

## 과제 2. 스택

# 과제 2

- ◆ 제출 마감일 : 4월 10일 (일요일) 10:59pm
- ◆ eClass 과제방에 제출
- ◆ 1인 과제
- ◆ 사용 언어
  - C / Java / Python / C++ 중 본인이 선택
  - 본인이 직접 만든 스택을 이용해야 한다.
- ◆ 제출 양식
  - 보고서 (표지 + 문제 내용과 해결 방안 + 결과 캡처)
  - 소스코드 (**별도 파일로 제출**)
- ◆ 보고서는 Word/HWP/PPT/PDF 중 하나로 제출

타인의 과제를 복사하지 말 것

# (1) 계산기

## ◆ 계산식을 입력하면 결과를 돌려주는 계산기 프로그램을 작성하라.

- 사용되는 연산자는 +, -, \*, /, ( ) 이다.
- 입력에 사용되는 수는 실수이다.
- 수식에서 괄호는 최대 1회만 사용한다. 겹쳐진 괄호는 쓰지 않는다.
- 잘못된 수식이라면 어느 위치에 문제가 있는지 알려줘야 한다.

## ◆ 실행 예)

$$2+3*4*4-1$$

$$= 49$$

# (1) 계산기

## ◆ 실행 예

$$2+(3*4)*2-12$$

$$=14$$

$$2+(3*4*2-12$$

^ 이 위치에 오류가 있습니다.

$$(11+3)*2*3-12$$

$$=72$$

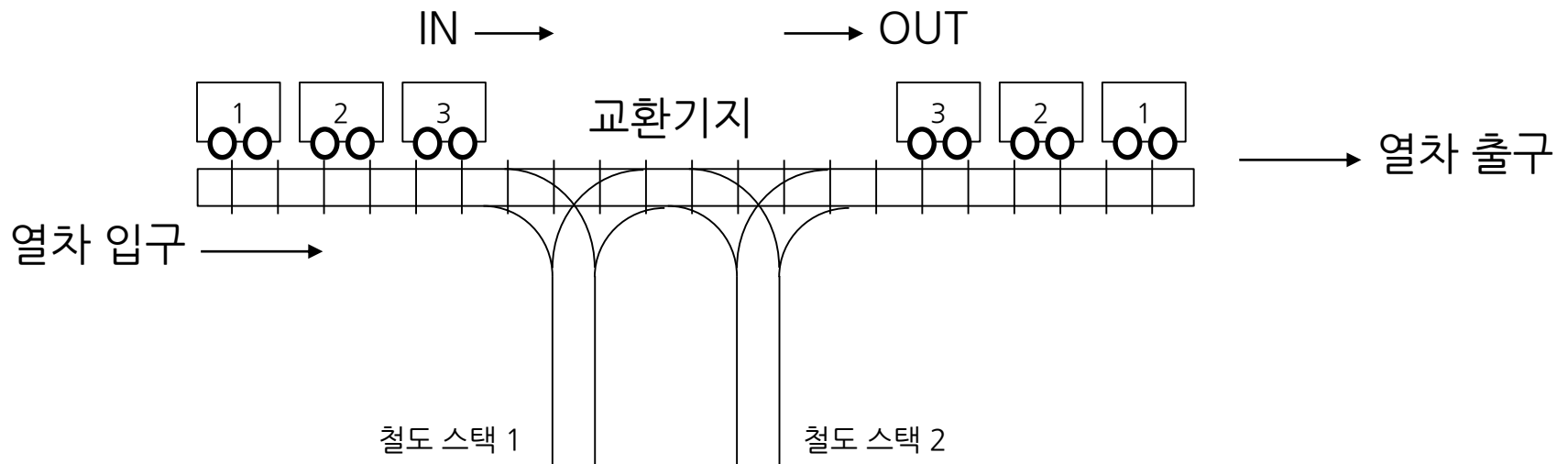
$$2+(3+*4^2-12$$

^ 이 위치에 오류가 있습니다.

## (2) 열차 재배치

### ◆ 이 곳은 지하철 기지이다.

- 왼쪽에서는 열차가 임의의 순서로 들어온다. 이 열차들을 오른쪽으로 나갈 때 순서대로 나가도록 만들어야 한다.
- 이를 위해 철도 스택  $n$ 개(2개~3개)를 만들었다.



## (2) 열차 재배치

- ◆ 철도 스택의 수와 들어오는 열차의 번호를 입력하면 열차들이 순차적으로 나갈 수 있도록 명령하는 기지 내 열차 컨트롤 프로그램을 작성하라.
  - 현재 준비된 철도 스택의 개수를 입력 받는다. (2, 3중의 하나)
  - 들어오는 열차는 스택에 넣거나 출구로 내보낼 수 있다.
  - 이미 출구로 나간 열차는 다시 진입할 수 없다.
  - 스택에 들어온 열차는 입구로 내보낼 수 없다. 스택에 들어온 열차는 다른 스택으로 보낼 수는 있다.

