

03강

스택

컴퓨터과학과 정광식 교수

오늘의 학습목차

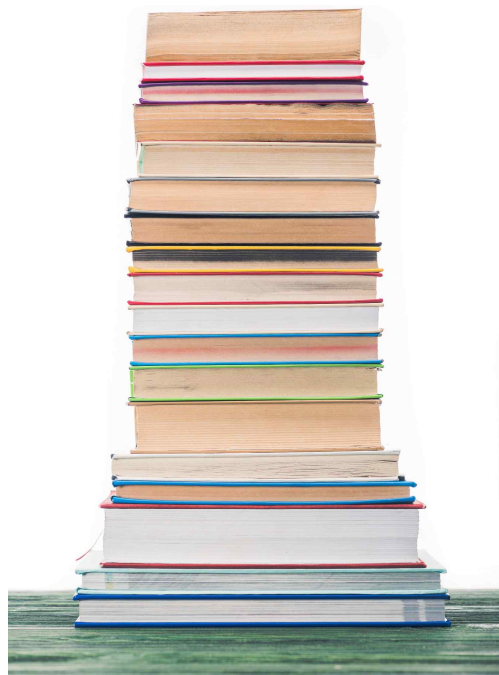
- 01 스택의 개념과 추상 자료형
- 02 스택의 응용과 구현
- 03 사칙 연산식의 전위/후위/중위 표현

01

스택의 개념과 추상자료형

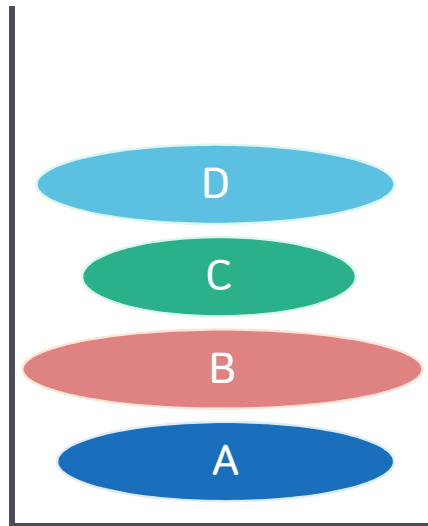
◆ 스택이란?

★ 스택의 정의



◆ 스택이란?

★ 스택의 정의



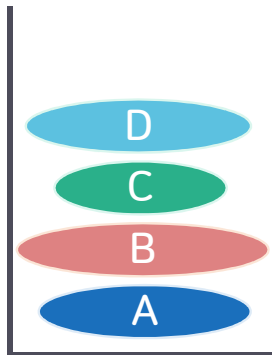
1

2

3

◆ 스택이란?

★ 스택의 정의



XML
DB
자료 구조
컴퓨터과학 개론

◆ 스택이란?

★ 스택의 정의

- ◆ 객체와 그 객체가 저장되는 순서를 기억하는 방법에 관한 추상 자료형
 - 가정 먼저 입력된 자료가 가장 나중에 출력되는 관계를 표현함

◆ 스택이란?

★ 스택의 정의

- ◆ 0개 이상의 원소를 갖는 유한 순서 리스트
- ◆ push(add)와 pop(delete)연산이 한곳에서 발생하는 자료구조

◆ 스택의 추상자료형

★ CreateS 연산

- Stack CreateS(maxStackSize) ::=
스택의 크기가 maxStackSize인 빈 스택을
생성하고 반환한다;

◆ 스택의 추상자료형

★ Push 연산

- Element Push(stack, item) ::=
 if (IsFull(stack))
 then { 'stackFull'을 출력한다; }
 else { 스택의 가장 위에 item을 삽입하고,
 스택을 반환한다; }

◆ 스택의 추상자료형

★ Push 연산

- Element Pop(stack) ::=
 if (IsEmpty(stack))
 then { 'stackEmpty'을 출력한다; }
 else {
 스택의 가장 위에 있는 원소(element)를
 삭제하고 반환한다;
 }

