



# 데이터 구조

6주차: 중위표기법을 전위표기법, 후위표기법으로 변환

# 후위표기법

연산자  
스택



데이터 저장소

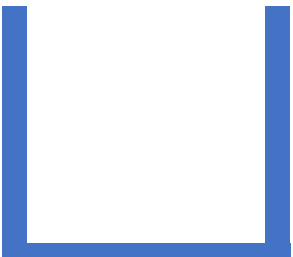


- 입력된 데이터를 왼쪽에서 오른쪽으로 문자 하나씩 입력 받는다.
- 첫번째 연산자는 무조건 stack에 push
- Stack에 Top이 가르키는 연산자(top\_op)와 입력 받은 연산자(new\_op)를 비교
  - $top\_op \geq new\_op$  인 경우:
    1. stack이 isEmpty() 일 때까지 반복 pop() 하여 저장소에 입력
    2. new\_op를 stack에 push()
  - $top\_op < new\_op$  인 경우:
    1. new\_op를 stack에 push()

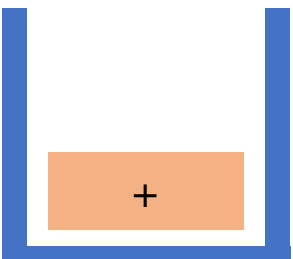
# 후위표기법

●  $A + B * C - D$

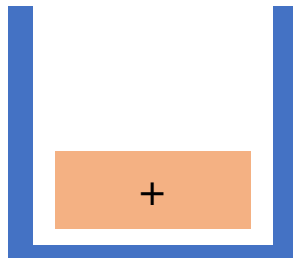
▪ Step 1



▪ Step 2



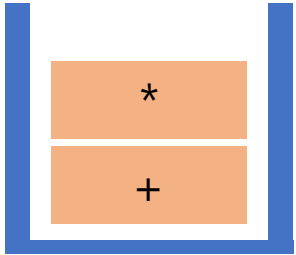
▪ Step 3



# 후위표기법

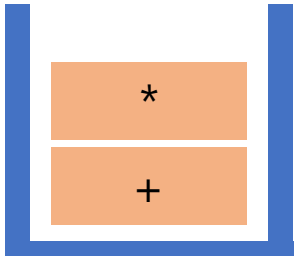
●  $A+B*C-D$

▪ Step 4

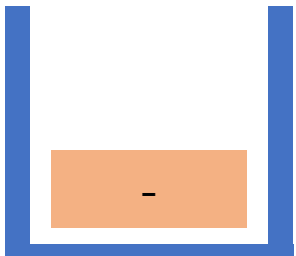


▪ "\*" 우선순위가 "+"보다 높으므로, "\*" push()

▪ Step 5



▪ Step 6

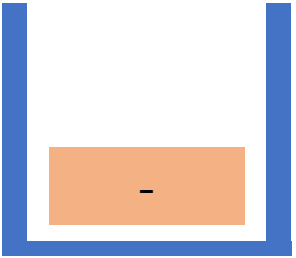


▪ "-" 우선순위가 "\*"보다 낮음  
stack에 저장된 모든 연산자를 pop() 후  
"-" push()

# 후위표기법

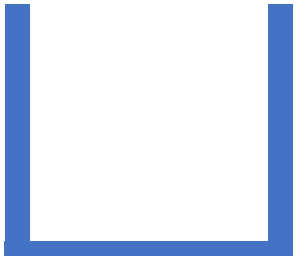
●  $A+B*C-D$

▪ Step 7



A	B	C	*	+	D	
---	---	---	---	---	---	--

▪ Step 8

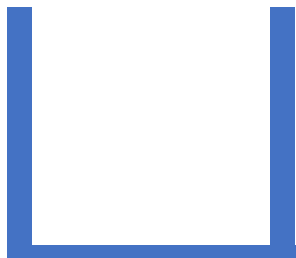


A	B	C	*	+	D	-
---	---	---	---	---	---	---

▪ 모든 입력된 데이터를 다 읽었으므로,  
stack에 저장된 데이터 반복하여 pop()

