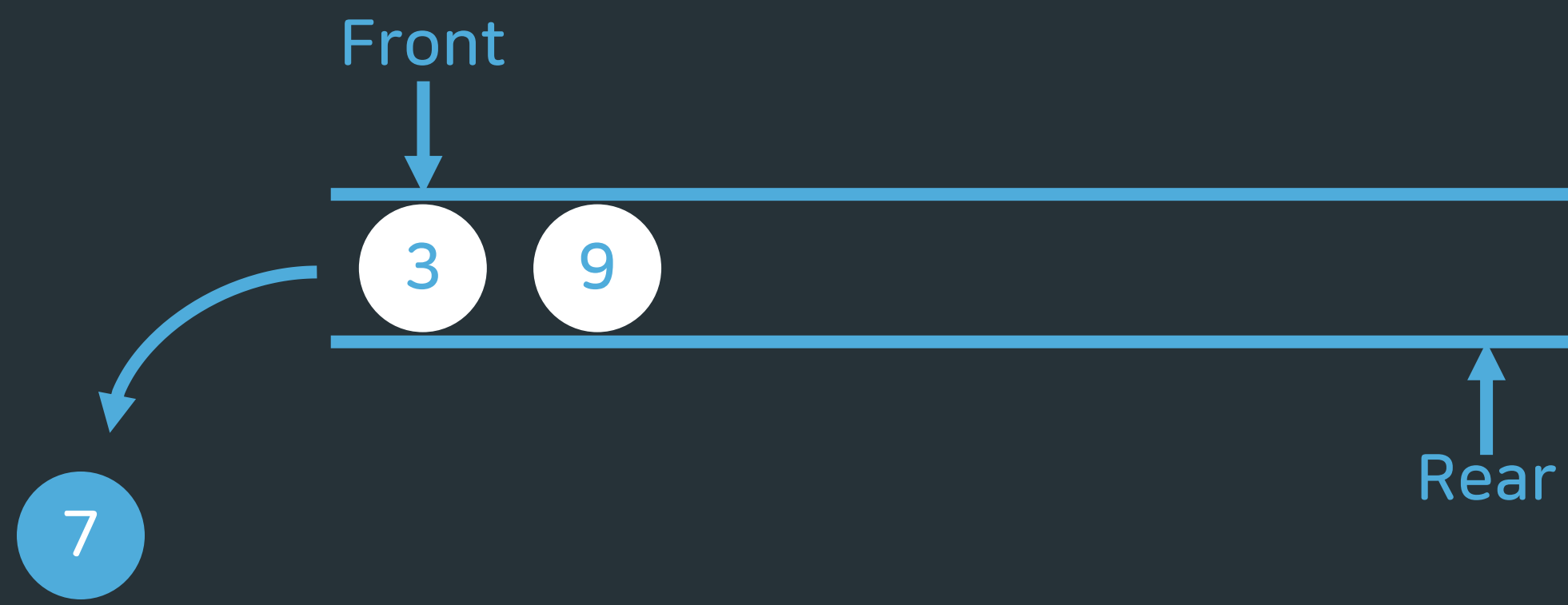
speeds

1	30	5
---	----	---

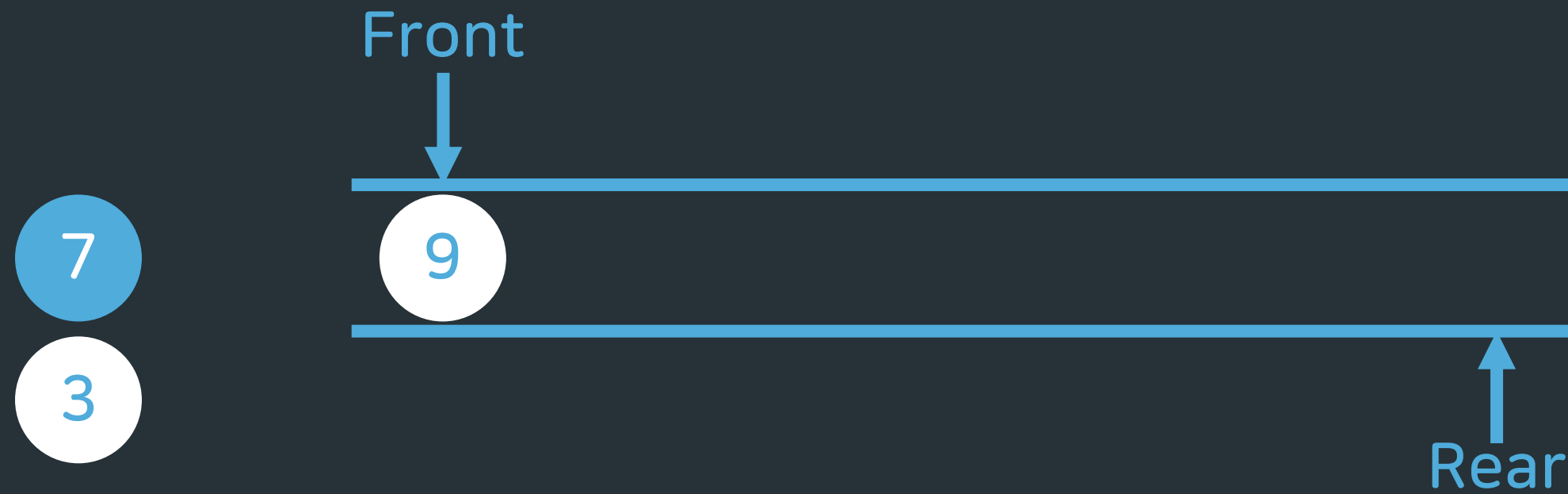