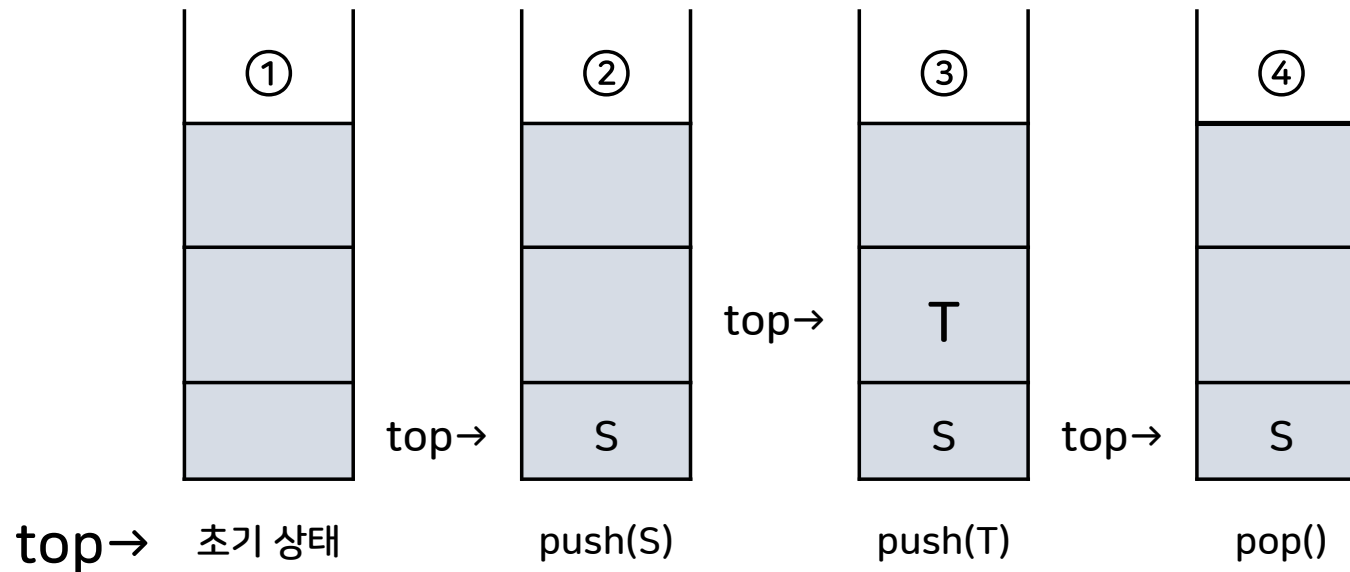
◆ 스택의 추상자료형

★ Pop/Push 연산의 실행

- ① CreateS(3);
- ② Push(stack, 'S');
- ③ Push(stack, 'T');
- ④ Pop(stack);
- ⑤ Push(stack, 'R');
- ⑥ Push(stack, 'P');
- ⑦ Push(stack, 'Q');
- ⑧ Pop(stack);

