

## Week 3

- 주의 사항: 부정행위 금지, STL 사용 금지(string 사용가능), 인터넷 금지, **배열 또는 링크드 리스트**를 이용하여 구현할 것.
- 표준 입출력 사용을 권장 (C는 scanf / printf, C++은 cin / cout)

### 문제 2

우리가 일반적으로 수식을 표기할 때, 다음과 같이 표기한다.

$$A * B + C$$

$$2 + 3 * 5$$

이와 같이 (피연산자)(연산자)(피연산자)의 순서로 두 피연산자 사이에 연산자를 표기하는 방법을 **중위표기법**이라고 부른다.

그런데 컴퓨터에서 중위표기 수식을 순서대로 계산할 경우, 연산자의 우선순위를 고려하지 못해 애로사항이 생기게 된다. 예를 들어  $2 + 3 * 5$ 의 경우,  $*$  연산이 우선순위가 있지만  $+$ 가 앞에 있기 때문에 순서대로 계산하는 컴퓨터에서는 이를 적절하게 처리하기가 힘들다.

이 때문에 컴퓨터 프로그램에서는 수식 계산을 쉽게 하기 위하여 중위 표기된 수식을 다음과 같이 변환하여 사용한다.

$$A B * C +$$

$$2 3 5 * +$$

이처럼 (피연산자)(피연산자)(연산자)의 순서로 **연산자의 우선순위를 고려**하여 연산자를 피연산자의 뒤에 표기하는 방법을 **후위표기법**이라고 부른다.

#### 주의)

1.  $+$ ,  $-$  연산 중 앞서 등장한 연산이 우선순위가 있다고 간주한다.

입력 :  $2+3*5-2*4$  → 출력 :  $235*+24*-$  (O) /  $235*24*-+$  (X)

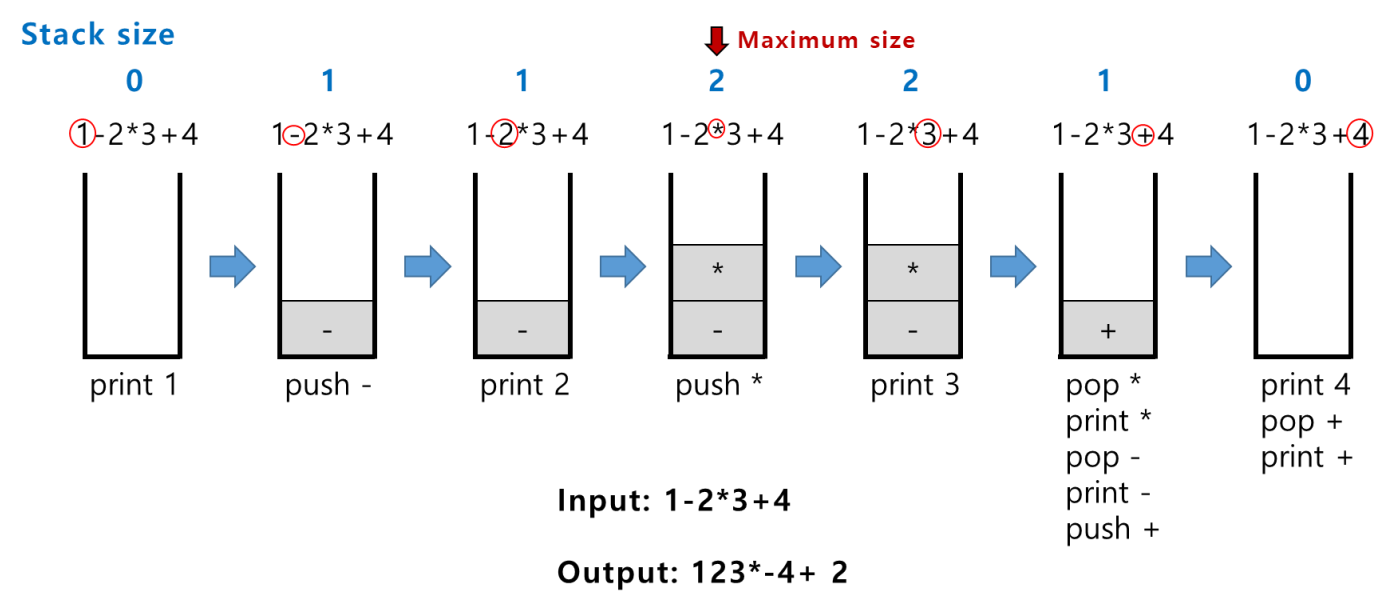
2.  $*$  연산들 간에도 앞서 등장한 연산이 우선순위가 있다고 간주한다.

입력 :  $1*2*3$  → 출력:  $12*3*$  (O) /  $123**$  (X)

3. 중위표기법에 먼저 등장한 숫자가 후위표기법에서도 먼저 등장한다.

입력 :  $2+3$  → 출력 :  $23+$  (O) /  $32+$  (X)

중위표기법이 입력으로 들어왔을 때 [그림 1]과 같이 후위표기법으로 변환하여 출력하고, 연산 과정에서 계산되는 스택의 최대 크기를 출력해보자.



[그림 1] 입력 "1-2\*3+4"에 대한 후위 표기법 변환 연산 수행 과정 예시

# 입력

표준 입력으로 다음과 같이 주어진다.

첫째 줄에 주어지는 수식의 수  $t$  ( $1 \leq t \leq 10,000$ )가 주어진다. 둘째 줄부터  $t$ 개의 줄에는 각각 중위표기법으로 적힌 수식이 띄어쓰기 없이 주어진다. 이 때 수식은 정수( $1 \leq N \leq 9$ )와 연산자(+, -, \*)로만 이루어진다. (단, 주어지는 수식의 길이  $P$  ( $3 \leq P \leq 99$ ))

# 출력

출력해야 하는 후위 연산식과 연산 과정에서 계산되는 스택의 최대 크기를 공백으로 구분하여 한 줄에 하나씩 출력한다.

# 예제 입출력

예제 입력	예제 출력
7	123*-4+ 2
1-2*3+4	12*3- 1

1*2-3	23*56*+7-3+ 2
2*3+5*6-7+3	789*+23*- 2
7+8*9-2*3	64*4*2*3+5- 1
6*4*4*2+3-5	37+6+ 1
3+7+6	25*4*3+17*+2+ 2
2*5*4+3+1*7+2	

### <Hint>

1. 입력 받은 string s의 길이는 **s.size()** 또는 **s.length()**로 알 수 있습니다.

```
string s = "hello";
```

```
int s_size = s.size();    // s_size의 값은 5가 된다.
```

2. 입력 받은 string s의 특정위치 문자는 **s.at(1)** 또는 **s[1]** 방법으로 접근할 수 있습니다.

```
string s = "hello";
```

```
s.at(0);    // 'h'
```

```
s[1];    // 'e'
```