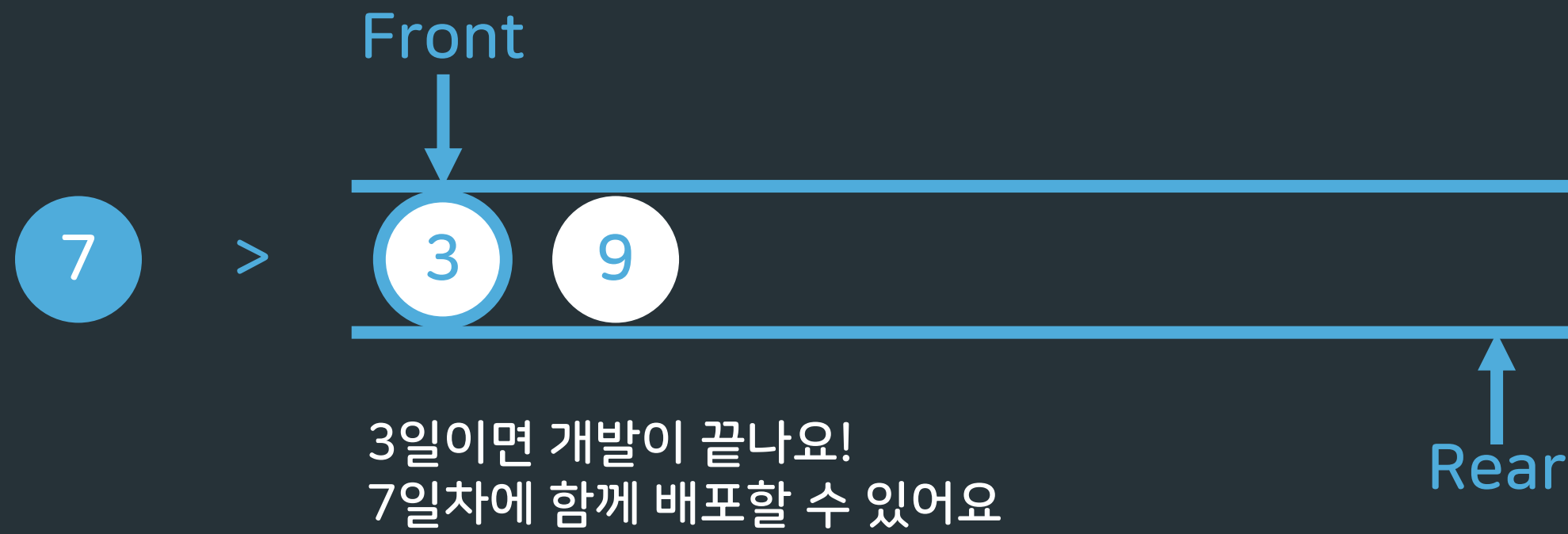
완료 시간
 $\text{Ceil}((100 - 55) / 5)$