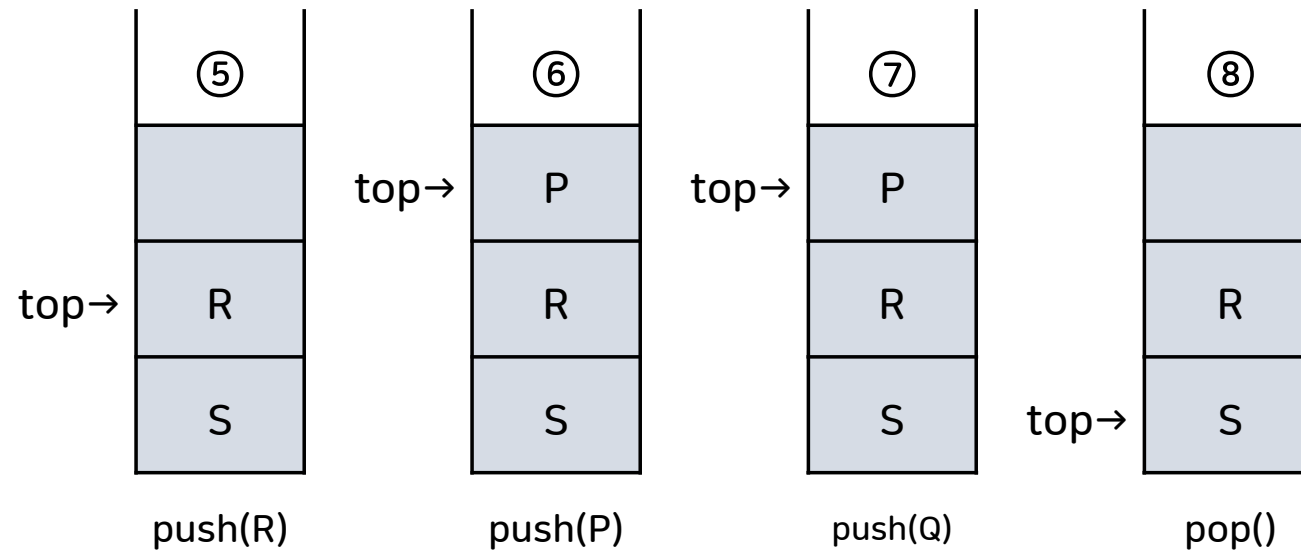
◆ 스택의 추상자료형

★ Pop/Push 연산의 실행



◆ 스택의 추상자료형

★ Pop/Push 연산의 실행



02

스택의 응용과 구현

◆ 스택의 응용

★ 스택의 다양한 응용

- ◆ 변수에 대한 메모리의 할당과 수집을 위한 시스템 스택
- ◆ 서브루틴 호출 관리를 위한 스택
- ◆ 연산자들 간의 우선순위에 의해 계산 순서가 결정되는 수식 계산
- ◆ 인터럽트의 처리와, 이후 리턴할 명령 수행 지점을 저장하기 위한 스택
- ◆ 컴파일러, 순환 호출 관리

◆ 스택의 연산

★ 스택의 삭제 연산

- ◆ 'top--'에서 사용된 '--' 연산자의 위치에 따라 연산의 적용순서가 달라질 수 있음

```
int a;  
a = 5;  
a = a--;
```

```
int a;  
a = 5;  
a = --a;
```

◆ 스택의 구현

★ 스택의 생성

```
#define STACK_SIZE 10  
typedef int element;  
element stack[STACK_SIZE];  
int top = -1;
```

◆ 스택의 구현

★ 스택의 삭제 연산

```
element pop( ) {  
    if (top == -1) {  
        printf("Stack is Empty!!\n");  
        return 0;  
    }  
    else  
        return stack[top--];  
}
```


◆ 스택의 구현

★ 스택의 삽입 연산

```
void push(element item) { //스택의 삽입 연산, item=400
    if (top >= STACK_SIZE-1) { //스택이 이미 FULL 인 경우
        printf("Stack is Full!!\n");
        return;
    }
    else
        stack[++top] = item;
}
```

03

사칙 연산식의 표현

◆ 수식의 표현

★ 수식의 계산

◆ 연산자의 계산순서를 생각해야 함

◆ $A+B*C+D$

➤ $((A+(B*C))+D)$

◆ 수식의 표현

★ 수식의 표기 방법

- ◆ 중위 표기법(infix notation)
 - 연산자를 피연산자 사이에 표기하는 방법
 - $A+B$
- ◆ 전위 표기법(prefix notation)
 - 연산자를 피연산자 앞에 표기하는 방법
 - $+AB$
- ◆ 후위 표기법(postfix notation)
 - 연산자를 피연산자의 뒤에 표기하는 방법
 - $AB+$