## (2) 열차 재배치

### ◆ 사용법

- 사용자는 철도로 들어오는 열차의 번호를 입력한다. 열차는 1번부터 9번까지 있다.
- 철도 스택을 이용해 출구로 나가는 열차는 1번부터 차례대로 나가야 한다. 이를 위해 열차 스택을 이용한다.
- 화면에는 다음의 내용을 표시한다.
  - 기지에 열차가 한 대씩 들어오는 상황 (3번 열차가 들어옴 : IN 3)
  - 철도 스택에 열차가 들어가는 상황(스택 번호 명시해야 함) (3번 열차가 1번 스택에 들어감 : PUSH 1, 3)
  - 철도 스택에서 열차가 나가는 상황(1번 스택에서 열차가 나감 : POP 1)
  - 대기 장소의 열차를 출구로 내보내는 상황 (OUT 1:1번 열차를 출구로 내보냄)
- 우리는 적은 횟수만큼 움직이는 것이 좋지만, 최적이 아니어도 괜찮다.

## (2) 열차 재배치

### ◆ 실행 예(최적의 정답이 아닐 수 있음)

스택의 개수 : 2

열차 번호 입력 : **397182645**

01 : IN(3)      3번 열차 들어옴  
02 : PUSH(1, 3)      3번을 1번 스택에  
03 : IN(9)      9번 들어옴  
04 : PUSH(2, 9)      9번을 2번 스택에  
05 : IN(7)      7번 들어옴  
06 : PUSH(2, 7)      7번을 2번 스택에  
07 : IN(1)      1번 들어옴  
08 : OUT(1)      1번을 출구로  
09 : IN(8)      8번 들어옴  
10 : PUSH(2, 8)      8번을 2번 스택에  
11 : IN(2)      2번 들어옴  
12 : OUT(2)      2번을 출구로  
13 : POP(1)      1번 스택에서 3번 뺌  
14 : OUT(3)      3번 출구로  
15 : IN(6)      6번 들어옴  
16 : PUSH(2, 6)      6번을 2번 스택에

계속

17 : IN(4)  
18 : OUT(4)  
19 : IN(5)  
20 : OUT(5)  
21 : POP(2)      2번 스택에서 6번  
22 : OUT(6)  
23 : POP(2)      2번 스택에서 8번  
24 : PUSH(1, 8)      8번을 1번 스택에  
25 : POP(2)      2번 스택에서 7번  
26 : OUT(7)  
27 : POP(1)      1번 스택에서 8번  
28 : OUT(8)  
29 : POP(2)      2번 스택에서 9번  
30 : OUT(9)  
종료(총 30회)

자료구조



## (2) 열차 재배치

### ◆ 실행 예(스택이 3개인 경우)

스택의 개수 : 3

열차 번호 입력 : **397182645**

01 : IN(3)	3번 열차 들어옴
02 : PUSH(1, 3)	3번을 1번 스택에
03 : IN(9)	9번 들어옴
04 : PUSH(2, 9)	9번을 2번 스택에
05 : IN(7)	7번 들어옴
06 : PUSH(3, 7)	7번을 3번 스택에
07 : IN(1)	1번 들어옴
08 : OUT(1)	1번을 출구로
09 : IN(8)	8번 들어옴
10 : PUSH(2, 8)	2번 스택에
11 : IN(2)	
12 : OUT(2)	
13 : POP(1)	1번 스택에서 3번 뺌
14 : OUT(3)	3번 출구로
15 : IN(6)	6번 들어옴
16 : PUSH(3, 6)	6번을 3번 스택에

계속

자료구조

스택의 개수가 달라지면  
운영 방식도 달라질 수 있다.

17 : IN(4)	
18 : OUT(4)	
19 : IN(5)	
20 : OUT(5)	
21 : POP(3)	3번 스택에서 6번
22 : OUT(6)	
23 : POP(3)	3번 스택에서 7번
24 : OUT(7)	
25 : POP(2)	2번 스택에서 8번
26 : OUT(8)	
27 : POP(2)	2번 스택에서 9번
28 : OUT(9)	
종료(총 28회)	

## (2) 열차 재배치

### ◆ 더 효율적인 재배치를 위한 방법

- 효율적이라는 말은 PUSH, POP이 덜 이루어진다는 의미이다.
- 열차는  $n$ 개의 스택 중 한 곳에 쌓을 수 있다. 어느 곳에 쌓는 것이 더 효율적일까?
- 스택의 개수가 달라지면 정책이 달라질 수 있다. 정책은 여러분이 정하는 것이다.
- 효율적인 스택 선택을 위해서 스택이 제공해야 하는 기능이 더 있는 않을까? 스택은 여러분의 요구에 의해 변경될 수 있다. 이 부분은 보고서에 나타내야 한다(중요).