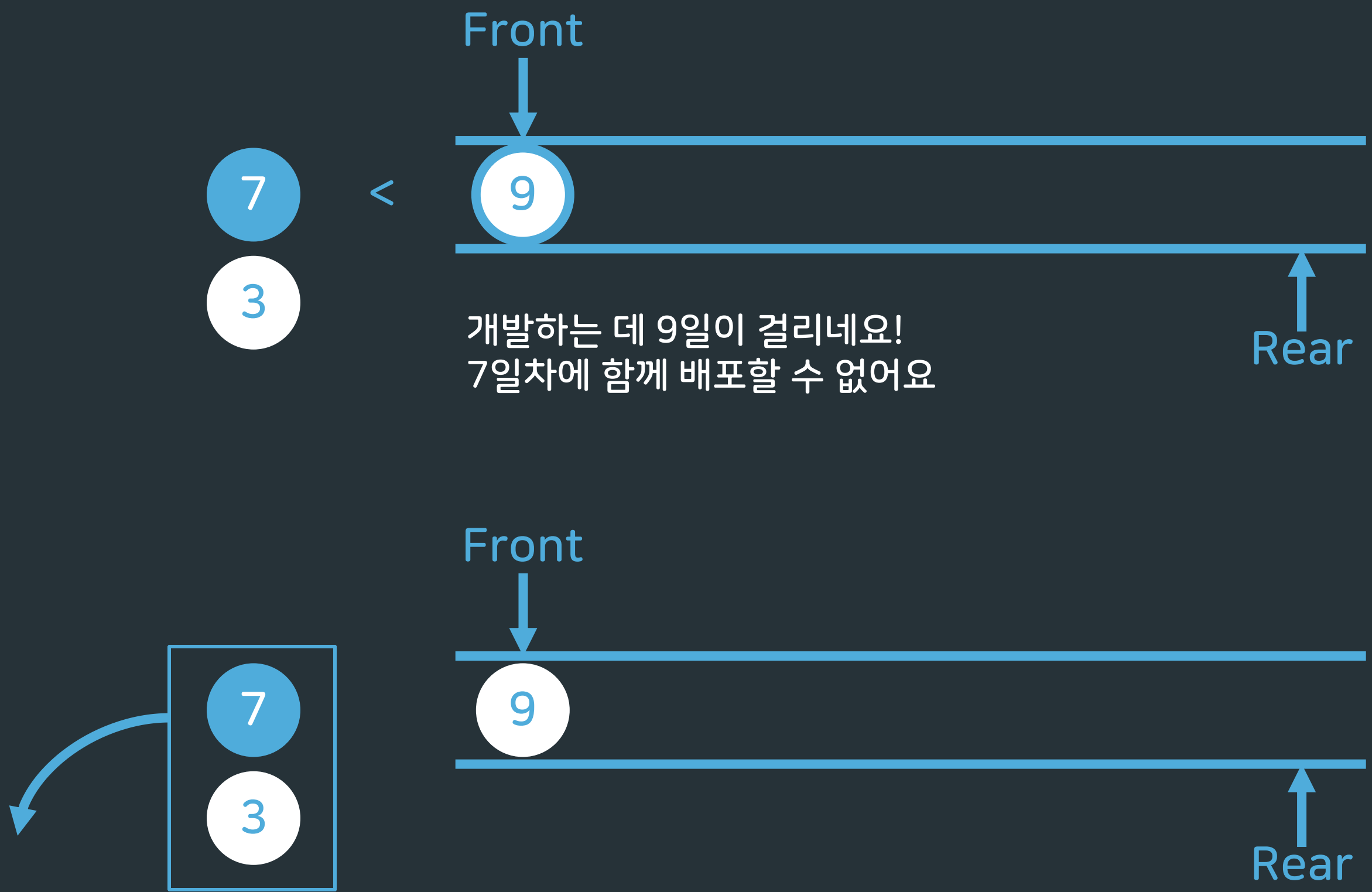


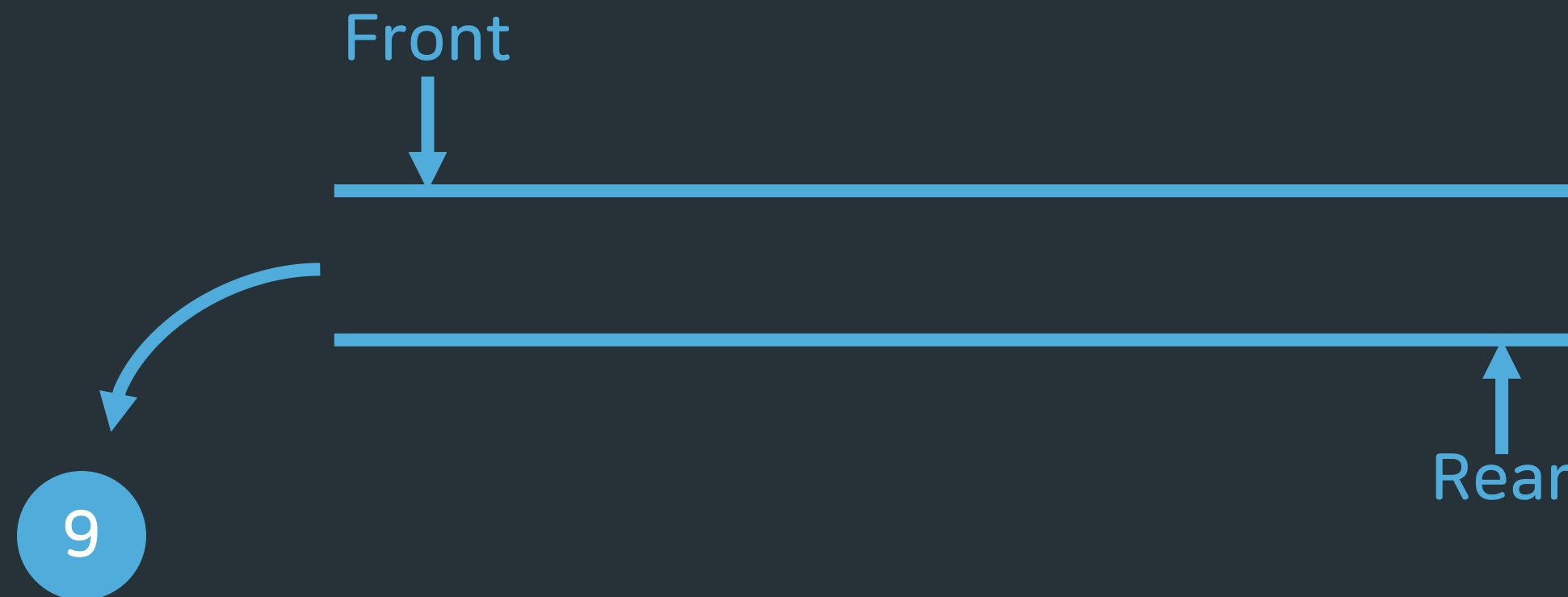


현재 작업을 완료하기까지 7일이 걸려요.
현재 작업과 배포될 수 있는 작업이 있을까요?



입출력 예1





남은 작업이 없어요.
9일차엔 하나만 배포하면 될 것 같아요!

/<> 10757번 : 큰 수 A + B - Bronze 5

문제

- 두 정수 A와 B를 입력받은 다음, A+B를 출력하는 프로그램을 작성하시오.