# 전위표기법

연산자  
스택



데이터 저장소

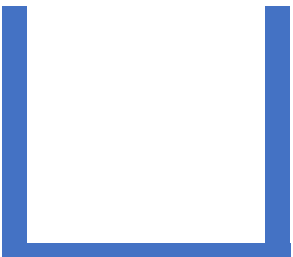


- 입력된 데이터를 오른쪽에서 왼쪽으로 문자 하나씩 입력 받는다.
- 첫번째 연산자는 무조건 stack에 push
- Stack에 Top이 가르키는 연산자(top\_op)와 입력 받은 연산자(new\_op)를 비교
  - $top\_op \geq new\_op$  인 경우:
    1. stack이 isEmpty() 일 때까지 반복 pop() 하여 저장소에 입력 (단, 반복시  $top\_op < new\_op$ 인 경우 push() 후 중단)
    2. new\_op를 stack에 push()
  - $top\_op < new\_op$  인 경우:
    1. new\_op를 stack에 push()

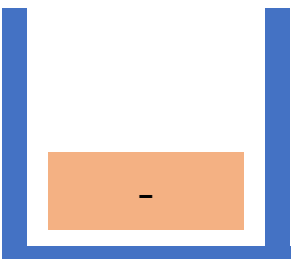
# 전위표기법

● A+B\*C-D

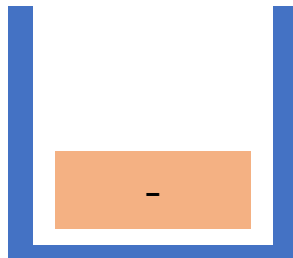
▪ Step 1



▪ Step 2



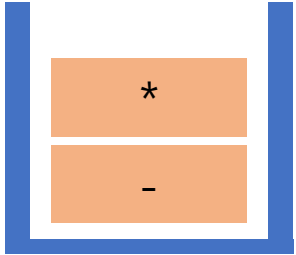
▪ Step 3



# 전위표기법

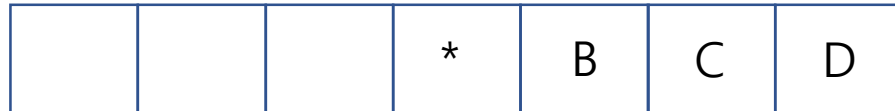
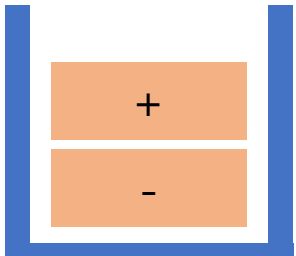
●  $A+B*C-D$

▪ Step 4



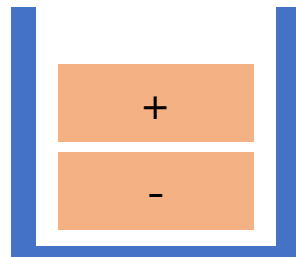
▪ "\*" 우선순위가 "-"보다 높으므로, "\*" push()

▪ Step 5



▪ "+" 우선순위가 "\*"보다 낮으므로, stack에 저장된 연산자를 반복하여 pop() 우선순위가 같은 "-" 때문에 반복 중지 후 "+" push()

▪ Step 6

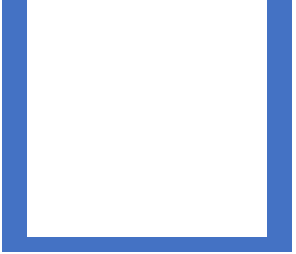




# 전위표기법

●  $A+B*C-D$

▪ Step 7



-	+	A	*	B	C	D
---	---	---	---	---	---	---

- 모든 입력된 데이터를 다 읽었으므로, stack에 저장된 데이터 반복하여 pop()