◆ 수식의 표현

★ 전위 표기법

$$(A - ((B+K) / D))$$

$$A + B \Rightarrow +AB$$

$$◆ (A - ((\underline{B+K}) / D))$$

$$\Rightarrow (A - (\underline{(+BK)} / \underline{D}))$$

$$\Rightarrow (\underline{A - (/ (+BK) D)})$$

$$\Rightarrow - A (/ (+BK) D)$$

◆ 수식의 표현

★ 후위 표기법

$$(A - ((B+K) / D))$$

$$A + B \Rightarrow AB+$$

$$◆ (A - ((\underline{B+K}) / D))$$

$$\Rightarrow (A - ((\underline{BK+}) / \underline{D}))$$

$$\Rightarrow (\underline{A - ((BK+) D /)})$$

$$\Rightarrow A ((BK+)D/) -$$

◆ 수식의 표현

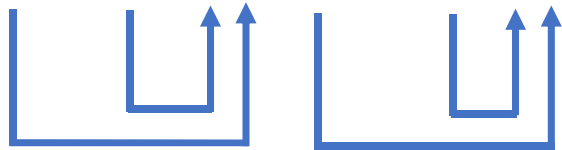
★ 중위 표기식의 후위 표기식 변환 방법

- ◆ 먼저 중위 표기식을 연산자의 우선순위를 고려하여 (피연산자, 연산자, 피연산자)의 형태로 괄호로 묶어줌
- ◆ 각 계산뭉치를 묶고 있는 괄호 안에서 연산자를 계산뭉치의 가장 오른쪽으로 이동시킴
- ◆ 각 계산뭉치를 하나의 피연산자로 고려하여 위를 반복함
- ◆ 괄호를 모두 제거함

◆ 수식의 표현

★ 중위 표기식의 후위 표기식 변환 방법

◆ $((A - (B / C)) - (D * E))$



◆ $((A (B C /) -) (D E *) -)$

◆ 수식의 표현

★ 후위 표기식의 계산 알고리즘

```
// 후위 표기식(369*+)을 계산하는 연산
element evalPostfix(char *exp) {
    int oper1, oper2, value, i=0;
    int length = strlen(exp);
    char symbol;
    top = -1;
```


◆ 수식의 표현

★ 후위 표기식의 계산 알고리즘

```
for(i=0; i<length; i++) {  
    symbol = exp[i];  
    if(symbol != '+' && symbol != '-'  
        && symbol != '*' && symbol != '/') {  
        value = symbol - '0';  
        push(value); }  
}
```

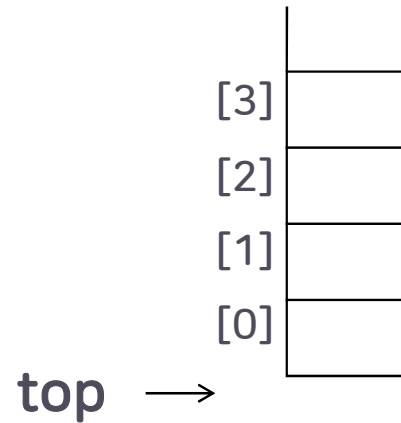
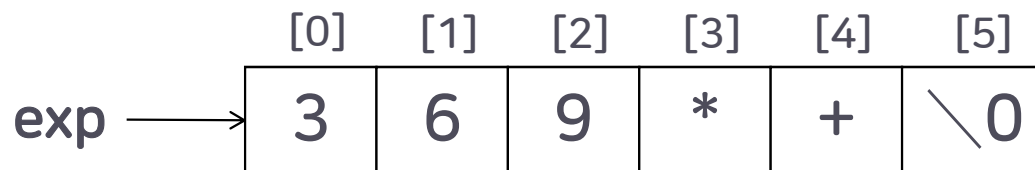
◆ 수식의 표현

★ 후위 표기식의 계산 알고리즘

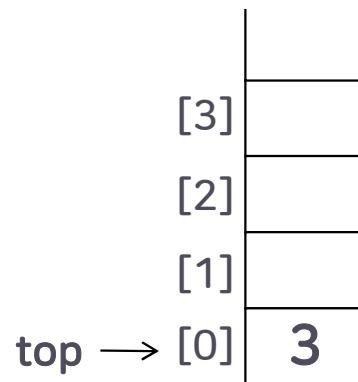
```
else { oper2 = pop( ); oper1 = pop( );  
  switch(symbol) {  
    case '+' : push(oper1 + oper2); break;  
    case '-' : push(oper1 - oper2); break;  
    case '*' : push(oper1 * oper2); break;  
    case '/' : push(oper1 / oper2); break;  
  } }  
return pop( ); }
```