제한 사항

- A와 B의 입력 범위는 $0 < A, B < 10^{10000}$

예제 입력1

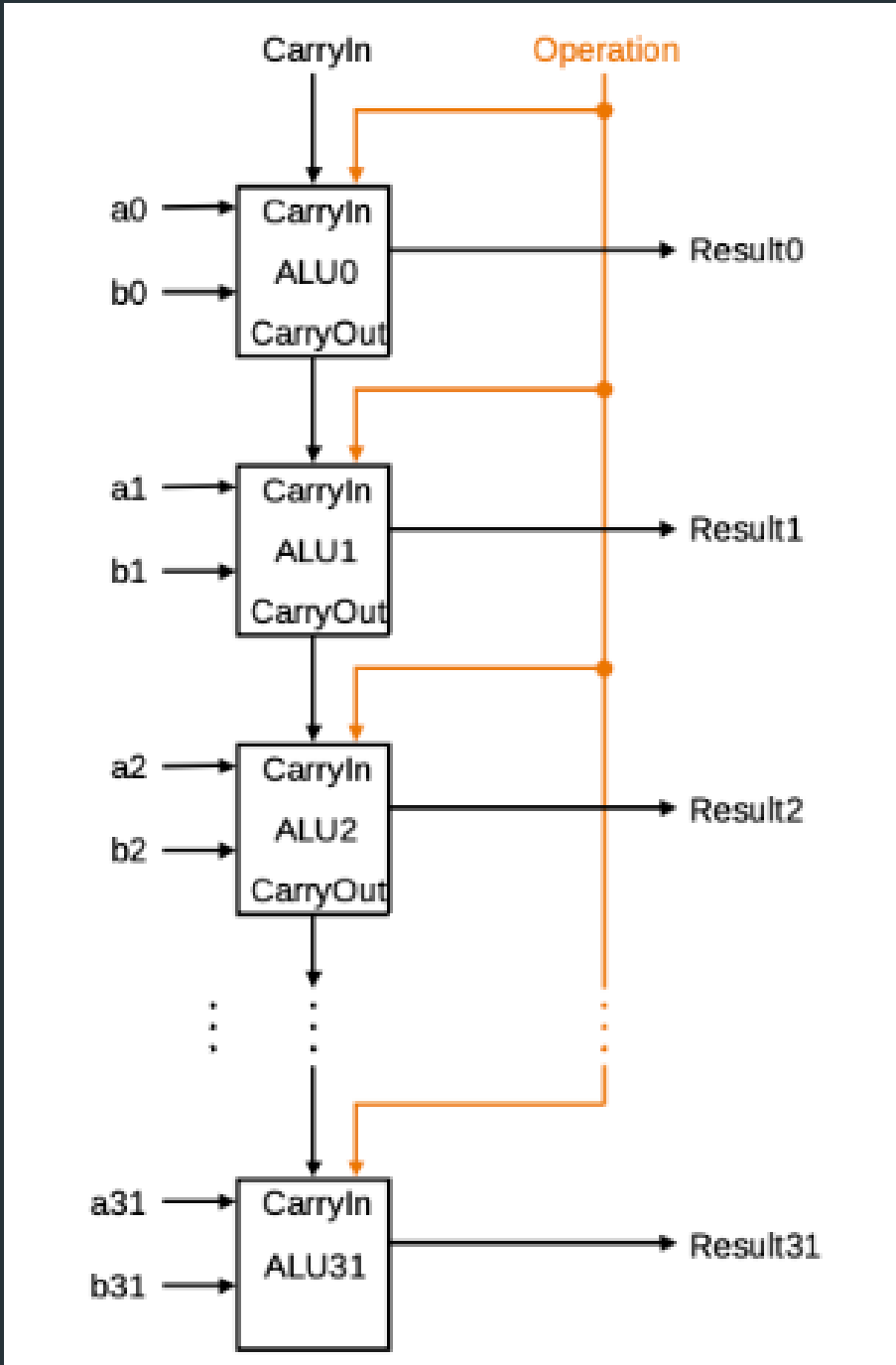
```
9223372036854775807 9223372036854775808
```

예제 출력1

```
18446744073709551615
```

- A,B가 최대 10^{10000}
- int : 최대 $2^{31} - 1$, long : 최대 $2^{63} - 1$
- $\text{int} < \text{long} \lllllllllll 10^{10000}$
- 따라서, 덧셈을 직접 구현하자

덧셈의 원리?!



$$\begin{array}{l} A = 54321 \\ B = 65432 \end{array}$$

$$\begin{array}{l} A = 456321 \\ B = 5432 \end{array}$$

$$\begin{array}{l} A = 1234 \\ B = 65432 \end{array}$$

따라서 $A + B$ 를 할 때,

A 의 한자리 수 + B 의 한자리 수 + carry(올림수)

⇒ 연산을 모든 자리 수에 대해서 해주면 된다!

⇒ $O(N)$ 의 시간복잡도

수행 시간 계산해봅시다!

어떤 수 X 의 자리 수를 아는 방법은 무엇인가요?

1 ~ 9 : 1자리

10 ~ 99 : 2자리

.

.

$10^{(N-1)} \sim 10^N - 1$: N 자리

X 는 최대 10^{10000} 이므로 100001 자리

$O(N) \rightarrow 100001$ 번의 계산 < 1초

추가로 풀어보면 좋은 문제!

/<> 2164번 : 카드2 - Silver 4

/<> 18115번 : 카드 놓기 - Silver 3

/<> 4889번 : 안정적인 문자열 - Silver 1

/<> 17299번 : 오등큰수 - Gold 3