

Title: _____

Name: _____

Date: _____

후위표기 계산 알고리즘

① 수식 입력

② 수식을 처음부터 진행

1	3	5	*	+
---	---	---	---	---

③ 숫자라면 → stack push

④ 연산자라면 → pop twice in stack.

이때 첫번째로 pop된 숫자 = 두번째 피연산자

두번째로 pop된 숫자 = 첫번째 피연산자

⑤ 수식을 끝까지 진행했다면

↳ 만약 숫자가 $\underbrace{\text{stack에}}$ 1개 이상 남아있는 경우 → 잘못된 후위표식

↳ 하나만 남은 경우 ⇒ 올바른 결과 & 수식

Title: _____

Name: _____

Date: _____

중위표기식 → 후위표기식 변환 알고리즘

① 표현식을 처음부터 읽는다

② 만약 **숫자**라면 → 그대로 출력

③ 만약 **연산자**라면 → Stack에 삽입

A. '(' 소괄호 인 경우 → 그냥 삽입

B. '+, -, *, /' 인 경우 → Stack의 top 확인
* 이때 여는 소괄호는 무시

이때 스택 (들어가는 연산자)이 Stack의 top에
존재하는 연산자보다 **우선순위가 크다면**
⇒ 무시하고 Stack 삽입

본인이 Stack의 top에 존재하는 연산자보다
우선순위가 작다면

⇒ Stack의 top에 있는 연산자보다 우선순위가
크거나 같을 때까지 Stack.pop()

⇒ 종료되면 스택을 Stack에 push()

C. ')' 인 경우

→ Stack.pop()을 계속 수행한다

'(' 여는 소괄호를 만나면 pop()하고 종료

* 이때 소괄호는 출력하지 않는다.