◆ 수식의 표현

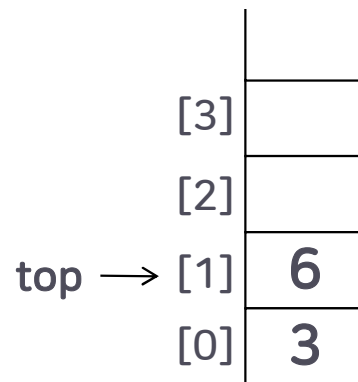
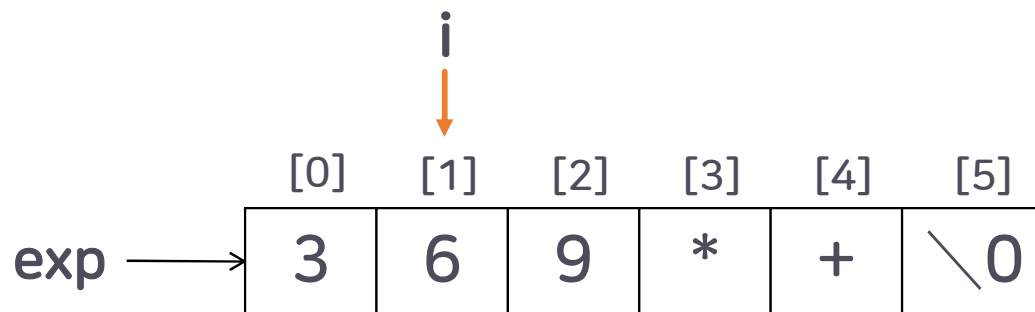
★ 수식이 저장된 배열과 스택의 초기 모습



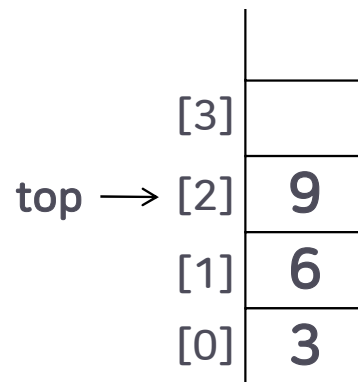
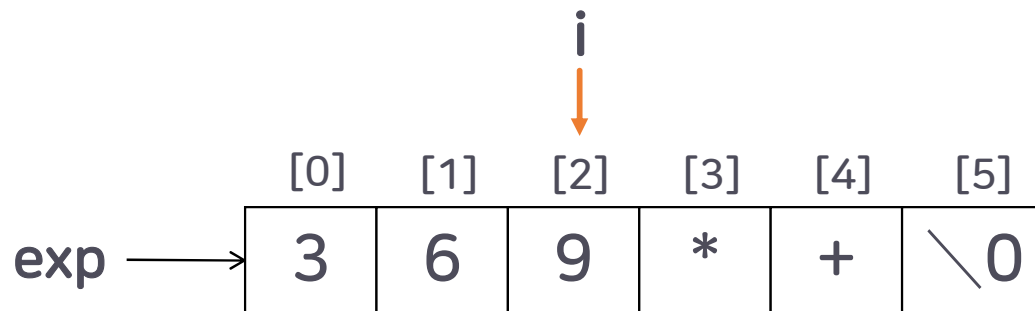
◆ 수식의 표현



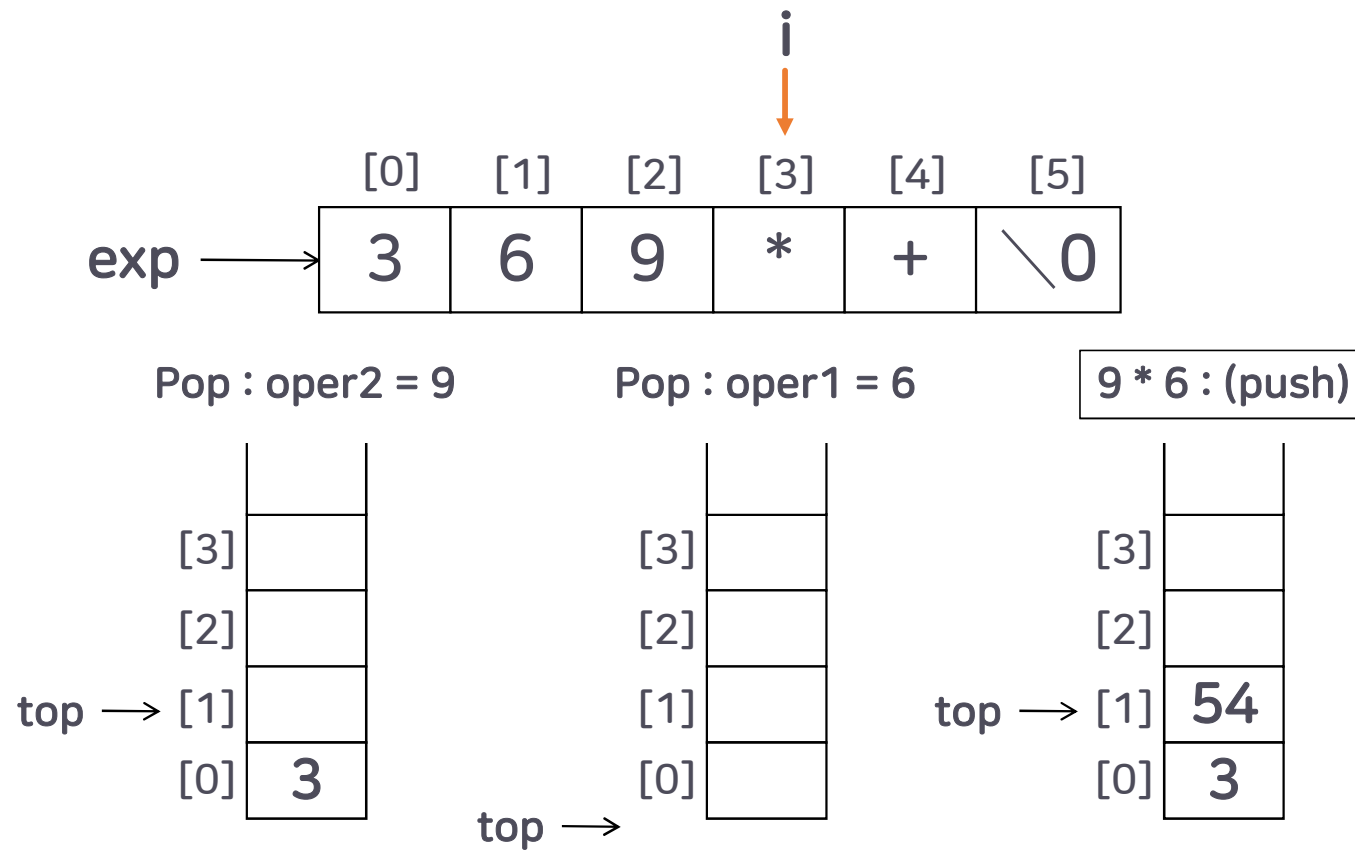
◆ 수식의 표현



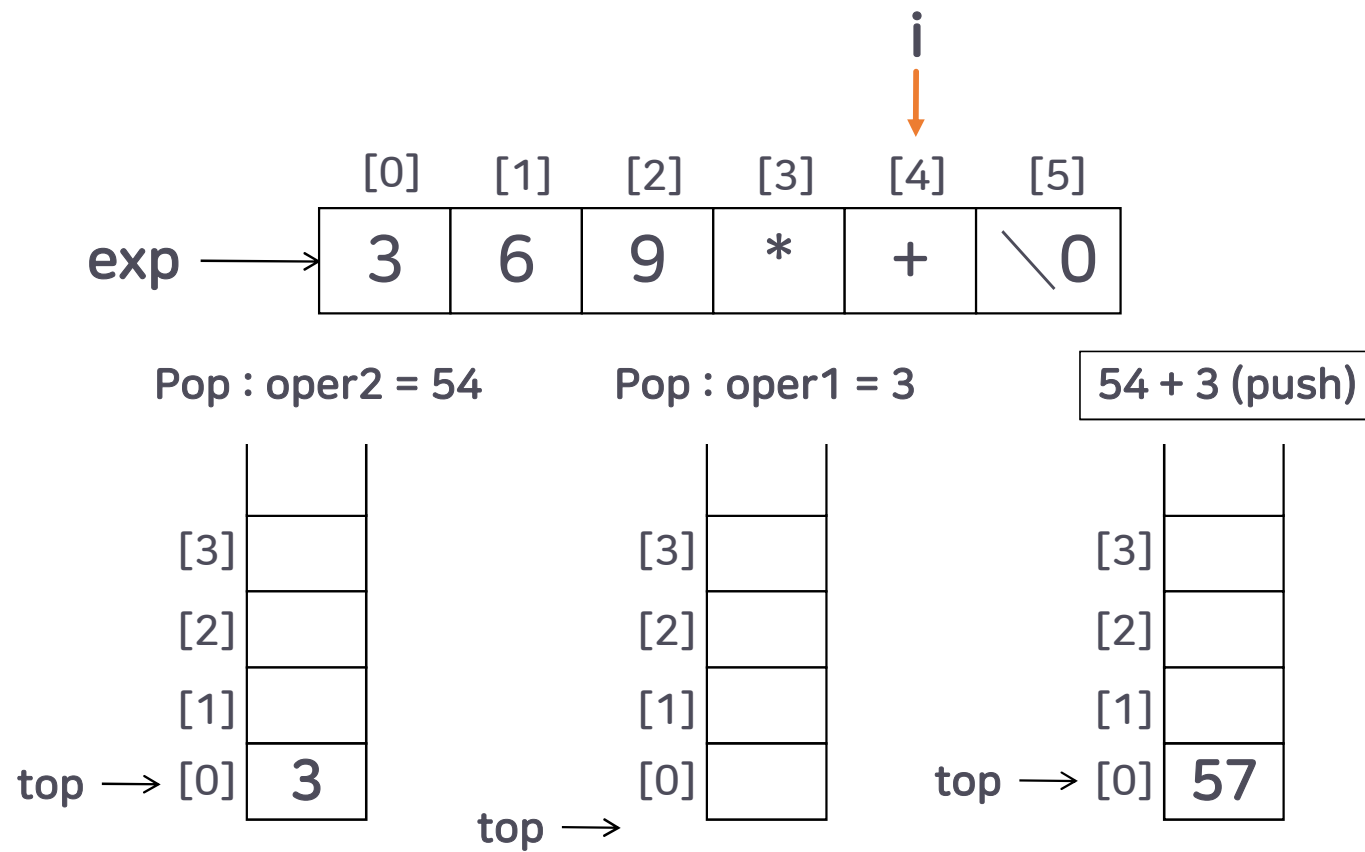
◆ 수식의 표현



◆ 수식의 표현

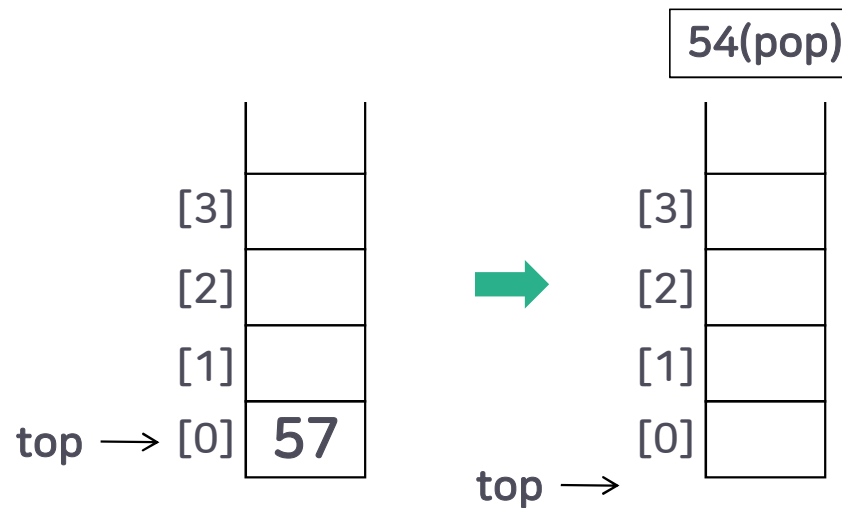


◆ 수식의 표현



◆ 수식의 표현

★ pop(57) 결과



다음 시간 안내

04강

수고하셨습니다.

큐